Robot Pentester

INFORME DEL PROYECTO

ACHS

Versión 0.0.1

14/11/2024

# HISTORIAL DE VERSIONES

| VERSIÓN | APROBADO POR | FECHA DE REVISIÓN | DESCRIPCIÓN DEL CAMBIO | AUTOR |
| --- | --- | --- | --- | --- |
| 0.0.1 | -- |  | Creación informe | Benjamin Ramirez |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# FUNCIONES Y RESPONSABILIDADES

| NOMBRE | ROL | RESPONSABILIDADES |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Tabla de contenidos

[HISTORIAL DE VERSIONES 2](#_Toc183690568)

[FUNCIONES Y RESPONSABILIDADES 2](#_Toc183690569)

[1. Introducción 4](#_Toc183690570)

[1.1 Descripción general 4](#_Toc183690571)

[1.1.1 Objetivos del Proyecto 4](#_Toc183690572)

[1.1.2 Alcance del proyecto 5](#_Toc183690573)

[2. Especificaciones Tecnicas 6](#_Toc183690574)

[2.1 Tecnologías Utilizadas 6](#_Toc183690575)

[2.2 Arquitectura del Sistema 7](#_Toc183690576)

[3. Funcionalidades Principales 9](#_Toc183690577)

[4. Reportes e Interpretación de Resultados 10](#_Toc183690578)

[5. Instalación y Configuración 14](#_Toc183690579)

[6. Guías de Uso 14](#_Toc183690580)

[7. Anexos 15](#_Toc183690581)

# Introducción

|  |
| --- |
| Robot Pentester es una herramienta de pruebas de seguridad automatizadas diseñada para identificar, analizar y reportar vulnerabilidades en páginas webs. Su objetivo principal es realizar análisis de ciberseguridad automatizados, integrando pruebas de seguridad en todo el ciclo de desarrollo de un proyecto, generando informes detallados y asegurando la calidad de los sistemas mediante la detección temprana de posibles fallos de seguridad. Esto ayuda a reducir riesgos, costos y tiempos en la mitigación de vulnerabilidades.  Robot Pentester utiliza distintas tecnologías para alcanzar sus objetivos, entre las cuales destacan JMeter para pruebas de carga y rendimiento, OWASP ZAP para escaneos de seguridad y Metasploit para la validación y simulación de ataques de alto riesgo en un entorno de pruebas seguro, como Kali Linux. Todas estas herramientas se integran de manera eficiente en una infraestructura gestionada con Docker, lo que facilita su implementación y uso en entornos de desarrollo continuo.  Esta solución permite a los equipos de ciberseguridad realizar análisis de ciberseguridad robustos y eficientes, optimizando tiempos y recursos. |

