**Guía2. Desarrollo Proyecto APT**

**Asignatura Capstone**

|  |
| --- |
| **1. Resumen avance Proyecto APT** |
| A continuación, encontrarás distintos campos que deberás completar con la información solicitada. |

|  |  |
| --- | --- |
| Resumen de avance proyecto APT | Hasta el momento, se han completado la fase de planificación, que incluyó la definición de los objetivos específicos, la selección de herramientas (Docker Compose, JMeter, OWASP ZAP), y el diseño de un flujo automatizado para el escaneo de vulnerabilidades. En términos de implementación, hemos configurado el ambiente de Docker Compose para gestionar contenedores de JMeter y OWASP ZAP, permitiendo replicar flujos de usuario y monitorear el tráfico generado para la detección de vulnerabilidades. También se han generado scripts preliminares en JMeter que emulan las acciones del usuario para automatizar el flujo funcional de pruebas. |
| Objetivos | No se han realizado ajustes significativos en los objetivos principales. Los objetivos específicos se mantienen en generar un flujo de análisis automatizado, identificar vulnerabilidades y producir informes detallados. |
| Metodología | Para el desarrollo de RobotPentester se eligió la metodología en cascada debido a su estructura secuencial, que permite una hoja de ruta clara y organizada, asegurando que cada fase se complete antes de pasar a la siguiente. Esta metodología facilita la documentación detallada, fundamental para cumplir con normativas de seguridad y para auditorías futuras. Además, la previsibilidad de la metodología en cascada permite estimar plazos y presupuestos con precisión, lo cual es clave en proyectos con restricciones estrictas. Al tratarse de un proyecto de ciberseguridad, este enfoque es ideal, ya que prioriza la estabilidad y confiabilidad, aspectos esenciales para un análisis de riesgos en entornos de desarrollo. |
| Evidencias de avance | Se presentan capturas de pantalla del ambiente Docker configurado, que muestran los contenedores de JMeter y OWASP ZAP en funcionamiento. Además, nos gustaría incluir un primer informe generado que detalla vulnerabilidades detectadas en el tráfico de prueba simulado por JMeter. Estas evidencias demuestran el avance en la configuración y ejecución del entorno de prueba y en la integración inicial de las herramientas. La calidad de este avance se respalda mediante el cumplimiento de prácticas recomendadas en la configuración de Docker y Jmeter |

|  |
| --- |
| **2. Monitoreo del Plan de Trabajo** |
| Examina cuidadosamente tu plan de trabajo, enfocándote especialmente en la columna de estado de avance y ajustes. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Plan de Trabajo | | | | | | | | | |
| Competencia o unidades de competencias | Actividades | Recursos | Duración de la actividad | Responsable | Observaciones | Estado de avance | Ajustes | |
| Análisis y planificación de requerimientos informáticos | |  | | --- | | Análisis de Requerimientos de Seguridad |  |  | | --- | |  | | Reuniones, Documentos de requisitos, Herramientas de análisis | 3 semanas | Sebastián Acevedo | Podrían surgir dificultades al identificar todos los requisitos de seguridad; reuniones frecuentes facilitarán la claridad. a | Completada | No se realizaron ajustes | |
| Calidad de software | Pruebas de Caja Negra y Caja Gris | Caja Negra: Nmap, Metasploit, Burp Suite, OWASP ZAP. Caja Gris: SQLMap, JMeter | 2 semanas | Maximiliano Hilmer | La combinación de pruebas internas y externas podría ser compleja; herramientas adecuadas y experiencia ayudarán | En curso | No se realizaron ajustes | |
| Gestión de proyectos informáticos | Evaluación, Gestión de Proyecto y de Riesgos | Word, Excel, Project, Visual Studio | 2 semanas | Sebastián Acevedo | La gestión eficaz de riesgos depende de identificar factores críticos a tiempo; reuniones periódicas ayudarán en el monitoreo. | Completado | No se realizaron ajustes | |
| Programación de software | POO en Python, Consultas a API de OWASP ZAP | Visual Studio Code, Python, OWASP ZAP | 7 semanas | José Valdebenito, Benjamín Ramírez, Maximiliano Hillmer, Sebastián Acevedo | La complejidad de la API podría requerir documentación y pruebas extensivas; el uso de POO agilizará el manejo de métodos. | En curso | Intentamos integrar Metasploit la complejidad para intentar atacar vulnerabilidades nos está llevando más tiempo del necesario | |
| Arquitectura de software | Definir la cantidad y función de los contenedores necesarios para ejecutar las pruebas. | Docker | 3 semanas | Maximiliano Hilmer | La configuración inicial de contenedores puede presentar problemas de compatibilidad; pruebas previas y ajuste de versiones serán claves. | Completado | | No se realizaron ajustes |

|  |
| --- |
| **3. Ajustes a partir del monitoreo** |
| Profundiza en las observaciones de tu plan de trabajo. Analiza las actividades planificadas y señala qué aspectos facilitaron u obstaculizaron la ejecución del plan. Plantea cómo abordaste y/o abordarás los obstáculos. Por último, señala los ajustes que realizaste al plan de trabajo a partir de este análisis. |

|  |
| --- |
| Factores que han facilitado y/o dificultado el desarrollo de mi plan de trabajo:  Factores facilitadores incluyen el acceso a recursos de código abierto, como Docker y OWASP ZAP, y la disponibilidad de entornos de pruebas configurables para simular tráfico. Cosas que dificultaron el desarrollo del plan fueron temas de compatibilidad, ya que, al usar distintos programas la compatibilidad de estas puede variar dependiendo de la versión, Para solucionar estas incompatibilidades, realizamos actualizaciones controladas de versiones y pruebas de estabilidad en entornos de prueba antes de implementarlas en el proyecto. Otro factor que nos dificultad, era el estudio de estas tecnologías, al usar tecnologías nuevas, tuvimos que hacer cursos y obtener información a través de la red, sin embargo también cursos gratis, cursos en YouTube e información en internet nos facilitaron la ejecución del plan |

|  |
| --- |
| Actividades ajustadas o eliminadas:  No se realizaron ajustes significativos, todo salió como estaba previsto. Sin embargo, al querer agregar una nueva funcionalidad la cual es metasploit y automatizar un proceso, cuando se encuentre una vulnerabilidad esta posteriormente sea explota en un sistema controlado como Kali Linux. Integrar esta lógica nos llevó más tiempo del esperado, la integración aún está en proceso. |

|  |
| --- |
| Actividades que no has iniciado o están retrasadas:  Todo el proyecto esta con los tiempos correctos, ninguna actividad se ha retrasado. Solo una nueva integración la cual es metasploit |