University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING

Department of Electrical and Computer Engineering & Computer Science

DSCI 6015
AI and Cybersecurity
Spring 2024

Instructor: **Dr. Vahid Behzadan**

Name: OMKAR ASHISH PEDNEKAR

Student ID: 00768475

Simplified Midterm

Abstract:

This report outlines the creation of a cloud-based API for detecting malicious PE files, achieved through AWS Sagemaker. The API employs a Random Forest binary classifier trained on a labeled dataset of binary feature vectors to categorize PE files as either malicious or benign.

Utilizing AWS Sagemaker, the project covered all stages from model construction and training to deployment, consolidating these tasks within the same Amazon Sagemaker instance. A user-friendly web application was developed to allow remote users to upload their executable (.exe) files and assess potential threats. Python served as the primary programming language for the project, with the inclusion of various ML libraries such as sklearn, pefile, nltk, among others, for model development and execution.

Prologue:

Random Forest Classifier:

The Random Forest classifier is a highly adaptable and potent machine learning technique utilized for both classification and regression tasks. Falling under the ensemble learning category, it amalgamates multiple individual models to generate predictions. In the context of Random Forests, these individual models take the form of decision trees, and the "forest" denotes a compilation of these trees, each constructed independently and making decisions based on input features.

The fundamental concept behind Random Forests involves injecting randomness into both the training process and prediction phase to produce a diverse set of decision trees. During training, each tree is crafted using a random subset of the training data and a random subset of features for each split. This randomness aids in diversifying the individual trees, mitigating the risk of overfitting, and enhancing the overall model performance. During prediction, the output of each tree is amalgamated through averaging or voting, with the most frequent class (for classification) or the average (for regression) serving as the final prediction.

A significant advantage of Random Forests is their proficiency in handling high-dimensional data with numerous features. They demonstrate robustness to noise and outliers in the data and necessitate minimal hyperparameter tuning compared to more intricate models. Moreover, Random Forests offer insights into feature importance, enabling users to discern which features exert the greatest influence on the model's predictions.

Overall, Random Forests find extensive application across diverse domains like finance, healthcare, and bioinformatics and now, by the use case of this project, for cybersecurity case studies as well!

Implementation:

First you can get the training model used as a Random Forest Classifier along with the codebase and the sample dataset (including all benign and malicious software) in this link –

https://storage.googleapis.com/aiec-s24/4-%20Training%20a%20Static%20Malware%20Detector.zip

DEPLOYMENT

Now that we have our model ready, we deploy it using an endpoint created on AWS Sagemaker to handle input data which will be provided by our python client. These files are then compressed into a model archive which is then later uploaded to an S3 bucket for deployment purposes.

CLIENT

The final stage of deployment is to create a python client for interaction with the model and then invoke a backend for the web application client. This client will fetch the data from uploaded PE executable and then send the features gathered from the executable to the Sagemaker endpoint for inference. Based on the model, the client shall output whether the file is malicious or benign.