## 1.1 Descripción general

|  |
| --- |
| 1.1.1 Objetivos del Proyecto |
| **Objetivo general**   * Desarrollar una herramienta de automatización de pruebas de seguridad, que permita identificar, analizar, reportar y explotar vulnerabilidades en aplicaciones web de manera continua y eficiente.   **Objetivos específicos**   * **Automatizar Pruebas de Seguridad:** Integrar diversas herramientas de análisis de vulnerabilidades, como OWASP ZAP para escaneos de seguridad, JMeter para pruebas de carga y rendimiento, y Docker para gestionar entornos aislados, asegurando que las pruebas se ejecuten de manera sistemática y eficaz. * **Validación y Explotación Controlada de Vulnerabilidades:** Implementar el uso de Metasploit para explotar vulnerabilidades detectadas por Robot Pentester en un entorno seguro, usando Kali Linux. Esta funcionalidad permite verificar la gravedad y el impacto de las vulnerabilidades en aplicaciones web y sistemas antes de aplicar correcciones. * **Generación de Informes Detallados:** Producir informes técnicos y resumidos que documenten las vulnerabilidades detectadas y, cuando sea pertinente, los ataques efectuados. Estos informes serán tanto exhaustivos para los equipos técnicos como concisos para un mayor entendimiento. * **Reducción de Riesgos y Costos:** Disminuir los riesgos y costos asociados con la detección tardía de vulnerabilidades, promoviendo la identificación y mitigación temprana de fallos de seguridad a lo largo del ciclo de desarrollo de software. * **Integración en Entornos de Desarrollo Basados en CI/CD:** Desplegar Robot Pentester en entornos basados en Docker para facilitar la integración continua (CI/CD) y permitir que las pruebas de seguridad se ejecuten de forma rápida y accesible tanto en entornos de prueba como de producción. |
| 1.1.2 Alcance del proyecto |
| **Análisis Automatizado de Seguridad:** Robot Pentester utilizará herramientas tecnológicas, como OWASP ZAP y Metasploit, para identificar vulnerabilidades de seguridad en aplicaciones web. Estas pruebas automatizadas incluirán la detección de problemas críticos de seguridad, como inyecciones de código, Cross-site scripting (XSS), y fallos de configuración de seguridad. Este análisis se realizará de forma continua, integrándose en el flujo de desarrollo para identificar y remediar vulnerabilidades de manera temprana y constante.  **Pruebas de Carga y Rendimiento con Jmeter:** JMeter se usará para evaluar la resistencia y capacidad de la aplicación web bajo distintas condiciones de carga. Este proceso permitirá simular múltiples usuarios y grandes volúmenes de tráfico para verificar cómo responde el sistema bajo estrés. Estas pruebas no solo detectarán puntos débiles en cuanto a rendimiento, sino que también ayudarán a identificar potenciales problemas de seguridad asociados a la sobrecarga del sistema  **Implementación y Escalabilidad con Docker:** El proyecto aprovechará Docker para ofrecer un entorno de pruebas configurado y escalable. Esto permite a los equipos de desarrollo y pruebas implementar y ejecutar Robot Pentester en distintos entornos sin complicaciones, garantizando que el sistema esté alineado con las necesidades de CI/CD (Integración y Entrega Continua). Docker también facilita la replicabilidad de los entornos de prueba, asegurando que las pruebas de seguridad se realicen en condiciones consistentes, independientemente del sistema operativo o configuración de la máquina.  **Generación de Reportes Detallados:** Robot Pentester generará informes exhaustivos que documentarán cada hallazgo de seguridad con detalles técnicos precisos. Estos reportes incluirán información sobre las vulnerabilidades detectadas, además, Robot Pentester generara un informe detallado técnico y un informe para un simple entendimiento.  V**alidación de Vulnerabilidades en un Entorno Controlado con Kali Linux y Metasploit:** Las vulnerabilidades críticas que identifique RobotPentester serán evaluadas en un entorno seguro y controlado utilizando **Kali Linux** y **Metasploit**. Esto permitirá realizar ataques simulados que validen el impacto real de cada vulnerabilidad, verificando la gravedad y naturaleza de las amenazas. |

