1 Objetivos

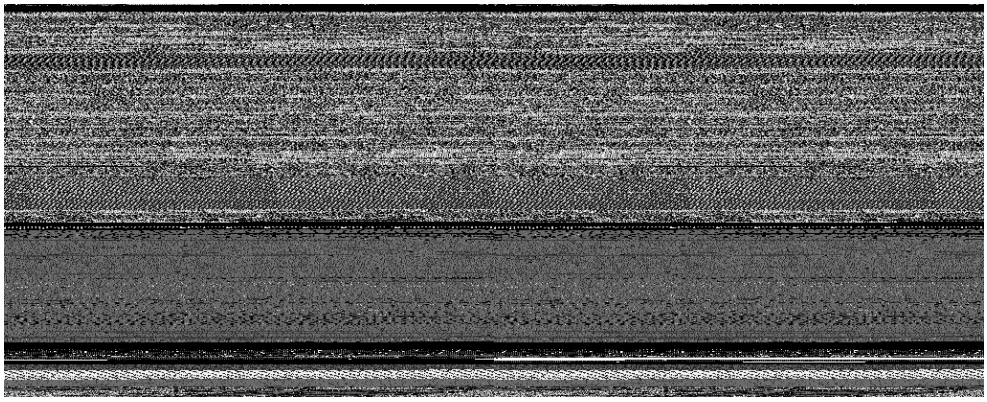
- Construir un modelo de DL que utilice imágenes de malware para la clasificación de familias
- Investigar sobre los ataques de evasión, inferencia, extracción y envenenamiento
- Utilizar el framework Adversarial Robustness ToolBox para atacar modelos de ML y DL

2 Preámbulo

La clasificación de malware es una tarea que involucra diversos retos. Cada tipo de análisis (estático y dinámico) tiene ventajas y limitaciones. Del lado del análisis estático la velocidad de análisis es su principal característica y su mayor reto consiste en diferenciar las llamadas a las DLLs y APIs sospechosas, de llamadas benignas. En el análisis dinámico la principal ventaja es registrar el comportamiento del malware con todo detalle, pero la preparación del entorno y ejecución requiere una inversión de tiempo y recursos técnicos considerables.

Las redes neuronales convolucionales se usan normalmente en la clasificación de imágenes, y de aquí surge la idea: ¿qué sucede si los bytes de un malware se pasan a una imagen? (artículo “Malware Images: Visualization and Automatic Classification”).

Las siguientes imágenes son dos ejemplares distintos de malware que pertenecen a la familia Adialer.C:



Los hashes de estos ejemplares son:

- 00bb6b6a7be5402fcfce453630bfff19
- 000bde2e9a94ba41c0c111ffd80647c2

Podemos observar a simple vista que las imágenes son bastante similares, y podemos determinar que ambos pertenecen a la misma familia. Entonces podemos considerar convertir los ejemplares de malware a imágenes y entrenar a la red neuronal con estas para clasificarlos.

Seguridad en modelos de data science

Los modelos de machine learning y deep learning son activos que están sujetos a los ciberataques, como cualquier otro activo digital.

Los ataques varían según su propósito y clasificación. En los ataques de caja negra, el adversario no conoce los detalles de implementación del modelo, en tanto que, en los ataques de caja blanca, el adversario si conoce los detalles.

Ataques de extracción

Las empresas que utilizan ML/DL para apoyar sus procesos de negocio invierten una gran cantidad de recursos en la investigación, desarrollo e implementación de sus modelos, y luego ofrecen un servicio pagado de clasificación a través de una API, por ejemplo.

Con esta información, se puede realizar un ataque de caja negra/blanca que consiste en utilizar un dataset y obtener la clasificación y confidencia a través de la API. Aun si utilizar la API tiene un costo, este será mínimo en comparación con los resultados. La idea es obtener las etiquetas y confidencia para cada una de las observaciones, y con ello, ¡entrenar un modelo propio! (Ataque de extracción).

