 Robot Pentester:

Automatización de ciberseguridad

Integrantes: Benjamin Ramirez

Sebastian Acevedo

Maximiliano Hillmer

Jose Valdebenito

Capstone PTY4614-003V - Duoc UC

# Índice.

Contents

[Índice. 2](#_Toc182562520)

[Abstract 3](#_Toc182562521)

[Inglés 3](#_Toc182562522)

[Español 3](#_Toc182562523)

[Reflexión sobre tiempos y avances del proyecto 4](#_Toc182562524)

[Tiempos 4](#_Toc182562525)

[Factores facilitadores y factores que dificultan 4](#_Toc182562526)

[Factores facilitadores 4](#_Toc182562527)

[Factores que dificultan el trabajo 5](#_Toc182562528)

[Enfrentamiento de dificultades. 5](#_Toc182562529)

[Metodología 5](#_Toc182562530)

[Metodología utilizada 5](#_Toc182562531)

[Fases desarrolladas 6](#_Toc182562532)

[Justificación de la metodología 6](#_Toc182562533)

[Evidencias 7](#_Toc182562534)

[Conclusión 8](#_Toc182562535)

[Conclusion 8](#_Toc182562536)

# Abstract

## Inglés

RobotPentester is a tool designed to automate the detection and mitigation of security risks in software development. This project focuses on analyzing web flows, identifying security flaws, and generating detailed reports using JMeter scripts. The document provides a comprehensive overview of the project's development process, including its relevance, methodology, challenges, and achievements. The relevance of the project lies in its potential to enhance cybersecurity in software development, reduce costs, and ensure continuous evaluation. The project is aligned with various competencies of the graduate profile, including software quality assurance, project management, and cybersecurity. Furthermore, it reflects the team's professional interests in cybersecurity and software automation.

## Español

RobotPentester es una herramienta diseñada para automatizar la detección y mitigación de riesgos de seguridad en el desarrollo de software. Este proyecto se enfoca en analizar flujos web, identificar fallos de seguridad y generar informes detallados utilizando scripts de JMeter. El documento ofrece una visión integral del desarrollo del proyecto, incluyendo su relevancia, metodología, desafíos y logros. La relevancia del proyecto radica en su potencial para mejorar la ciberseguridad en el desarrollo de software, reducir costos y asegurar una evaluación continua. El proyecto está alineado con varias competencias del perfil de egreso, incluyendo aseguramiento de la calidad del software, gestión de proyectos y ciberseguridad. Además, refleja los intereses profesionales del equipo en ciberseguridad y automatización de software.

# Reflexión sobre tiempos y avances del proyecto

## Tiempos

El proyecto fue cumplido y llevado a cabo sin atrasos, todas las actividades fueron cumplidas en los tiempos establecidos en la carta Gantt. Sin embargo, nos gustaría destacar que actividades como algunas tareas de integración que fueron vista como una versión mejorada de robot pentester como integrar metasploit para hacer ataques a las vulnerabilidades encontradas por el robot, han tomado mas tiempo del necesario, esto debido, a la nueva adquisición de conocimientos y la dificultad para poder integrarlo al robot.

Sin embargo, estamos satisfechos, ya que al ser una actividad agregada para un futuro no afecta el hecho de que las demás actividades han sido completadas y el robot pentester funciona sin problemas

## Factores facilitadores y factores que dificultan

Robot Pentester es un proyecto de ciberseguridad, con el cual, tuvimos que aprender mucho para lograr el objetivo, sin embargo, al tener un equipo eficiente y una estructura clara el objetivo se ha podido lograr de la mejor manera

### Factores facilitadores

* **Acceso a recursos y herramientas de código abierto:** Herramientas como Docker, OWASP ZAP y JMeter proporcionaron un buen punto de partida y redujeron costos al no requerir licencias.
* **Apoyo de la planificación inicial:** Una estructura bien definida en la metodología cascada facilitó un enfoque claro y ordenado.
* **Colaboración del equipo:** La asignación adecuada de tareas y la buena comunicación entre los miembros permitieron avanzar en paralelo en diferentes áreas.
* **Documentación técnica:** Las guías y recursos en línea para las herramientas utilizadas fueron fundamentales para superar problemas técnicos y aprender nuevas funcionalidades.

### Factores que dificultan el trabajo

* **Compatibilidad entre herramientas:** Las versiones de Docker y OWASP ZAP presentaron incompatibilidades que ralentizaron la fase de integración.
* **Curva de aprendizaje:** Aunque las herramientas eran conocidas, ciertas configuraciones avanzadas requirieron tiempo adicional para ser comprendidas y aplicadas correctamente.
* **Limitaciones de tiempo:** Algunas actividades clave coincidieron con períodos de alta carga académica, dificultando dedicar más horas al proyecto en esas semanas.
* **Problemas de conectividad en entornos compartidos**: Durante las pruebas, algunos problemas de red afectaron el tráfico monitoreado por OWASP ZAP, retrasando los análisis.

