ÍNDICE TFG RAQUEL ROMERO

1. Introducción
    - 1.1 Motivación
    - 1.2 Solución Propuesta
    - 1.3 Estructura de la memoria

2. Contexto teórico
    - 2.1 Auditorías de seguridad: normativa y metodologías
    - 2.2 Iptables
    - 2.3 NFLOG
    - 2.4 Analizadores de tráfico
    - 2.5 Python como herramienta de automatización
    - 2.6 EDR
        - 2.6.1 OpenEDR
    - 2.7 SIEM
        - 2.7.1 Elastic Stack
    - 2.8 Docker
    - 2.9 systemd-resolve
    - 2.10 Librerías de testing
        - 2.10.1 Bandit
        - 2.10.2 Flake8

3 Estado del Arte
    - 3.1 Windows Event Tracing
    - 3.2 eBPF
    - 3.3 Comparativa entre NFLOG y ULOG.
    - 3.4 Metodología de evaluación LINCE y metodologías europeas.
    - 3.5 Herramientas de automatización.
    - 3.6 Conclusiones

4. Objetico y planificación
    - 4.1 Objetivos
        - 4.1.1 Objetivos primarios
        - 4.1.2 Objetivos secundarios
        - 4.1.3 Asignaturas relacionadas
    - 4.2 Planificación y costes
        - 4.2.1 Planificación
        - 4.2.3 Costes

5. Diseño e implementación dela herramienta com-filter
    - 5.1 Análisis del problema
    - 5.2 Metodología
    - 5.3 Diseño
        - 5.3.1 Decisiones
        - 5.3.2 Diseño final
    - 5.3 Implementación
    - 5.4 Test
    - 5.5 Problemáticas resueltas

6. Evaluación de la herramienta com-filter
    - 6.1 Filtrado de ICMP.
        - 6.1.1 Diseño
        - 6.1.2 Implementación
        - 6.1.3 Test

        - 6.1.4 Conclusiones

6.2 Filtrado por usuario del sistema systemd-resolve.
6.2.1 Diseño
6.2.2 Implementación
6.2.3 Test
6.2.4 Conclusiones
6.3 Test funcionales automatizados
6.3.1 Bandit
6.3.2 Flake8

7. Auditoría de la comunicaciones de OpenEDR y ELK Stack.
7.1 Análisis inicial
7.2 Metodología
7.3 Despliegue del entorno
7.4 Escenario de evaluación
7.5 Funciones de seguridad
7.6 Automatización del análisis de comunicaciones seguras
7.6.1 Análisis del problema
7.6.2 Diseño
7.6.3 Implementación
7.7 Pruebas funcionales
7.8 Veredicto de la auditoría.

8. Conclusiones y trabajo futuro
8.1 Conclusiones
8.2 Trabajo futuro
8.3 Valoración personal

9. Bibliografía

1. Introducción

1.1 Motivación

- Por qué son importantes las auditorías de seguridad, alguna estadística mundial/europea relacionada con el aumento de los ataques en proporción con el aumento de auditorías realizadas, qué problemas o metologías afectan al rendimiento del evaluador, hincapié en los problemas generales de estudiar las comunicaciones de un producto y las soluciones que se están tomando actualmente en las auditorías.

1.2 Solución Propuesta

- Se explica que se realizará una autoría de EDR/SIEM y la normativa elegida para ello. Se menciona el “scope” del estudio que serán únicamente las comunicaciones y que, para ello, se mostrará una metodología de auditoría más eficiente desarrollada con la herramienta de com-filter y junto al parser. Se mencionan las características principales de ambas herramientas para justificar que son una solución, pero no se entra en demasiado detalle.

1.3 Estructura de la memoria

- Se explica cada capítulo de la memoria y en los que sea necesario se justificará el formato elegido.

2. Contexto teórico

- Se hace una pequeña introducción del contexto que abarca el TFG y se procede con subsecciones.

2.1 Auditorías de seguridad: normativa y metodologías

- Explicación teórica de lo que es una auditoría de seguridad en términos generales. Se explican las partes que suelen conformarlas y las metodologías usadas en los laboratorios.
- Se continúa con la exposición de los distintos esquemas de seguridad y normativas, por distinción entre países, profundidad de la evaluación, etc…
- Explicar en mayor profundidad la metología LINCE en la que se basará el TFG. (Si se hace demasiado largo, dividir por apartados para facilitar la lectura)

2.2 Iptables

- Explicar iptables y sus variantes. Conceptos teóricos fundamentales de iptables.

