Assignment 2

Secure AI Systems : Red and Blue Teaming an MNIST Classifier

DUE: October 3, 11:59 pm

Introduction:
To understand and apply fundamental concepts of secure AI systems. Develop a simple AI/ML application, identify its vulnerabilities, and then engage in a red team/blue team exercise to exploit and defend the system, respectively. This will be done using the classic MNIST handwritten digit dataset.

Team Size : 2 people

Tasks

1) Create a simple Python application using a deep learning framework like TensorFlow or PyTorch. The application should train a Convolutional Neural Network (CNN) to classify handwritten digits from the MNIST (https://www.kaggle.com/datasets/hojjatk/mnist-dataset) dataset.
2) After training, evaluate your model's performance on a clean test set. Document the model's performance metrics such as accuracy, loss, confusion matrix and inference time.
3) Document Threat Modeling applying **STRIDE** framework
4) Use a static analysis security testing (SAST) tool (https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools) to scan your code for common security vulnerabilities. Document and sharing the findings of SAST and fix vulnerabilities, if any
5) **Data Poisoning** – Select a subset of MNIST training data (about 100 of them with "7" as digit). Add a small and colored square to corner of the images to create poisoned dataset.
   Method 2 : For a given clean image set, generate adversarial samples using Fast Gradient Sign Method (FGSM) or Projected Gradient Descent (PGD) that are undistinguishable but are misclassified by the model using Adversarial Robustness Tool (ART) or Foolbox (https://github.com/bethgelab/foolbox)
6) Now test the CNN model again with the new test data and report the model performance
7) **Protection (Blue Teaming)**: Use the adversarial sample along with clean data to train the model. Now test and report the performance metrics to baseline the model.

Deliverables
Report that contains (a) Link of GitHub repository of CNN implementation for image classification (with MNIST dataset) (b) Performance Metrics of classification (c) Threat Model of the implementation (d) Link of Repo of code used for generating adversarial dataset (e) Performance metrics with adversarial dataset

Evaluation Matrix

1) Model training and report performance metrics – 15%
2) Threat Modeling – 10%
3) Implement SAST to identify vulnerabilities – 25%

4) Generation of adversarial dataset & data poisoning – 25%
5) Retest the model and report new performance metrics with clean and adversarial sample – 25%