Dirección de Proyectos II

DIVISIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN

INGENIERÍA EN REDES INTELIGENTES Y CIBERSEGURIDA

DATOS NOMBRE DEL ALUMNO(A):

García Martínez Luis Daniel
Cortes Alcalá Jessica Banelly
De la Rosa Vázquez Marco Antonio
Rodríguez López Ricardo

Sistema de Detección de Intrusos (IDS)

1. Introducción

El presente documento describe el desarrollo de un **Sistema de Detección de Intrusos (IDS)** basado en Python y Flask. Este sistema permite monitorear el tráfico de red en busca de actividad sospechosa y realizar escaneos de puertos para detectar posibles vulnerabilidades.

2. Objetivo

El objetivo principal de este IDS es proporcionar una herramienta sencilla pero efectiva para la detección de intrusiones en una red, ayudando a los administradores de sistemas y seguridad a identificar posibles amenazas en tiempo real.

3. Funcionalidades

El IDS desarrollado cuenta con las siguientes funcionalidades principales:

1. **Monitoreo de tráfico de red**: Captura y analiza paquetes de red para identificar patrones anómalos o sospechosos.
2. **Escaneo de puertos**: Permite verificar los puertos abiertos en una dirección IP objetivo para identificar posibles vulnerabilidades.
3. **Interfaz web**: Se implementa una interfaz web amigable desde la cual los usuarios pueden ejecutar el monitoreo y el escaneo de manera sencilla.
4. **Verificación de USB**: Analiza los puertos del dispositivo el cual guarda un id de verificación dentro del código para saber si la usb está registrada como autorizada o maliciosa
5. **Analizador de archivos**: Este analizador requiere de que se suba un archivo dentro del mismo en donde se podrá lograr verificar si el archivo contiene algun malware, se encuentra corrupto o presenta alguna falla en caso de no ser asi lo marcara como seguro

4. Interfaz

Dentro del código se encuentra una interfaz fácil y cómoda de usar para el usuario, ya que esta no cuenta con demasiada lógica en la que sea entendible y esta pueda llegar a confundir a la persona que lo utilice en el funcionamiento del mismo.

Esta interfaz se basa en un código simple de HTML implementando diversos estilos con lenguaje de css que es el más adecuado para interfaces de páginas web

La interfaz cuenta con una variedad de selección la cuales se basan en:

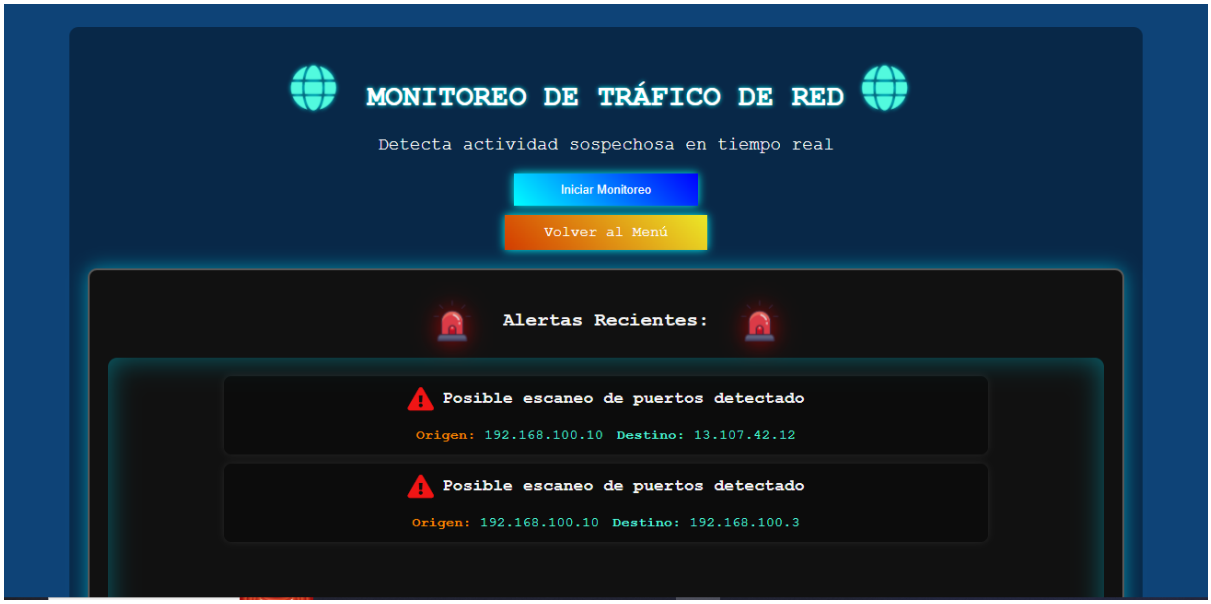
- Monitoreo de Tráfico
- Escaneo de Puertos
- Verificación de USB