# Especificaciones Tecnicas

|  |
| --- |
| 2.1 Tecnologías Utilizadas |
| Robot Pentester trabaja con una variedad de tecnologías, las cuales trabajan en conjunto y sinergia para el funcionamiento de esta.  A continuación, se detallan cada una de ellas.   * **Python (\*)**: Es el lenguaje principal en el cual se programa el desarrollo de scripts, la versión utilizada, ofrece características avanzadas para el proyecto, Como se menciona en su página oficial las principales son:   + **Integración sencilla**: Facilita el mantenimiento del código y conecta las distintas tecnologías usadas en el proyecto.   + **Mejoras en la ejecución asíncrona**: Perfecto para coordinar múltiples tareas como el monitoreo del tráfico con OWASP ZAP.   + **Gestión de flujo de trabajo**: A través de bibliotecas se automatiza el inicio y sincronización de procesos, como la ejecución de pruebas o el análisis de resultados.   + **Reportes personalizados**: Generación automatizada de informes detallados en formato HTML mediante plantillas específicas ofrecidas por OWASP ZAP y su complemento Export Report. * **Docker (\*)**: Es la tecnología clave que usa Robot Pentester, Como se describe en su página oficial. Docker permite la orquestación y aislamiento de los diferentes componentes. El uso de Docker facilita el despliegue e interacción entre contenedores.   Docker fue utilizado por las siguientes razones:   * + **Aislamiento**: Cada herramienta (JMeter, OWASP ZAP, Metasploit) corre en su propio contenedor, evitando conflictos de configuración.   + **Escalabilidad**: Permite levantar múltiples instancias de contenedores para manejar cargas más grandes o realizar pruebas en paralelo.   + **Portabilidad**: Robot Pentester puede ejecutarse el proyecto en cualquier sistema compatible con Docker sin preocuparse por las dependencias locales. * **Jmeter (\*)**: Herramienta importante, su principal funcionamiento es replicar flujos funcionales de los usuarios, sin Jmeter no sería posible generar datos que permitan a OWASP ZAP identificar fallos de seguridad. Jmeter nos permite registrar y simular interacciones de usuario en aplicaciones web. Jmeter nos genera tráfico de datos los cuales son utilizados por OWASP ZAP para realizar su escaneo. JMeter es compatible con múltiples protocolos y servicios, incluyendo HTTP/HTTPS, WebSockets, bases de datos, y más. Esto asegura que Robot Pentester pueda ser utilizado en una amplia variedad de aplicaciones web * **OWASP ZAP (\*):** Es primordial esta herramienta para Robot Pentester, OWASP ZAP se encarga de realizar un análisis de vulnerabilidades, OWASP ZAP funciona como un proxy el cual inspecciona y analiza el tráfico web de la página a la cual Robot Pentester le está aplicando su escaneo. Sus funcionalidades abarcan desde el mapeo básico del sitio hasta el descubrimiento profundo de vulnerabilidades críticas.   OWASP ZAP, fue utilizado principalmente por:   * + **Escaneo pasivo**: Analiza el tráfico de la aplicación en busca de vulnerabilidades sin modificarlo.   + **Automatización de pruebas de seguridad**: En RobotPentester, se utiliza una API para ejecutar escaneos de manera programada y generar informes detallados.   + **Integración con otras herramientas:** ZAP intercepta el tráfico generado por JMeter y lo analiza en busca de vulnerabilidades. * **Kali Linux y Metasploit (\*):** Metasploit es una nueva herramienta añadida a Robot Pentester, ejecutado en un contenedor Docker con Kali Linux, es una plataforma avanzada para realizar pruebas de penetración. Su integración en Robot Pentester agrega una capa adicional de validación de vulnerabilidades críticas**.**   Las principales razones para usar metasploit es:   * + **Explotación de vulnerabilidades**:Permite comprobar si las vulnerabilidades detectadas son explotables.   + **Módulos personalizados**: Los scripts de Python pueden controlar Metasploit para realizar pruebas automatizadas de exploits.   + **Entorno controlado**: Al ejecutarse en un contenedor de Kali Linux, las pruebas no afectan al entorno real del cliente.   (\*) Python 3.10.11, documentación oficial disponible en <https://www.python.org/doc/>  (\*) Docker versión 20.10.21, documentación oficial disponible en <https://docs.docker.com/>  (\*) Apache JMeter versión 5.5, guía oficial disponible en <https://jmeter.apache.org/>  (\*) OWASP ZAP versión 2.15.0, documentación accesible en <https://www.zaproxy.org/docs/>  (\*) Kali Linux, documentación oficial disponible <https://www.kali.org/docs/>  (\*) Metasploit documentación y información disponible en <https://docs.metasploit.com/> y <https://github.com/DanMcInerney/pymetasploit3> |
| 2.2 Arquitectura del Sistema |
| A continuación, se mostrará la arquitectura de Robot Pentester.  Descripción:  **Creación del Flujo Funcional y Script de Pruebas.**  En esta etapa, se utiliza JMeter para grabar recorridos funcionales por las páginas web de prueba. Resultando en un script.  **Levantamiento de Contenedores con Docker Compose.**  Se utiliza un archivo de configuración Docker Compose para desplegar tres contenedores principales en paralelo, Jmeter, OWASP ZAP y Kali Linux.  **Contenedor de Jmeter.**  Se encarga de hacer la ejecución del script generado con anterioridad.  **Contenedor de OWASP ZAP.**  Realiza escaneos de seguridad pasivos y activos para identificar vulnerabilidades en la aplicación.  **Identificación de vulnerabilidades.**  A través del contenedor de OWASP ZAP, se identifican las vulnerabilidades.  **Contenedor de Kali Linux.**  Se ejecuta un script de metasploit creado en Python el cual detecta las vulnerabilidades y las explota en un ambiente controlado.  **Generación de Informes Finales.**  Los resultados de las pruebas de carga (de JMeter), los escaneos de vulnerabilidades (de OWASP ZAP) y las explotaciones (de Kali Linux) se consolidan en informes detallados. Estos informes proporcionan un panorama integral sobre el rendimiento y la seguridad de la aplicación.  **Diagrama de arquitectura de Robot Pentester**  Diagrama  Descripción generada automáticamente  \*Seba revisará diagrama\* |