2.3 NFLOG

- Explicar qué es el NFLOG, sus conceptos teóricos y usos frecuentes.

2.4 Analizadores de tráfico

- Aquí se presentarán los diferentes analizadores de tráfico que se usan, explicando sus características y funcionalidades principales así como sus diferencias.
- Wireshark, tcpdump, tshark

2.5 Python como herramienta de automatización

- Sin entrar en demasiado detalle acerca de la programación en Python, se hace una introducción al lenguaje destacando su facilidad de uso para la automatización. Se comentan las principales librerías utilizadas iptc, pyshark y argparser. También se mencionará Poetry como tecnología usada para la posterior distribución de código con pip.

2.6 EDR

- Contextualización previa acerca de lo que es un EDR.

2.6.1 OpenEDR
- Que es OpenEDR y sus funcionalidades completas.

2.7 SIEM
- Qué es un SIEM, diferencia con un EDR

2.7.1 Elastic Stack
- Explicar la pila elastic, la funcionalidad de sus componentes. Formas de despliegues. Explicar funcionalidad de EDR incluida.

2.8 Docker
- Explicar en que consiste Docker y los contenedores, no demasiado largo, ya que se usará para el despliegue de la Elastic Stack.

2.9 systemd-resolve
- Explicar el usuario del sistema que se usará para el segundo test.

2.10 Librerías de testing
2.10.1 Bandit
2.10.2 Flake8

3 Estado del Arte
- Se hace una introducción al estado del arte.

3.1 Windows Event Tracing
- Comentar el funcionamiento y resaltar la compatibilidad únicamente con sistemas Windows.
- Explicar las soluciones a raíz de ahí como Winshark, de nuevo solo para Windows.

3.2 eBPF
- explicar las funcionalidades del proyecto haciendo hincapié en los niveles de uso (red, kernel, aplicación).
- Comentar la solución que se propone en el proyecto ecapture hecho con eBPF.

3.3 Comparativa entre NFLOG y ULOG.
- Aquí se aprovecha la explicación ya dada en el contexto acerca de NFLOG y se explica qué es el ULOG y las diferencias entre ambos.

3.4 Metodología de evaluación LINCE y metodologías europeas.
- Se hace una comparativa entre LINCE, Common Criteria y otra normativa más.

3.5 Herramientas de automatización.
- Aquí se pondrá en disputa Python como lenguaje de scripting y automatización, comparándolo con los lenguajes también usados para tareas similares como PHP.

3.6 Conclusiones
- Se expondrán las conclusiones extraídas para tomar las decisiones finales acerca de lo que se va a utilizar. Es aquí donde se incluyen las tablas comparativas de los apartados anteriores.

4. Objetivos y planificación

4.1 Objetivos
4.1.1 Objetivos primarios
- Se define el objetivo general que será llevar a cabo una auditoría de las comunicaciones de un entorno EDR/SIEM. Como objetivo más específico

se explicará que se quiere optimizar dicho proceso y para ello se tendrá como tarea el desarrollo y diseño de com-filter.

4.1.2 Objetivos secundarios

- Como objetivos secundarios, se buscará facilitar las tareas del evaluador con una herramienta complementaria que una vez filtradas las comunicaciones, permita filtrar la información relevante de cara a las comunicaciones. Se mencionan aquí las subtareas de testing, test con icmp y test con system resolver.

4.1.3 Asignaturas relacionadas

- En esta sección se expondrá un listado de las asignaturas y sus conceptos sobre los que se va a profundizar o de las que extrae conocimiento principalmente para llevar a cabo el TFG.

4.2 Planificación y costes

4.2.1 Planificación

- En este apartado se explicará la planificación inicial (diagrama de Gant) que se determinó y las dificultades que se tuvo para seguirlas. Aquí se mencionarán los diferentes approaches que se tomaron de primeras. Finalmente, se mostrará el diagrama de Gannt que se terminó cumpliendo.

4.2.3 Costes

- Costes de horas, material usado, comparativa con precios reales.

5. Diseño e implementación de la herramienta com-filter

5.1 Análisis del problema

- Aquí mencionarán en mayor detalle los problemas a resolver/mejorar con la herramienta. Se descompondrá el problema en pequeñas partes: funcionalidades, tipo de implementación, metodología de uso, etc...
- Se llegará finalmente a una conclusión justificada sobre las bases de com-filter.