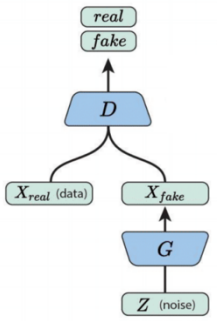
Dado que el nuevo modelo se entrenará en la forma en que el modelo objetivo clasifica, este tendrá resultados muy similares, sin invertir la gran cantidad de recursos que el modelo original.

Ataques de inferencia

Muchos modelos son entrenados con una combinación de datasets públicos y privados. Un atacante puede crear un modelo que permita saber si un registro fue utilizado como parte del entrenamiento (Membership). Un atacante también puede inferir data de un modelo a partir de indicar la clase buscada.

Ataques de evasión

Generative Adversarial Networks (GANs) son una forma de construir un modelo generativo al tener dos redes neuronales compitiendo una contra otra.



Una red toma el papel del generador (**G**), que convierte ruido aleatorio en imitaciones de data, intentando engañar al discriminador.

La otra red toma el papel del discriminador (**D**), que trata de distinguir data real de data falsa creada por el generador. Esto se puede aprovechar para realizar ataques con data que engañen a los modelos de clasificación.

Ataque de envenenamiento

Se aprovecha de la debilidad del entrenamiento federado, pues los nodos locales no siempre toman medidas de seguridad para asegurar la confiabilidad de sus fuentes de datos. La versión más peligrosa de este ataque es un backdoor, pues confunde al modelo únicamente para un patrón específico, y es muy difícil de detectar.

3 Desarrollo

El laboratorio consiste en dos partes. En la primera parte, se deberá desarrollar un modelo de Deep Learning que clasifique el malware en base a su imagen. En la segunda parte, se implementarán dos ataques contra el modelo anterior.

Primera parte

Se utilizará el dataset proporcionado en Canvas (maligm_dataset.zip), que contiene imágenes en formato .PNG de 25 familias distintas de malware. Debe realizarse el pre-procesamiento especialmente en el conteo de observaciones por familia. Considere prescindir de familias que tengan pocas observaciones. Plotee las imágenes de malware.

Tip: Utilice el jupyter proporcionado de guía para separa las familias de malware.

Luego construya una red neuronal con las capas/función de activación/optimizador que considere convenientes. Trabaje con 70% entrenamiento y un 30% pruebas, con el número de épocas que

considere conveniente (siempre que no ocurra sobreajuste). Muestra las métricas del modelo. Guarde su modelo.

Segunda parte

Implemente dos ataques (de diferente categoría), utilizando el framework Adversarial Robustness ToolBox, originalmente desarrollador por IBM, y donado recientemente a The Linux Foundation.

https://adversarial-robustness-toolbox.org/

Este framework contiene módulos de ataque y defensa, métricas, etc; y soporta framewroks como TensorFlow, Keras, Scikit-Learn, PyTorch, etc., todo tipo de data (imágenes, tablas, video, etc.) y tareas de machine learning (clasificación, generación, etc.)

El modelo víctima del ataque será el modelo desarrollado en la primera parte.

Sugerencia: instalar el ART framework y probar los ejemplos vistos en clase, antes de realizar los ataques sobre el modelo víctima, para asegurar que la herramienta fue instalada correctamente y que funciona sin problemas.

4 Calificación

- Se debe entregar el link al repositorio en Github del laboratorio que debe incluir:
    - Jupyter Notebook: desarrollo del modelo, explicación de los ataques elegidos, evidencia de los pasos realizados y prueba del ataque contra el modelo.
- La fecha de entrega será el domingo **5 de mayo a las 23:59 horas.**
- Plagio parcial o total anula el proyecto, y se elevará el caso a la Dirección para las sanciones administrativas.
- Rúbrica
    - Desarrollo del modelo: 30%
    - Ataque 1: 35% (explicación del ataque, 15%, implementación 20%)
    - Ataque 2: 35% (explicación del ataque, 15%, implementación 20%)