# Funcionalidades Principales

|  |
| --- |
| **Escaneo de seguridad automatizado**  Robot Pentester ofrece capacidades de escaneo automatizado para identificar vulnerabilidades de seguridad en aplicaciones web. A través de OWASP ZAP, analiza múltiples aspectos de la seguridad de un sitio web para así identificar la mayor cantidad de vulnerabilidades posibles.  **Gestión y Manejo de sesiones**  Robot Pentester permite guardar, cargar y gestionar sesiones de escaneo, lo que facilita trabajar con grandes proyectos o realizar análisis iterativos. Lo que permite tener continuidad de los escaneos en caso de interrupciones, Una referencia histórica de los análisis hechos, que está directamente relacionados con el uso de OWASP ZAP, que permite trabajar con sesiones para preservar el estado de un análisis y continuar o reutilizarlo posteriormente.  **Generación de informes personalizados**  Una de las características más útiles de Robot Pentester es su capacidad para generar informes detallados y visualmente claros, adaptados a diferentes necesidades. Robot Pentester es capaz de generar dos tipos de informes:  **HTML estándar:** Informe genérico para revisiones rápidas.  **Risk Confidence HTML:** Informe enfocado en riesgos priorizados por confianza y severidad.  Los informes generados por Robot Pentester contienen, clasificación de vulnerabilidades por severidad como también detalles técnicos sobre cada hallazgo.  **Pruebas de alta Severidad**  Utilizando Metasploit con Kali Linux, RobotPentester realiza pruebas controladas para validar las vulnerabilidades críticas identificadas. Esto permite verificar su impacto y definir planes de mitigación.  **Análisis completo de trafico**  RobotPentester captura y analiza el tráfico HTTP/HTTPS interceptado para identificar configuraciones inseguras, endpoints sensibles o patrones sospechosos, Robot Pentester hace un análisis de encabezados HTTP para detectar prácticas inseguras.  **Escalabilidad y Ejecución en Contenedores**  Lo que destaca a Robot Pentester la utilización de Docker para garantizar una configuración consistente y escalable. Esto permite desplegar la herramienta en cualquier entorno con mínima configuración. |

# Reportes e Interpretación de Resultados

|  |
| --- |
| A continuación, se muestran los informes obtenidos de pruebas hechas con Robot Pentester, por temas de seguridad no es posible mostrar el informe completo, si es necesario debe ser autorizado por una persona de rango alto.   * Las siguientes imágenes son referencias del informe detallado, donde podemos apreciar detalles como los parámetros con los cuales se priorizan vulnerabilidades:      * Se puede observar cómo se crean tablas dependiendo de las vulnerabilidades encontradas y sus parámetros.        * Al igual que se puede observar como se detallan las vulnerabilidades encontradas.      * El informe tradicional es más simple, donde se puede observar como se muestran la cantidad de vulnerabilidades encontradas con su nivel de riesgo, además de detallarlas. |

# Instalación y Configuración

|  |
| --- |
| La instalación y configuración de Robot Pentester se realiza siguiendo un conjunto de pasos básicos que aseguran la correcta implementación de las herramientas necesarias para ejecutar pruebas de carga y seguridad en entornos web. Este punto proporciona una introducción breve a los elementos clave del proceso.  Los elementos por ver en la instalación son:   1. Preparación del Entorno 2. Clonación del Repositorio 3. Configuración de Archivos 4. Ejecución de Pruebas Iniciales   Para un desglose detallado de los pasos, consulte **el Anexo 1: Instalación y configuración**, donde encontrará instrucciones paso a paso para completar el proceso. |

# Guías de Uso

|  |
| --- |
| El uso de Robot Pentester permite a los usuarios realizar pruebas automatizadas de carga y seguridad en aplicaciones web. Este capítulo ofrece una introducción a los flujos básicos de trabajo y los objetivos que se pueden alcanzar con la herramienta. Para instrucciones detalladas y ejemplos prácticos, consulte el **Anexo 2: Guía de Usuario.**  El flujo básico para usar Robot Pentester es:   1. Creación de un Script de Pruebas con Jmeter 2. Ejecución de los Contenedores 3. Análisis de Vulnerabilidades 4. Revisión de Resultados   Para un desglose detallado de los pasos, consulte el Anexo 2: Guía de Usuario, donde encontrará información detallada para su uso.  ¿Qué esperamos con el uso de robot Pentester?   * Los principales puntos que esperamos al usar Robot Pentester son. * Detectar y priorizar vulnerabilidades antes de que afecten a los usuarios finales. * Simular escenarios de alta carga para optimizar el rendimiento de la aplicación. * Reducir el esfuerzo manual en pruebas de seguridad y generación de reportes. |

# Anexos

|  |  |  |
| --- | --- | --- |
| ID | NOMBRE | UBICACIÓN |
| 1 | Instalación y Configuración |  |
| 2 | Guía de usuario |  |