Here are the steps of demonstration of user-interaction with the python client –

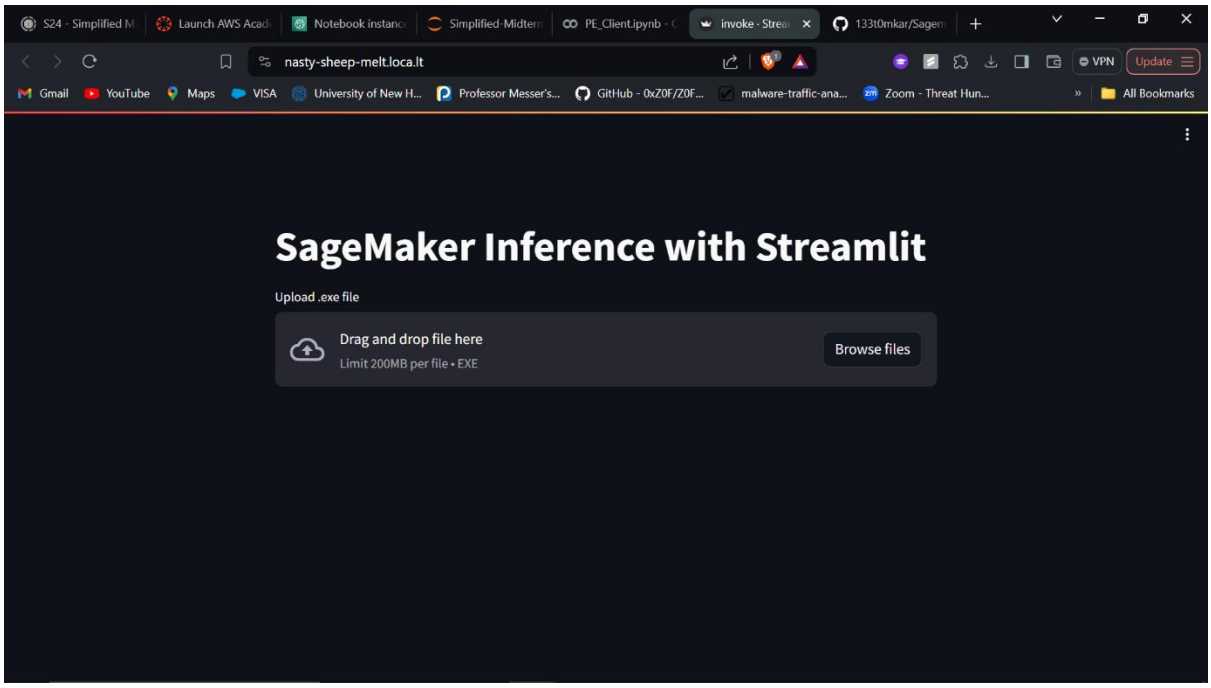

Fig 1: Python Client using AWS Sagemaker

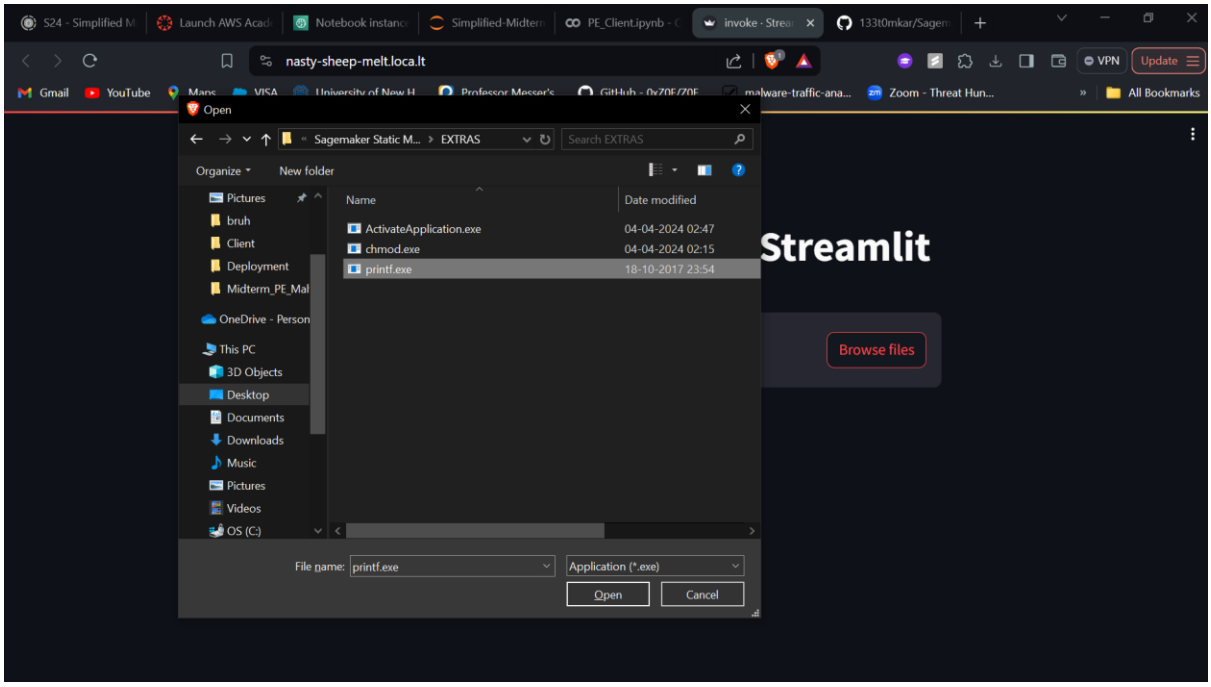

Fig 2: Uploading a malicious/ benign sample to our web application

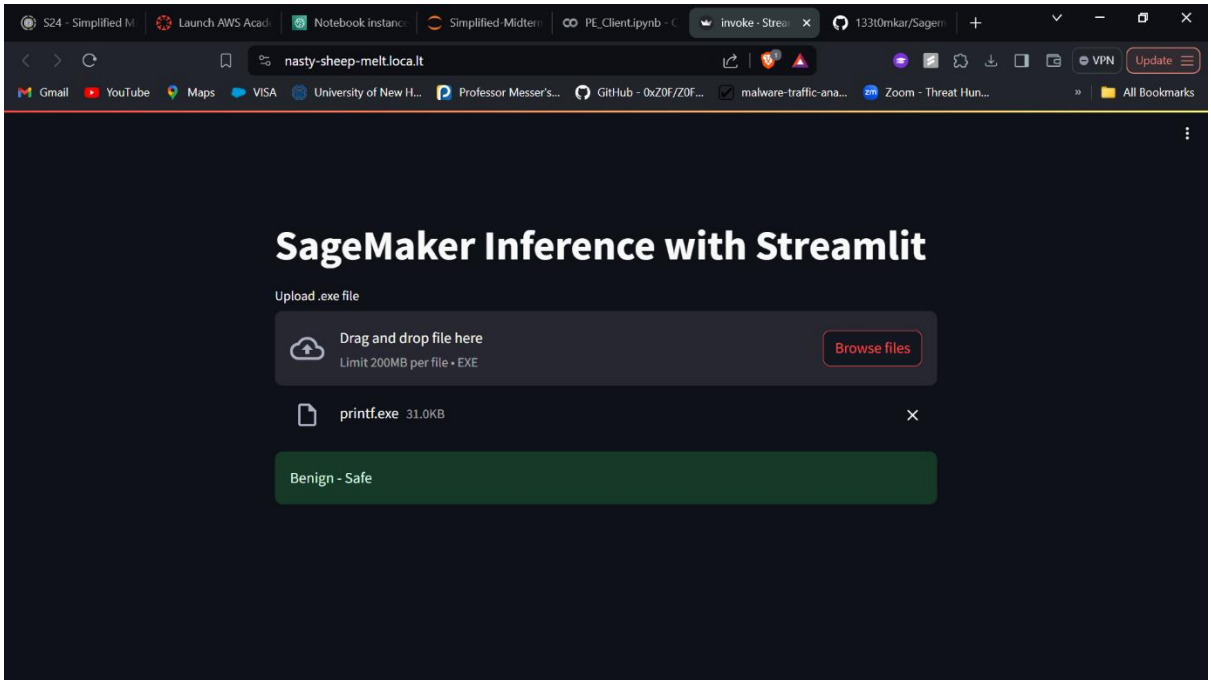

Fig 3: Output result for the given PE executable

Project Results:

I was able to do the following tasks successfully –

1. Deploy the model trained for Lab 5.4 as an API endpoint on AWS Sagemaker.

2. Develop a Python client which takes in an executable file, extracts relevant features, and retrieves classification results from the Sagemaker endpoint.

3. Test your client and endpoint with one malware PE file and one benign PE file from the test dataset (created during Lab 5.4) and demonstrate it in your demo video.

References:

Random Forest Classifier:

https://en.wikipedia.org/wiki/Random_forest

Model deployment:

https://docs.aws.amazon.com/sagemaker/latest/dg/deploy-model.html