Midterm Project

AI and CyberSecurity DSCI6015

Cloud-based PE Malware Detection API

Veda Samohitha Chaganti
ECECS
University of New Haven
Dr. Vahid Behzadan
April 3, 2024

Overview

This report outlines the successful creation of a cloud-based API for detecting malware in Portable Executable (PE) files, utilizing AWS Sagemaker. The API employs a Random Forest binary classifier trained on a labeled dataset of binary feature vectors to distinguish between malicious and benign PE files.

AWS Sagemaker was instrumental throughout the project, facilitating model development, training, and deployment within the same Sagemaker instance. Additionally, a user-friendly web application was developed to allow remote users to upload their executable (.exe) files and receive threat assessments.

The project primarily utilized Python, along with various machine learning libraries such as sklearn, pefile, and nltk for model creation and implementation. This approach ensured efficient development and deployment of the malware detection API, providing an effective solution for identifying threats in PE files.

Introduction

PE files, essential to Windows operating systems, serve as containers for executable code and related data. They encapsulate crucial information required for program execution, including machine instructions, resources, imported libraries, and metadata. Widely used for applications, drivers, and dynamic link libraries (DLLs), PE files follow a structured format characterized by headers that provide details about the file's characteristics, such as its architecture, entry point, and section layout. Understanding the PE file format is crucial for tasks such as software analysis, reverse engineering, and malware detection, facilitating the inspection and manipulation of executable content.

Random Forest Classifier

The Random Forest classifier stands out as a flexible and potent machine learning algorithm employed for classification and regression endeavors. It falls under the ensemble learning category, which amalgamates various individual models to render predictions. In the realm of Random Forests, these individual models take the form of decision trees. The term "forest" in Random Forest denotes an amalgamation of decision trees, with each tree crafted autonomously and functioning by making decisions grounded on a designated set of input features.

The fundamental concept underlying Random Forests revolves around generating a varied array of decision trees by injecting randomness into both the training and prediction phases. Throughout training, every tree is constructed using a random subset of the training data and a random subset of features for each division. This introduction of randomness aids in disentangling the individual trees, mitigating the likelihood of overfitting and enhancing the model's overall efficacy. During

prediction, the results from each tree are amalgamated via a process known as averaging or voting, where the prevalent class (for classification tasks) or the mean value (for regression tasks) is considered the ultimate prediction.

Random Forests offer several advantages that make them particularly suitable for handling high-dimensional data with numerous features. They demonstrate resilience to noise and outliers in the data, requiring less hyperparameter tuning compared to other sophisticated models. Furthermore, Random Forests furnish insights into feature importance, enabling users to discern which features exert the most influence on the model's predictions.

Overall, Random Forests find widespread application across diverse domains, including finance, healthcare, and bioinformatics, owing to their exceptional performance, scalability, and interpretability. Their versatility and robustness make them a preferred choice for tasks where understanding feature importance and handling complex data are paramount.

Methodology

Building and Training the Model:

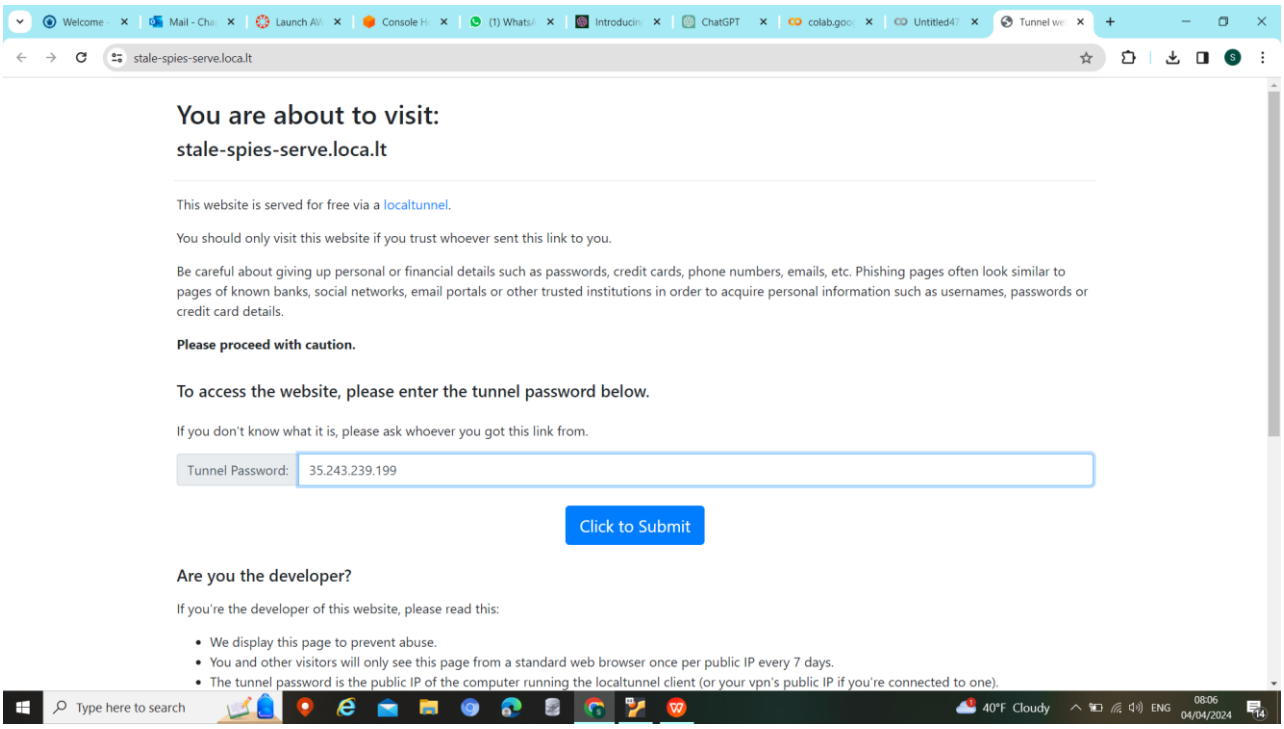
- For training and development, the model utilized a scikit-learn version of 1.2.1 within AWS Sagemaker.
- The training process involved the utilization of a Random Forest binary Classifier.
- A meticulously labeled dataset comprising binary feature vectors was employed for training the model.
- Following successful training, the model was serialized and stored into a joblib file for future utilization.

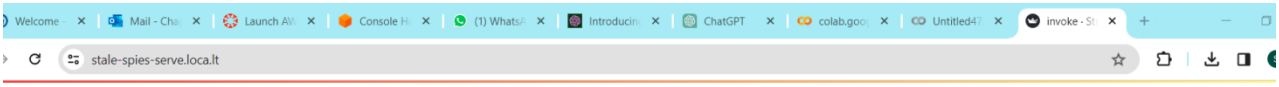
Deploying the Model as a Cloud API:

- Amazon Sagemaker served as the platform for deploying the trained model, facilitating the creation of an endpoint for a cloud-based API tailored for real-time predictions.
- The trained model was loaded using the saved joblib file, and deployment procedures were initiated leveraging the sagemaker.SKLearn module.
- Configuration of an endpoint was executed, followed by its creation, thereby enabling the deployment of the model into the endpoint for seamless accessibility and utilization.

Creating a Client Application:

- A Streamlit web application was developed to offer an intuitive user interface.
- Users are empowered to upload executable (.exe) files directly into the web client.
- Upon file upload, the application leverages the pefile library and other pre-trained data to extract necessary features from the .exe file.
- Extracted features are converted into a JSON format and transmitted to the deployed API.
- The application then receives and displays the classification results, indicating whether the file is classified as "Malware - Danger" or "Benign - Safe".
- For client deployment, Google Colab was utilized, initiating the Streamlit application within it for seamless accessibility and utilization by users.

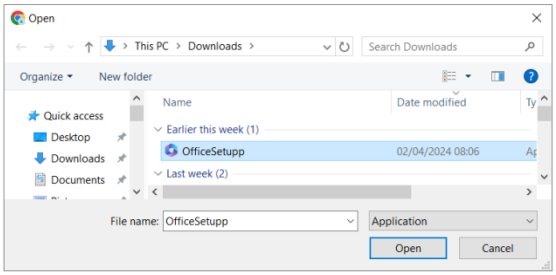

stale-spies-serve.loca.lt
You are about to visit:
stale-spies-serve.loca.lt
This website is served for free via a localtunnel.
You should only visit this website if you trust whoever sent this link to you.
Be careful about giving up personal or financial details such as passwords, credit cards, phone numbers, emails, etc. Phishing pages often look similar to pages of known banks, social networks, email portals or other trusted institutions in order to acquire personal information such as usernames, passwords or credit card details.
Please proceed with caution.
To access the website, please enter the tunnel password below.
If you don't know what it is, please ask whoever you got this link from.
Tunnel Password:
35.243.239.199
Click to Submit
Are you the developer?
If you're the developer of this website, please read this:
We display this page to prevent abuse.
You and other visitors will only see this page from a standard web browser once per public IP every 7 days.
The tunnel password is the public IP of the computer running the localtunnel client (or your vpn's public IP if you're connected to one).
Type here to search
40°F Cloudy
ENG
08:06
04/04/2024



