Formato Informe Técnico

Informe Técnico del Proyecto:

Inteligencia Artificial para la detección de malware

Información general

Fecha: viernes 08 de diciembre del 2023

Nombre del estudiante: Daniel Bustamante Lagart

Matricula del estudiante: ACM30299

Materia: Técnicas generales de ataques 1

Docente: César Antonio Ríos Olivares

Fecha de entrega: lunes 11 de diciembre del 2023

Resumen del proyecto: Como autor de este proyecto, mi compromiso radica en explorar las fronteras de la ciberseguridad mediante el desarrollo de sistemas de Inteligencia Artificial especializados en el análisis de malware.

Índice

1. Objetivo del proyecto

En este proyecto vamos a investigar cómo la convergencia entre la Inteligencia Artificial y el análisis de malware no solo fortalece las defensas digitales, sino que también proporciona respuestas más rápidas y efectivas ante amenazas persistentes y evolutivas en la sociedad digital, también crearemos nuestra inteligencia artificial para demostrar el impacto.

Los algoritmos seleccionados de Inteligencia Artificial según los requisitos del proyecto son las redes bayesianas por lo que demostraremos porque no son las adecuadas para el análisis de malware, sin embargo, usaremos un algoritmo que si promete mucho para esta tarea (Bosques Aleatorios). Compararemos los resultados y usaremos el que mejor haga su trabajo.

2. Material, Equipo o Software a utilizar

Para este proyecto se utilizarán los siguientes materiales, equipo o software:

- Maquina Windows
- Anaconda versión 2023.09
- Visual Studio code 1.85.0
- Malware LagarCry.exe (Visto en la materia Control y gestión de la información)
- Instalador de Zoom o cualquier software benigno
- Google chrome

3. Actividades previas de investigación

Antes de comenzar el proyecto, se realizaron las siguientes actividades de investigación:

3.1. MALWARE Y SUS CATEGORÍAS

Malware: El malware es un software intruso que está diseñado deliberadamente para provocar daños en equipos y sistemas informáticos.

Categorías.

Ransomware: Es un tipo específico de malware que se encarga de evitar que el usuario tenga acceso a su propia información (documentos, videos, imágenes, archivos) a través de la encriptación de dichos archivos.

Spyware: El atacante o hacker utiliza este tipo de malware para monitorear la actividad de los usuarios en internet. A través de un spyware es posible recolectar información sensible

Gusanos: Los gusanos informáticos tienen el objetivo de replicarse. Es decir, al infectar un dispositivo buscan pasar al siguiente, ya sea para dañar los sistemas informáticos o para robar información de los usuarios.

Troyanos: Este software malicioso le debe su nombre al famoso relato del caballo de Troya, ya que se trata de un virus que se esconde en un software que aparenta ser confiable.

Adware: El adware tiene la tarea de bombardear nuestro dispositivo con anuncios publicitarios no deseados.

Botnets: También llamados redes de robots, equipos o de códigos, los botnets son aquellos que se encargan de ejecutar un malware de forma simultánea utilizando varias computadoras.

Rootkits: Los Rootkits son utilizados para suplantar los permisos y controles de administrador con la finalidad de tomar el control de un sistema operativo por completo sin levantar ningún tipo de alerta del antivirus o software antimalware.

3.2. TÉCNICAS DE DETECCIÓN DE MALWARE

Detección de firmas: Es el enfoque clásico en el que se basaron los productos antivirus cuando surgieron. Funciona comparando los archivos almacenados o descargados en el equipo contra una lista de códigos maliciosos conocidos.

Detección de patrones / cadenas: Es una técnica utilizada en ciberseguridad para identificar software malicioso (malware) basándose en la identificación de patrones específicos en el código o el comportamiento del programa.

Análisis heurístico: El análisis heurístico es un método de detección de virus mediante el cual se examina el código en busca de propiedades sospechosas. Se diseñó para detectar virus nuevos desconocidos y versiones modificadas de las amenazas existentes (Usa algoritmos de IA para detectar anomalías).

3.3 TÉCNICAS AVANZADAS DE OCULTAMIENTO QUE USA EL MALWARE

Extensiones VBE: Es un formato de archivo que permite incrustar scripts de Visual Basic for Applications (VBA) en documentos de Microsoft Office. Al utilizar esta técnica, el malware puede ocultar su código malicioso dentro de documentos legítimos y aprovechar la ejecución de scripts permitida por las aplicaciones de Office.

