Proyecto final

Curso de Sistemas Operativos y Laboratorio

Implementación de un clasificador Naive Bayes para detección de archivos maliciosos en un entorno de sistema operativo Linux

# Miembros del equipo

Nelson Puerta

William Torres

Clara Villadiego

# Resumen

Este proyecto propone la implementación de una herramienta basada en el algoritmo de clasificación **Naive Bayes**, que se encuentra en el capítulo 6 del libro *Machine Learning* de Tom Mitchell, para la detección de archivos maliciosos en sistemas operativos Linux. Para ello se recopilarán archivos benignos y maliciosos, se extraerán características relevantes, y se entrenará un modelo que pueda clasificar los archivos con base en su contenido. El enfoque de este proyecto busca ofrecer un marco práctico para la aplicación del aprendizaje automático, y fortalecer la seguridad de sistemas Linux mediante un método liviano, adaptable y de bajo costo, con potencial de uso tanto en entornos académicos como en escenarios profesionales de ciberseguridad.

# Introducción

El creciente número de amenazas dirigidas a sistemas operativos, especialmente a aquellos basados en Linux, ha generado la necesidad urgente de implementar mecanismos automáticos y efectivos para la detección de archivos maliciosos. Muchos de los métodos actuales dependen de firmas o reglas estáticas, lo que los hace vulnerables ante nuevas variantes de malware diseñadas para evadir dichas defensas.

En este contexto, donde la seguridad informática representa uno de los principales desafíos tecnológicos, resulta crucial desarrollar soluciones inteligentes capaces de adaptarse a amenazas emergentes. Dado que Linux es ampliamente utilizado en servidores, dispositivos IoT y entornos académicos, contar con herramientas eficientes de análisis que aprovechen técnicas de aprendizaje automático se vuelve fundamental para detectar amenazas desconocidas sin depender exclusivamente de firmas predefinidas, elevando así el nivel de protección del sistema operativo.

# Antecedentes o marco teórico

**Aspectos teóricos necesarios para comprender el desafío y llevarlo a cabo**

**Algoritmo Naive Bayes:** es un modelo de clasificación supervisado basado en el teorema de Bayes, el cual, calcula la probabilidad de que una instancia pertenezca a una clase determinada dado un conjunto de características. Su principal suposición es que todas las características son condicionalmente independientes entre sí dado el valor de la clase, lo que simplifica enormemente los cálculos probabilísticos.

**Extracción de características:** consiste en aplicar técnicas de minería de datos para identificar y representar atributos relevantes y discriminativos a partir del contenido de archivos. En el contexto de detección de *malware*, esto puede incluir análisis estático de archivos binarios o scripts, considerando elementos como el número y tipo de instrucciones, presencia de cadenas sospechosas, entropía del archivo, encabezados, permisos y más. Estas características extraídas son utilizadas como variables de entrada para entrenar el modelo de clasificación.

**Clasificación binaria:** es una tarea fundamental del aprendizaje automático supervisado, donde el objetivo es asignar una de dos posibles etiquetas a cada instancia del conjunto de datos. En este proyecto, la clasificación binaria se aplica para distinguir entre archivos benignos (no maliciosos) y archivos maliciosos (potencialmente dañinos). A partir de las características extraídas de los archivos, el modelo predice la probabilidad de que un archivo pertenezca a cada clase.

**Relación que tiene esta teoría con los temas del curso de Sistemas Operativos**

El proyecto permite analizar el comportamiento de un programa real respecto a múltiples componentes del sistema operativo:

* Planificación de procesos al medir tiempos de CPU.
* Gestión de memoria al evaluar estructuras y cargas de datos.
* Acceso a disco con archivos de entrenamiento/test.
* Conocimiento de estructuras de archivos en Linux y su análisis binario/textual.
* Estudio de atributos como permisos, usuarios propietarios y modos de ejecución.
* Automatización de tareas con scripts mediante el uso de *Bash* y herramientas del sistema (file, readelf, strings, grep) para el procesamiento.

# Objetivos

**Objetivo general**

Desarrollar e implementar en C una herramienta funcional basada en el algoritmo Naive Bayes para la clasificación de archivos benignos y maliciosos en sistemas operativos Linux, logrando un prototipo capaz de demostrar la viabilidad del enfoque propuesto.

**Objetivos específicos:**

* Recopilar y organizar al menos 200 archivos (100 benignos y 100 maliciosos) representativos del entorno Linux. Extraer características relevantes de estos archivos usando herramientas de sistema como *file*, *readelf*, *strings* y *grep*, almacenando los datos preprocesados en un formato adecuado para el entrenamiento del modelo.
* Desarrollar en C la lógica del algoritmo Naive Bayes, incluyendo las funciones para el cálculo de probabilidades. Entrenar el modelo con el dataset preprocesado, asegurando que el clasificador pueda distinguir entre archivos benignos y maliciosos y que sea capaz de guardar y cargar el modelo entrenado para su uso posterior.
* Crear una aplicación en C que permita al usuario seleccionar un archivo y clasificarlo usando el modelo entrenado. Realizar pruebas exhaustivas del prototipo con archivos de prueba, evaluando su precisión y rendimiento para identificar posibles mejoras y asegurar la estabilidad de la herramienta.