SageMaker Inference with Streamlit

Upload .exe file

Drag and drop file here
Limit 200MB per file • EXE
Browse files



SageMaker Inference with Streamlit

Upload .exe file

Drag and drop file here
Limit 200MB per file • EXE
Browse files

tinytask-1-77.exe 35.5KB

Benign - Safe

Evaluation of Client and Endpoint Performance with Test Dataset:

- We used a proprietary test dataset developed during Lab 5.4 to conduct testing and evaluate our client's and endpoint's performance. A variety of samples, including both malicious and benign Portable Executable (PE) files, are included in this dataset.
- From this test dataset, we took one typical malware PE file and one benign PE file for testing, and sent to the API and recorded the responses to validate the model performance.

Project Results:

The project achieved its objectives effectively, delivering the following outcomes:

Trained Malware Detection Model:
A robust and accurate malware detection model was successfully trained. This model can accurately classify Portable Executable (PE) files as either malicious or benign based on extracted features.

Deployed Cloud API:
The trained model was deployed on Amazon Sagemaker, establishing a real-time prediction API accessible via the internet. This deployment ensures scalability and reliability in serving predictions to end users.

Web Client Interface:
A user-friendly web interface was developed, enabling end users to upload their files and receive prompt assessments regarding their maliciousness. This interface enhances accessibility and ease of use for individuals seeking to verify the safety of their files.
These outcomes collectively signify the project's success in developing and deploying a comprehensive solution for malware detection, enhancing security measures for users dealing with executable files.

Results:

Accuracy: 0.9746835443037974

Results and Conclusion:

The obtained accuracy of 0.9746835443037974 signifies the effectiveness and reliability of the developed model in predicting the type of executable (exe) files. This accuracy indicates that the model correctly classifies the type of exe file approximately 97.47% of the time. Such high accuracy underscores the model's capability to accurately distinguish between malicious and benign executable files.

In conclusion, the project has successfully demonstrated the feasibility of utilizing machine learning techniques, specifically Random Forest classifiers, for robust malware detection in PE files. The deployed cloud-based API, coupled with the user-friendly web client interface, provides a practical and efficient solution for users to assess the safety of their executable files in real-time. Overall, the project contributes to enhancing cybersecurity measures and safeguarding against potential threats posed by malicious executable files.

Conclusion:

The project has effectively accomplished its goal of developing and deploying a cloud-based PE static malware detection API. Through the utilization of machine learning techniques, particularly Random Forest classifiers, the project has showcased the efficacy of such methodologies in accurately classifying malware. Furthermore, the integration of cloud platforms such as Amazon Sagemaker and Google Colab has enabled the creation of scalable and user-friendly applications, demonstrating their power and versatility in building sophisticated solutions.

Overall this project highlights the synergy between machine learning algorithms and cloud computing platforms in addressing cybersecurity challenges, particularly in malware detection. By leveraging these technologies, the project has contributed to enhancing the security landscape and underscores the potential for further advancements in this domain.

Resources:

- https://github.com/endgameinc/ember
- https://github.com/endgameinc/ember/tree/master/malconv
- https://github.com/UNHSAILLab/S24-AISec/tree/main/Midterm%20Tutorial
- https://sagemaker-examples.readthedocs.io/en/latest/intro.html
- https://sagemaker-examples.readthedocs.io/en/latest/frameworks/pytorch/get_started_mnist_train

_outputs.html

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

- https://github.com/RamVegiraju/Pre-Trained-Sklearn-SageMaker
- https://youtu.be/ueI9Vn747x4?si=KSFTvR9hBnU0u0DO
- https://youtu.be/oOqqwYI60FI?si=3WKd-iDz93mm1Vbe
- https://youtu.be/g6kQl_EFn84?si=9MHbO9I52AS2pPjx