PowerShell: PowerShell es una interfaz de línea de comandos y un lenguaje de scripting desarrollado por Microsoft. Los atacantes a menudo utilizan scripts de PowerShell para ejecutar comandos maliciosos en sistemas comprometidos debido a su versatilidad y capacidad para evadir las soluciones de seguridad tradicionales.

Redes Tor/I2p: Tor (The Onion Router) e I2p (Invisible Internet Project) son redes diseñadas para proporcionar anonimato y privacidad en línea. Algunos malware utilizan estas redes para ocultar la comunicación entre el sistema infectado y los servidores de comando y control.

Técnicas mixtas (twitter, facebook, github): Algunos malware utilizan plataformas de redes sociales como Twitter, Facebook o GitHub para alojar o recibir comandos maliciosos. Estos comandos pueden ser encriptados o camuflados en el tráfico normal de estas plataformas.

404: Algunos malware utilizan respuestas HTTP 404 (páginas no encontradas) para ocultar la comunicación con servidores de comando y control. En lugar de enviar

solicitudes directas, el malware puede utilizar técnicas que parezcan solicitudes de error 404 para comunicarse.

Análisis del entorno: Algunos malware realizan análisis del entorno antes de ejecutar sus cargas útiles. Esto implica verificar variables del sistema, configuraciones y entorno de ejecución para adaptarse y evitar la detección.

WMI: WMI es una infraestructura de gestión de Windows que permite a los administradores realizar tareas de administración en sistemas locales y remotos. Los atacantes pueden utilizar WMI para ejecutar comandos maliciosos de forma remota, lo que dificulta la detección.

3.4 ANÁLISIS DE MALWARE Y REVERSING

Análisis estático: El análisis estático de malwares es un conjunto de técnicas que permiten estudiar, prever y observar el funcionamiento de este tipo de softwares sin necesidad de ejecutarlos. El análisis se hace por medio de la revisión del código fuente del archivo y la identificación de elementos maliciosos en el mismo. De este modo, un analista puede hacerse una idea de las tareas que ejecuta el malware en un sistema sin correr el riesgo de infectar el ordenador con este.

Análisis dinámico: Los análisis dinámicos son realizados en máquinas virtuales preparadas para ejecutar malwares dañinos y demostrar su funcionamiento en tiempo real. Se diferencian del análisis estático de malwares en que en los análisis dinámicos se ejecuta el software malicioso y se le permite que ejecute tareas dañinas en un entorno virtual controlado.

Ingeniería inversa: La ingeniería inversa es un proceso por medio del cual los analistas de malware ejecutan el software con su código fuente ensamblado, para poder ver el paso a paso de cada una de sus tareas. Este método combina las dos fases anteriores y ofrece un análisis detallado acerca del funcionamiento del virus.

3.5 INRELIGENCIA ARTIFICIAL

Machine learning: El aprendizaje automático, aprendizaje automatizado, aprendizaje de máquinas o aprendizaje computacional es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial

Redes bayesianas: los modelos de Naive Bayes son una clase especial de algoritmos de clasificación de Aprendizaje Automático, o Machine Learning, tal y como nos referiremos de ahora en adelante. Se basan en una técnica de clasificación estadística llamada “teorema de Bayes”.

Bosques aleatorios: Un random forest (o bosque aleatorio en español) es una técnica de Machine Learning muy popular entre los Data Scientist. Es un conjunto de árboles de decisión que trabajan juntos. Cada árbol vota por la clase, y la clase con más votos se elige.

3.6 EVALUACIÓN DE ALGORITMOS

Los algoritmos de redes bayesianas no tienen buen rendimiento para el análisis de malware, mientras que las Bosques aleatorios y los algoritmos de vecinos más cercanos son los mejores para estas tareas, a continuación, mostrare las siguientes tablas que comprueban el hecho.

