UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3094 - SECURITY DATA SCIENCE

Sección 10

Ing. Jorge Yass

Proyecto: Fase 3

Carlos Alberto Raxtum Ramos, 19721
Javier Alejandro Cotto Argueta, 19324
Juan Manuel Marroquin Alfaro, 19845
Jose Abraham Gutierrez Corado, 19111
Walter Danilo Saldaña Salguero, 19897

GUATEMALA, 30 de mayo de 2023

I. Introducción

Actualmente a pesar de tener bastantes medios de comunicación de forma virtual, como las redes sociales, whatsapp, videollamadas nunca se ha dejado de usar el correo electrónico, el cual representa más del 50% de la población mundial utiliza el canal de comunicación según el artículo de 99Firms además que el departamento de marketing son unos de los principales usuarios del correo electrónico según el estudio realizado por Kinsta. Basado en estos datos, se nos ocurrió estudiar uno de los malwares que más desapercibidos navegan por la red y de forma sigilosa incrustados en archivos adjuntos en correos electrónicos, *stegomalware*. Este malware pasa desapercibido en la mayoría de algoritmos de detección debido a que se encuentra dentro de la estructura interna del archivo o imagen mediante esteganografía, una técnica de cifrado para ocultar mensajes secretos en la estructura de los archivos.

Cuando se trata de imagenes, existen tres tipos principales de stego malware, los cuales son:

1. LSB (Least Significant Bit) modifica el bit menos significativo de cada píxel de una imagen para ocultar el mensaje secreto.
2. S-UNIWARD (Universal Wavelet Relative Distortion) es un algoritmo esteganográfico diseñado para ocultar mensajes en imágenes digitales modificando los coeficientes de las transformadas wavelet.
3. WOW (With Or Without) es una técnica que consiste en modificar la intensidad de los píxeles de la imagen de forma que los cambios sean imperceptibles para el ojo humano. El nombre "con o sin" viene del hecho de que el mensaje puede ocultarse con o sin modificar las intensidades de los píxeles, en función del nivel de seguridad deseado.

Con los métodos más populares definidos, nos hemos enfocado en el análisis de imágenes contaminadas por S-UNIWARD y las preguntas claves que surgieron en la primera fase del proyecto han cambiado con respecto a las actuales, pues llegamos a unas más concretas conforme se iba realizando, las preguntas clave son:

- ¿Existen posibles indicadores que desencadenan FN o FP en la predicción a los que nos podamos anticipar para preprocesar los datos?
- Si existen,¿Cuáles serían los indicios que pueden indicar la presencia de un stegomalware en un documento ?

- ¿Qué impacto tiene un stegomalware?

La literatura consultada para la realización del proyecto está enfocada en como se llega a generar este malware en archivos y los diferentes tipos para saber cómo analizarlo:

- Este es un estudio realizado en la detección de stegomalware en imagenes: Stegomalware Study.
- La invisibilización de stegomalware y sus diferentes tipos: Invisibilidad del stegomalware y sus tipos
- El cómo crear stegomalware fácilmente: Crear stegomalware

Por último para este estudio se utilizaron tres modelos de clasificación de imágenes los cuales serán comparados a lo largo del artículo para saber cual sería el mejor a implementar para detectar el stegomalware.

II. Métodos

Para la fuente de datos utilizamos la colección de imágenes llamada "Steganalysis.tar.gz" proveniente de **Image dataset (Paper: Paired mini-batch training)**. Esta contiene más de 30 mil imágenes "limpias" (en otras palabras que no contienen stego malware) y también contiene las mismas imágenes (visualmente hablando) pero con stego malware tipo S-UNIWARD (Universal Wavelet Relative Distortion).

El procedimiento de balanceo de datos no tuvo que realizarse, debido a que los conjuntos de datos que encontremos tenían la misma cantidad de datos entre los dos conjuntos(imágenes normales y alteradas). Dentro de la exploración de datos, debido a que estos son pixeles de las imágenes, no se tuvo que hacer una ingeniería de características para obtener las columnas más importantes y encontrar la correlación entre estas.

Antes de la elección de los algoritmos a utilizar para el estudio, se decidió el obtener una muestra debido a la cantidad total de datos que se tenía, de esta forma se podría empezar la división del conjunto en varios conjuntos(entrenamiento, prueba y validación) y así empezar con los 3 modelos escogidos:

- Modelo Naive Bayes
  Dentro de los modelos más simples para la clasificación de conjuntos de datos se encuentra este. La elección de este surgió a partir que los datos que se tienen son pixeles lo cual son independientes de uno con el otro, por lo tanto evita la relación entre ellos y el cómo es el resultado final de unirlos y generar una imagen. Como métricas de evaluación del modelo se

obtuvieron:
Accuracy: 0.49
F1 Score: 0.65
Precision: 0.99
Recall: 0.49

Las cuales se comentarán más adelante.