5.2 Metodología

- Se explica la metodología de trabajo empleada; herramientas, entorno, etc.

5.3 Diseño

5.3.1 Decisiones

- Se explicarán las decisiones de diseño que se tuvieron que tomar según las premisas de lo que se buscaba.

5.3.2 Diseño final

- Dadas una decisiones tomadas, aquí se expondrá el diseño final que satisface todo lo anterior. Las librerías, funcionalidades, etc...

5.3 Implementación

- Aquí se determinará y explicará todo lo relacionado con la implementación, explicando el código, las funciones principales y cómo se usaría.

5.4 Test

- Se muestra una pequeña prueba de que la herramienta funciona pero sin entrar en mucho detalle de evaluación.

5.5 Problemáticas resueltas

- Se expondrán los problemas encontrados durante el desarrollo de herramienta y las soluciones que se dieron.

6. Evaluación de la herramienta com-filter

6.1 Filtrado de ICMP.

6.1.1 Diseño

- Cuáles son las premisas de diseño según lo que se busca (comprobar que se puede filtrar el ping de un usuario)

6.1.2 Implementación

- Explicar código de generación de ruido. Cómo se usaría.

6.1.3 Test

- Cómo se lleva a cabo el test, pasos y resultados.

6.1.4 Conclusiones

- Conclusión de los resultados obtenidos en este primer escenario.

6.2 Filtrado por usuario del sistema systemd-resolve.

6.2.1 Diseño

- Cuáles son las premisas de diseño según lo que se busca (comprobar que se pueden filtrar las comunicaciones de un usuario del sistema)

6.2.2 Implementación

- Explicar qué se ha necesitado para llevar a cabo el test y lo que se necesita.

6.2.3 Test

- Cómo se lleva a cabo el test, pasos y resultados.

6.2.4 Conclusiones

- Conclusión de los resultados obtenidos en el segundo escenario.

6.3 Test funcionales automatizados

6.3.1 Bandit

- Se mostrarán los resultados de testear vulnerabilidades de seguridad con Bandit

6.3.2 Flake8

- Se muestran los resultados de testear la calidad del código con Flake8.

7. Auditoría de la comunicaciones de OpenEDR y ELK Stack.

7.1 Análisis inicial

- Se introduce al entorno de la auditoría. Se analizan qué funcionalidades se llevarán a evaluación. Se mencionan también todas las asunciones iniciales que se tomarán.

7.2 Metodología

- Se explican las herramientas que se van a utilizar para la evaluación y el proceso que se va a llevar a cabo.

7.3 Despliegue del entorno

- Se muestran los pasos a seguir para desplegar el entorno de evaluación.

7.4 Escenario de evaluación

- Se explica detalladamente todo el entorno de evaluación: esquema de las comunicaciones, funcionalidades de cada parte, etc.

7.5 Funciones de seguridad
- Se exponen y explican las funciones de seguridad aplicables a la auditoría que se va a llevar a cabo, en este caso, todas las relacionadas con las comunicaciones.

7.6 Automatización del análisis de comunicaciones seguras

7.6.1 Análisis del problema
- Se explica el problema o retraso que genera analizar protocolos y comunicaciones contra una norma de seguridad.

7.6.2 Diseño
- Se expone la propuesta de diseño de una herramienta que automatice ese proceso. Las premisas que seguirá y lo que deberá de implementar y cómo.

7.6.3 Implementación
- Implementación de la herramienta en python, explicación del código y su funcionamiento.

7.7 Pruebas funcionales
- Aquí se dividirá por subsecciones cada test que se haga, explicando lo que se pretende poner a prueba, la funcionalidad de seguridad que debería de presentar, los pasos replicables a seguir y el resultado obtenido.
- En las pruebas que lo requieran, se emplearán las herramientas previamente desarrolladas.

7.8 Veredicto de la auditoría.
- Se hace un resumen de lo que se ha hecho y de los resultados obtenidos. Determinando si la evaluación tendría un veredicto positivo o no según las premisas marcadas como seguras.

8. Conclusiones y trabajo futuro

8.1 Conclusiones
- Resumen de lo que se ha llevado a cabo y conclusiones sacadas.

8.2 Trabajo futuro
- Mejoras que se podrían hacer o ampliaciones, margen de mejora.

8.3 Valoración personal
- Opinión personal del trabajo realizado

9. Bibliografía