- Analizador de archivos en busca de malware

Primero se presenta una interfaz de selección al usuario para que este aplique lo que necesite en dado caso de la selección le mandara una pantalla en donde realiza una acción diferente cada vez.

Monitoreo de Red

En este apartado podremos observar que nos arroja 2 botones los cuales son para regresar al menú principal o para realizar el monitoreo de los dispositivos en dado caso que se genere una alerta necesaria de dicho problema la interfaz nos mandará una alerta donde señale el tipo de ataque así como la ip de origen y destino



Escaneo de Puertos

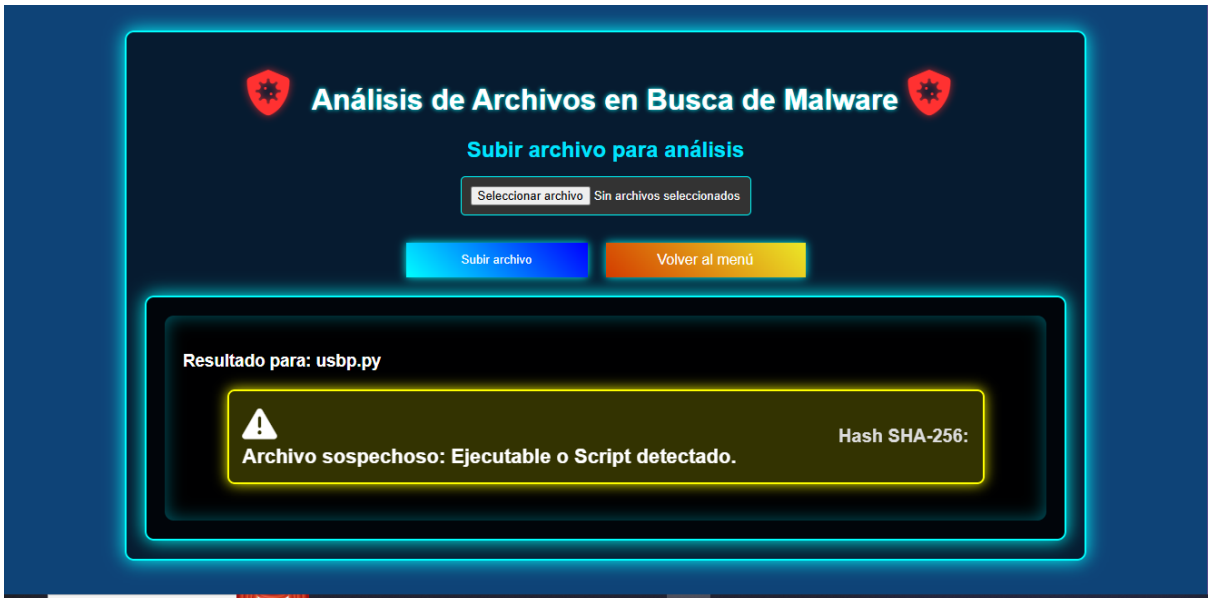
En este apartado podremos visualizar que primero nos pide una ip para escanear esto hará que la aplicación busque en el dispositivo cuales son los puertos abiertos y nos lo mostrará en alertas en donde se observa el número de puerto el protocolo al que pertenece, el estatus del puerto y el servicio bajo el que funciona dicho puerto