### Enfrentamiento de dificultades.

* **Compatibilidad entre herramientas:** Se resolvió mediante la implementación de pruebas en entornos locales controlados antes de configurar todo en Docker. Además, se realizaron actualizaciones selectivas de versiones hasta encontrar las combinaciones óptimas.
* **Curva de aprendizaje:** Se invirtió tiempo adicional en la investigación de funcionalidades avanzadas de OWASP ZAP y en la consulta de documentación oficial y foros de la comunidad.
* **Limitaciones de tiempo:** Se reprogramaron actividades para priorizar tareas críticas y redistribuir actividades secundarias entre los miembros del equipo con mayor disponibilidad.

# Metodología

## Metodología utilizada

Se opto por utilizar la metodología **Cascada,** esto debido a su estructura secuencial, que permitió abordar cada etapa de forma organizada y detallada. Esta metodología fue clave para garantizar que todas las fases del proyecto se completaran correctamente antes de avanzar a la siguiente.

## Fases desarrolladas

1. **Análisis de requerimientos:**

* En esta fase, se identificaron los requisitos de seguridad específicos para el proyecto, documentando aspectos clave como las herramientas a utilizar, el alcance de las pruebas y los estándares de calidad esperados.
* A través de esta fase utilizamos Documentación técnica, reuniones con el equipo, análisis de guías de OWASP ZAP y JMeter.

2. **Diseño de arquitectura:**

* Se definió una arquitectura basada en Docker, con contenedores dedicados para cada herramienta (JMeter y OWASP ZAP).
* Este diseño aseguró que cada herramienta pudiera operar de forma aislada, reduciendo las interferencias y facilitando el monitoreo del tráfico.
* Se realizó un esquema detallado de las interacciones entre los contenedores, asegurando que la información fluya de manera eficiente entre las herramientas.

3. **Implementación de pruebas:**

* Se crearon scripts personalizados en JMeter para simular flujos funcionales de usuarios. Estos scripts generaron el tráfico necesario para que OWASP ZAP pudiera analizar y detectar vulnerabilidades en tiempo real.
* OWASP ZAP se configuró como un proxy para monitorear y escanear activamente el tráfico generado, enfocándose en vulnerabilidades críticas en aplicaciones web.

4. **Generación de informes:**

* Los resultados de las pruebas fueron procesados para generar informes automatizados que incluyeron una lista priorizada de vulnerabilidades, en el cual se generan dos informes, uno técnico y bien detallado y uno más simple y conciso.

## Justificación de la metodología

* **Estructura y claridad:** La metodología cascada permitió mantener un flujo de trabajo claro y organizado, asegurando que no se pasara por alto ninguna fase crítica.
* **Documentación detallada:** Cada fase produjo documentación exhaustiva, esencial para auditorías de seguridad y futuras iteraciones del proyecto.
* **Previsibilidad:** La metodología facilitó estimaciones precisas de tiempo y recursos, lo cual fue clave para coordinar actividades en el marco del semestre.
* **Estabilidad y confiabilidad:** En un proyecto de ciberseguridad, donde los riesgos son altos, la cascada ofreció una estructura predecible y estable que minimizó errores.

# Evidencias

A continuación, se mostrarán evidencias del avance y como hemos ido desarrollando el proyecto, se mostrarán las herramientas y algunos logs correspondientes.

Por temas de seguridad y filtrado de datos, no podemos mostrar evidencias como el git y el código del proyecto.

*Pruebas de JMeter a una página de la achs, por temas de seguridad y datos no se puede mostrar el resultado.*

*A screenshot of a computer

Description automatically generated*

*Levantamiento del robot pentester, el cual ejecuta un Docker compose, en el cual se generan todas las imágenes creadas con Docker y sus configuraciones para que le robot pueda ser utilizado a posterior.*

*A screenshot of a computer

Description automatically generated*

*Resultado de las imágenes levantadas*

*A screenshot of a computer

Description automatically generated*

# Conclusión

RobotPentester ha demostrado ser una solución eficaz para automatizar la identificación y mitigación de vulnerabilidades en el desarrollo de software. Durante el proyecto, logramos integrar herramientas de código abierto como OWASP ZAP y JMeter en una arquitectura basada en Docker, lo que permitió obtener resultados confiables y relevantes en nuestras pruebas. Aunque enfrentamos desafíos relacionados con la compatibilidad de herramientas y la curva de aprendizaje, las soluciones implementadas garantizaron el éxito del proyecto. Este esfuerzo no solo fortalece nuestras habilidades técnicas y de gestión, sino que también aporta valor significativo al ámbito de la ciberseguridad, estableciendo una base sólida para futuras mejoras en el sistema.

# Conclusion

RobotPentester has proven to be an effective solution for automating the identification and mitigation of vulnerabilities in software development. Throughout the project, we successfully integrated open-source tools like OWASP ZAP and JMeter into a Docker-based architecture, ensuring reliable and relevant testing outcomes. While we faced challenges related to tool compatibility and the learning curve, the implemented solutions ensured the project's success. This effort not only strengthens our technical and management skills but also delivers significant value to the cybersecurity field, providing a strong foundation for future enhancements to the system.