- Gradient boost classifier

- SVM Model

III. Discusión

Los resultados obtenidos en el presente no lograron los objetivos de la investigación, sobre lograr detectar stego-malware en una imágen PNG. Hubo varios factores que influyeron en el bajo rendimiento de los modelos, de los cuales podemos destacar los siguientes: algoritmo utilizado para generar el stego-malware, la elección del modelo, datos de entrenamiento.

En cuanto al algoritmo para la generación de stego-malware, se profundizó sobre varios algoritmos, siendo estos: LSB (Least Significant Bit), S-UNIWARD (Universal Wavelet Relative Distortion), WOW (With Or Without). De estos, se encontró que hay diferencias significativas entre la implementación de la stegografía utilizada, y por lo tanto a fines de esta investigación se decidió optar por utilizar solo un algoritmo para delimitar mejor las variables, y en este caso se utilizó el S-Uniward ya que este por las distorsiones que realiza a la imágen, sería más fácil de poder reconocerlo con un modelo de clasificación de imágenes, sin embargo, como se profundizará a continuación, no se puede probar esta hipótesis dado que el modelo seleccionado no fue el más eficiente en llevar a cabo esta tarea,.

La siguiente variable influyente en los resultados de la investigación, es sobre el modelo utilizado. Se eligieron 3 modelos para comparar sus métricas y utilizar el que mejor desempeño tuviese. Como se ve en la tabla de resultados, el de mejor desempeño fue el naive bayes. Lo que diferencia a naive bayes de los otros modelos es que presupone que cada característica dentro del set de datos de entrenamiento, es independiente entre sí. La lógica de utilizar este modelo fue que dado que el algoritmo de stegografía seleccionado distorsiona la imágen, si consideramos cada pixel como dependiente de su contorno (pixeles alrededor), entonces el modelo no podría diferenciar la distorsión de una imagen normal. Entiéndase esto con el siguiente ejemplo, si tenemos una imágen con una línea recta, y la distorsionamos podríamos obtener una imágen de una línea un poco torcida, entonces el modelo no tiene forma de saber que la imágen original era una línea recta, y por lo tanto estaría muy sesgado al set de entrenamiento. La hipótesis fue que la presunción de independencia de bayes permitiría no cometer dicho error, sino que, ya que la distorsión

sucede en pixeles particulares que no dependen significativamente de la imágen original, el modelo pudiera encontrar la presencia de esos pixeles. Sin embargo, la hipótesis no se cumplió, pues la métrica del modelo fue muy baja. Una de las razones por la que pudo suceder es por el dataset de entrenamiento utilizado, lo cual nos lleva al siguiente punto.

La tercera variable con relación al cumplimiento de los objetivos de la investigación es el conjunto de datos de entrenamiento. Conseguir stego-malware es bastante complicado por su escasa disponibilidad. Para esta investigación la data utilizada fue considerablemente pobre, con 500 imágenes limpias y 500 imágenes con stego-malware incrustado. Se sabe que dicha cantidad de data no es suficiente para entrenar eficazmente un modelo, más cuando se trata de clasificar características demasiado sutiles como lo es el stego.

Entonces, las hipótesis planteadas para esta investigación no se logran cumplir por las razones expuestas: algoritmo de stegografía, modelo seleccionado y por el conjunto de datos utilizados.

IV. Resultados

SVM model

```
Accuracy: 0.23865546218487396
F1 Score: 0.23953431195224795
Precision 0.20068027210884354
Recall 0.21299638989169675
```

Imagen 1 SVM model

Gradient boost classifier

```
Accuracy: 0.23697478991596638
F1 Score: 0.23824367884626138
Precision 0.2755102040816326
Recall 0.2515527950310559
```

Imagen 2 Resultados Gradient Boost Classifier

Naive Bayes

```
Accuracy: 0.4907563025210084
F1 Score: 0.6561859918713822
Precision 0.9931972789115646
Recall 0.4924114671163575
```

Imagen 3 Naive Bayes

V. Conclusiones y recomendaciones

- De los modelos probados, naive bayes fue el que mejores métricas obtuvo, se intuye que la presunción de independencia entre datos que lo caracteriza pudo encontrar hallazgos en el archivo analizado.
- El método usado para generar el stego-malware influye en el entrenamiento y rendimiento del modelo, pues las características difieren.

Para futuras investigaciones se recomienda:

- Realizar ingeniería de características más allá de los bits de la imágen para encontrar detalles que faciliten el entrenamiento del modelo. Por ejemplo, analizar el header del archivo, ya que por no ser visible en la imágen, suele ser una técnica para ocultar stego-malware.
- Realizar un modelo con redes neuronales, pues suelen ser más eficientes para el análisis de imágenes, sin embargo, es de investigar cómo será su rendimiento en otro tipo de documentos como PDF.
- Conseguir un conjunto de datos más amplios, y por la escasez también se puede probar auto-generando muestras propias.