Detección de USB Conectados

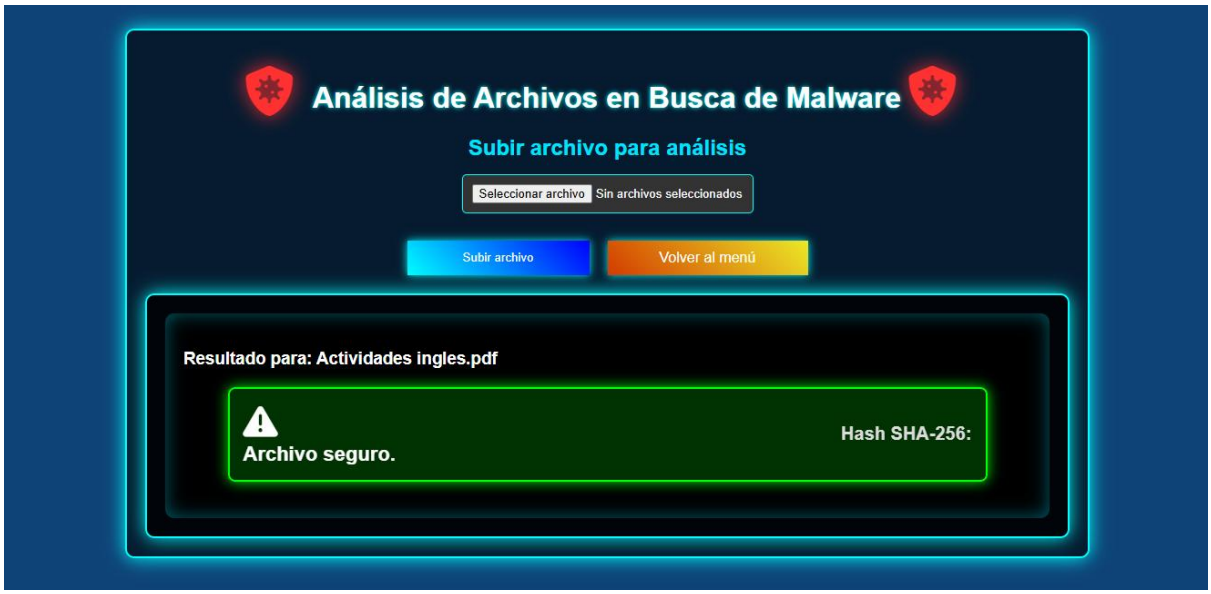
Esta funcionalidad se basa en una lista que está dentro del código en donde analiza y marcará como seguras las USB que nosotros queramos que sean de acceso para el sistema o las que se tenga un registro seguro en caso opuesto marcará como malicioso y mantendrá una alerta

Análisis de Archivos

En este apartado nos pedirá primero que podamos subir el archivo el cual vamos a escanear y posteriormente este mismo nos dirá que ocurre con el archivo y cual es su estatus como se muestra en la siguiente imagen:



En caso de ser un archivo seguro se vera de la siguiente manera:



5. Tecnologías utilizadas

Para el desarrollo del IDS se utilizan las siguientes tecnologías:

- **Python**: Lenguaje de programación principal del sistema.
- **Flask**: Framework para el desarrollo de la interfaz web y la comunicación entre el usuario y los módulos del IDS.

- **Scapy**: Biblioteca de Python utilizada para la captura y análisis de paquetes de red.
- **Nmap**: Herramienta utilizada para el escaneo de puertos.
- **Socket:** Biblioteca utilizada para la comunicación en red y detección de puertos abiertos.
- **PyUSB:** Biblioteca utilizada para la gestión de dispositivos USB en Python.
- **HTML, CSS y JavaScript:** Tecnologías para el diseño y funcionalidad de la interfaz web.