# Metodología

Este proyecto usará el lenguaje. Primero, en la Fase 1, reuniremos y prepararemos datos (archivos buenos y malos) usando comandos de Linux como *readelf* o *hexdump*. Luego, en la Fase 2, construiremos un programa clasificador con el algoritmo Naive Bayes en C, que aprenderá de esos datos. Finalmente, en la Fase 3, probaremos el programa, crearemos una forma sencilla de usarlo por comandos, y haremos los ajustes finales y la documentación.

**Herramientas**

* Lenguaje C
* Plataforma: Ubuntu Linux
* Comandos Linux: readelf, file, grep, stat, hexdump

**Actividades necesarias para cumplir los objetivos:**

Fase 1: Preparación de datos (Objetivo específico 1)

* Investigación y selección de datasets
* Recopilación de archivos (benignos/maliciosos)
* Extracción y preprocesamiento de características

Fase 2: Implementación del clasificador (objetivo específico 2)

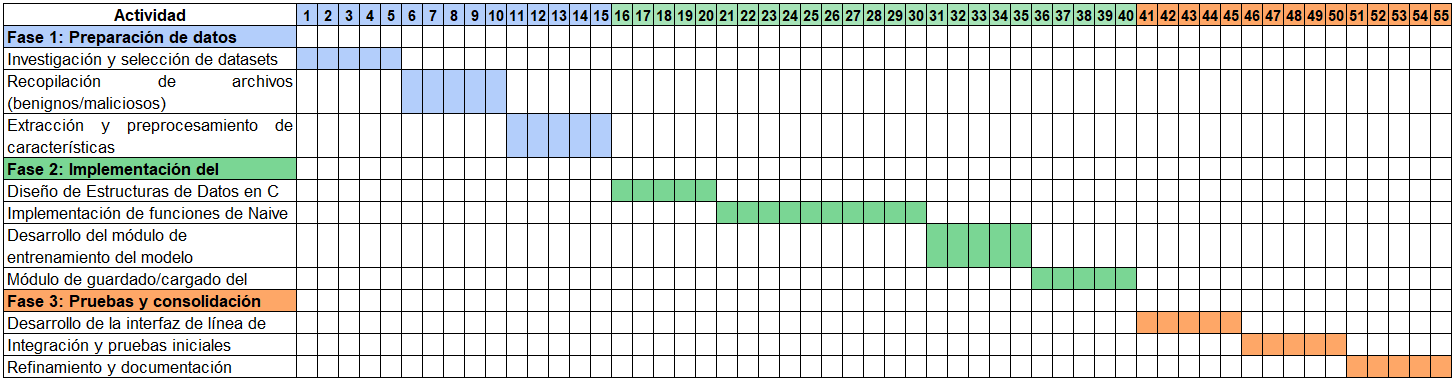
* Diseño de estructuras de datos en C
* Implementación de funciones de Naive Bayes
* Desarrollo del módulo de entrenamiento del modelo
* Módulo de guardado/cargado del modelo

Fase 3: Pruebas y consolidación (Objetivo específico 3)

* Desarrollo de la interfaz de línea de comandos
* Integración y pruebas iniciales
* Refinamiento y documentación

# Cronograma

Diagrama de Gantt para el desarrollo de las actividades:



# Referencias

* Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). Deep Learning for Android Malware Classification. Lecture Notes in Computer Science, 9978, 141-160.
* Referencia general sobre análisis de malware o herramientas en Linux: Welivesecurity. (2022, 5 enero). Herramientas de Linux para el análisis de malware. ESET Latinoamérica. Recuperado de <https://www.welivesecurity.com/la-es/2022/01/05/herramientas-de-linux-para-el-analisis-de-malware/>
* Diaz, J. (s.f.). Inteligencia Artificial para la detección de binarios maliciosos [Tesis de maestría no publicada]. Universitat Oberta de Catalunya. Recuperado de <https://openaccess.uoc.edu/bitstream/10609/138409/6/jdiaznavTFM0122memoria.pdf>
* Vinod, P. (2014). Linux Malware Detection Using eXten-ded–Symmetric Uncertainty. En R. S. Chakraborty, V. Matyas, & P. Schaumont (Eds.), Security, Privacy, and Applied Cryptography Engineering (SPACE 2014) (pp. 209-222). Springer, Cham.
* Mitchell, T. M. (1997). Machine learning. McGraw-Hill.