Modelo	Exactitud	Precisión	Sensibilidad	Medida-F1	Media CV	Tiempo de entrenamiento (s)
Regresión logística	0.9992	0.9991	0.9994	0.9992	0.9992	0.6570
KNN con K=1	**0.9999**	**1.0000**	**0.9999**	**0.9999**	0.9998	0.0140
KNN con K=3	0.9998	0.9999	0.9997	0.9998	0.9997	0.0140
KNN con K=5	0.9998	0.9999	0.9996	0.9998	0.9996	0.0140
KNN con K=7	0.9997	0.9999	0.9994	0.9997	0.9996	0.0140
KNN con K=9	0.9996	0.9999	0.9993	0.9996	0.9996	0.0150
Árbol de decisión	0.9998	0.9998	0.9998	0.9998	0.9999	0.2680
Bosques aleatorios	**0.9999**	**1.0000**	**0.9999**	**0.9999**	**1.0000**	5.7740
Naive Bayes	0.9969	0.9954	0.9985	0.9969	0.9968	0.0490
SVM	0.9996	0.9995	0.9997	0.9996	0.9996	6.6873
MLP	0.9999	0.9999	0.9998	0.9999	0.9998	7.5092

Imagen1. Describe la exactitud de algunos algoritmos de Inteligencia artificial para el análisis de malware.

Modelo	VP	FP	VN	FN
Regresión logística	17450	16	17348	11
KNN con K=1	17457	0	17364	4
KNN con K=3	17456	1	17363	5
KNN con K=5	17454	1	17363	7
KNN con K=7	17451	1	17363	10
KNN con K=9	17449	1	17363	12
Árbol de decisión	17458	3	17361	3
Bosques aleatorios	17459	0	17364	2
Naive Bayes	17435	81	17283	26
SVM	17456	8	17356	5
MLP	17459	0	17364	2

Imagen2. Describe los falsos y verdaderos positivos de los algoritmos al detectar malware.

Podemos apreciar que los algoritmos de Bosques aleatorios son los mejores para el análisis de malware, estos algoritmos fueron alimentados con el mismo dataset.

3.7 COMPOSICIÓN DE UN ARCHIVO .EXE (Encabezados PE)

Los encabezados PE (Portable Ejecutable) son una parte fundamental de los archivos ejecutables en el formato PE, que es utilizado por sistemas operativos como Windows.

Encabezado DOS (código auxiliar de MS-DOS):

- Contiene información para sistemas MS-DOS.
- Permite la ejecución del programa en sistemas MS-DOS antiguos.

Firma PE:

- Identifique el archivo como un PE ejecutable.
- Es un valor constante que indica el inicio del encabezado PE.

Encabezado COFF (formato de archivo de objeto común):

- Contiene información sobre la arquitectura del procesador, la sección de código, la sección de datos, entre otros.
- Especifica el tamaño de las secciones y otros detalles esenciales.

Encabezado opcional:

- Proporciona información adicional sobre la ejecución del programa.
- Incluye detalles como la dirección base preferida de carga, el tamaño de la imagen, la alineación de secciones, etc.

Encabezados de sección:

- Describe cada sección del programa (por ejemplo, .text para código ejecutable, .data para datos inicializados, .rdata para datos de solo lectura, etc.).
- Incluye información sobre la dirección virtual, el tamaño, el puntero de archivo, etc.

Directorios de datos:

- Contiene direcciones de tablas importantes, como la tabla de importación/exportación, la tabla de direcciones de funciones (IAT), entre otras.

Tabla de importación:

- Enumere las funciones y bibliotecas externas que utiliza el programa.
- Incluye información sobre cómo localizar y enlazar estas funciones durante la ejecución.

Tabla de exportación:

- Contiene información sobre las funciones y datos que el programa proporciona para que otros programas los utilicen.

Tabla de recursos:

- Almacena recursos como iconos, cuadros de diálogo, etc.
- Facilita la internacionalización y personalización del programa.

3.8 CONCLUSIÓN

la investigación aborda de manera rápida algunos temas que investigamos para poder realizar la herramienta, entender estos conceptos facilitó la elaboración y sobre todo una mayor visión de cómo debía funcionar cada parte de ella. Desde la comprensión de sus categorías y técnicas de ocultamiento hasta la aplicación de métodos avanzados, incluyendo la inteligencia artificial fue fundamental entenderlo para avanzar con la programación. Esta información es crucial para el desarrollo de herramientas más efectivas y proactivas en la lucha contra las amenazas cibernéticas.