6. Estructura del Proyecto

El proyecto está organizado en la siguiente estructura de archivos:

```
/IDS
├── /static            # Archivos estáticos (CSS, JS)
├── /templates         # Archivos HTML
│   ├── index.html     # Página principal con acceso a todas las funciones
│   ├── monitor.html   # Interfaz para monitoreo de tráfico
│   ├── scanner.html   # Página para escaneo de puertos
│   ├── usb.html       # Interfaz de detección de USB
│   └── analysis.html  # Página de análisis de archivos
├── app.py             # Servidor Flask (maneja la lógica y las funciones para el ids)
└── requirements.txt   # Dependencias (Flask, Scapy, etc.)
```

7. Metodología de Desarrollo

Para el desarrollo del IDS se sigue la metodología **Cascada**, debido a su estructura secuencial y ordenada. Esta metodología se compone de las siguientes fases:

1. **Análisis de Requisitos**: Se definen los objetivos, funcionalidades y herramientas necesarias para el IDS.
2. **Diseño del Sistema**: Se estructura la arquitectura del sistema, incluyendo los módulos de monitoreo, escaneo y la interfaz web.
3. **Implementación**: Desarrollo de cada módulo en Python, asegurando su correcta integración con Flask y otras herramientas.
4. **Pruebas**: Se realizan pruebas en diferentes entornos de red para verificar la detección de tráfico sospechoso y escaneo de puertos.
5. **Despliegue**: Implementación del sistema en un entorno real o de prueba para su uso.
6. **Mantenimiento y Mejoras**: Se aplican actualizaciones y optimizaciones según sea necesario.

8. Desarrollo del Sistema

8.1 Módulo de Monitoreo de Tráfico

- Se implementa en `app.py`.
- Utiliza `Scapy` para capturar paquetes de red.
- Filtra paquetes para identificar tráfico sospechoso.
- Requiere la instalación de **Scapy** mediante `pip install scapy`.

8.2 Módulo de Escaneo de Puertos

- Se desarrolla en `scanner.py`.
- Usa **nmap** o **socket** para detectar puertos abiertos en un host.
- Proporciona información detallada de los puertos detectados.
- Se requiere instalar **python-nmap** mediante `pip install python-nmap` si se usa Nmap.

8.3 Módulo de Verificación de USB

- Se implementa en `usb.py`.
- Compara dispositivos USB conectados con una lista de dispositivos autorizados.
- Notifica si un dispositivo es considerado malicioso.
- Utiliza **PyUSB**, que se instala con `pip install pyusb`.

8.4 Módulo de Análisis de Archivos

- Permite la carga de archivos desde la interfaz web.
- Verifica si un archivo contiene malware o está corrupto.
- Proporciona un estatus de seguridad del archivo analizado.

8.5 Servidor Flask

- Se encuentra en `app.py`.
- Gestiona las solicitudes de la interfaz web y ejecuta los módulos de monitoreo y escaneo.
- Conecta los módulos del IDS con la interfaz gráfica para un uso más sencillo.
- Flask se instala con `pip install flask`.

9. Conclusión

Este sistema de Detección de Intrusos (IDS) representa una solución funcional y eficaz para la vigilancia de seguridad en redes. A través de la implementación de tecnologías como Python, Flask y Scapy, se ha logrado desarrollar un sistema accesible y flexible para la administración de la red, ofreciendo una interfaz web sencilla que facilita la interacción del usuario con el sistema. Python, con su amplio ecosistema de bibliotecas, ofrece la robustez necesaria para procesar y analizar el tráfico de red, mientras que Flask se encarga de crear una interfaz web ligera y fácil de usar, lo que permite a los administradores gestionar los eventos y alertas de manera eficiente.

Este documento y el sistema en sí están abiertos a futuras ampliaciones. Se pueden incorporar nuevas funcionalidades, como la integración con otros sistemas de seguridad o la mejora de algoritmos de detección mediante técnicas avanzadas como el análisis de comportamiento o el aprendizaje automático. Además, se pueden realizar mejoras para optimizar la precisión en la detección de intrusos, así como la eficiencia en el manejo de recursos del sistema.