4. Instrucciones precisas a realizar

Para llevar a cabo este proyecto, se deben seguir las siguientes instrucciones.

1.-Instalar Python 3.12.1 (https://www.python.org/downloads/).

2.-Instalar Visual Studio Code (https://code.visualstudio.com/download).

3.-Instalar anaconda (https://www.anaconda.com/download).

4.-Seguir el manual de sistema para desarrollar la herramienta.

5.-Seguir el manual de usuario para usar la herramienta.

5. Resultados y conclusiones esperadas del proyecto

Los resultados y conclusiones esperadas de este proyecto son:

- Detectar lagartcry como malware
- Detectar zoom Installer como software benigno
- Demostrar el potente funcionamiento de la Inteligencia Artificial en la ciberseguridad
- Ver la efectividad de los bosques aleatorios en contra de las redes bayesianas

6. Producto final esperado

El producto final esperado de este proyecto es:

Una herramienta fácil de usar para el usuario que combine la Inteligencia artificial y la ciberseguridad para detectar malware con pocos falsos positivos.

7. Documentos Entregables

- Manual de usuario de proyecto
- Manual de sistema del proyecto
- Manual técnico del proyecto
- Presentación empresarial de la herramienta

8. Bibliografía

Staff, F. (2022, August 2). México registra 80,000 millones de intentos de ciberataques en 2022. Forbes México. https://www.forbes.com.mx/mexico-registra-80000-millones-de-intentos-de-ciberataques-en-2022/

Ehrhardt, M. (2023, January 18). Ciberataques e inflación, lo que más preocupa alas empresas. dw.com. https://www.dw.com/es/ciberataques-e-inflaci%C3%B3n-lo-que-m%C3%A1s-preocupa-a-las-empresas/a-64430159

Jiménez, M. M. (n.d.). Casos de ciberataques a empresas en 2022. https://www.piranirisk.com/es/blog/ciberataques-empresas-en-2022

7 formas en las que tus dispositivos se pueden infectar con malware. (2021, January 5). https://www.welivesecurity.com/la-es/2021/01/05/formas-comunes-dispositivos-pueden-infectarse-con-malware/

Software anti-malware frente a software antivirus: ¿Qué diferencia hay? (2023, April 19). www.kaspersky.es. https://www.kaspersky.es/resource-center/preemptive-safety/malware-remover-vs-antivirus-software

Malwarebytes. (2020, June 3). ¿Es lo mismo antimalware y protección antivirus? Malwarebytes. https://es.malwarebytes.com/antivirus/

Sistemas EDR: qué son y cómo ayudan a proteger la seguridad de tu empresa | Empresas | INCIBE. (n.d.). https://www.incibe.es/empresas/blog/sistemas-edr-son-y-ayudan-proteger-seguridad-tu-empresa

¿Qué es XDR? (n.d.). Trend Micro. https://www.trendmicro.com/es_mx/what-is/xdr.html

KeepCoding, R. (2022, August 23). Análisis estático de malwares | KeepCoding Bootcamps. KeepCoding Bootcamps. https://keepcoding.io/blog/analisis-estatico-de-malwares/

¿En qué consiste el análisis heurístico? (2023, August 18). www.kaspersky.es. https://www.kaspersky.es/resource-center/definitions/heuristic-analysis

Chavez, J. J. S. (2023, October 3). ¿Qué es un malware? Tipos y cómo funcionan. https://deltaprotect.com/blog/tipos-de-malware

Técnicas avanzadas de ocultamiento de malware. (2017, March 7). [Slide show]. PPT. https://es.slideshare.net/dfpluc/tecnicas-avanzadas-de-ocultamiento-de-malware

PE Explorer: EXE File Editor, DLL View Scan Tool for 32-bit Windows PE files. (n.d.). http://www.pe-explorer.com/

Karthikapadmanaban. (2023, February 15). Malware Detection using Random Forest. Kaggle. https://www.kaggle.com/code/karthikapadmanaban/malware-detection-using-random-forest/input

Samuel Mouro González (A Coruña, septiembre de 2022.). Técnicas de aprendizaje máquina para análisis de malware. https://ruc.udc.es/dspace/bitstream/handle/2183/32112/MouroGonzalez_Samuel_TFG_2022.pdf?sequence=3