School of Computing and Information Systems
The University of Melbourne
COMP90073 Security Analytics,
Semester 2 2020
Project 2: Machine learning based cyberattack detection

Release: Tue 1 Sep 2020
Due: 1pm, Tue 20 Oct 2020
Marks: The Project will contribute 20% of your overall mark for the subject; you will be assigned a mark out of 20, according to the criteria below.

Overview

There are two tasks in this project: Task I aims to develop your skills in applying **unsupervised** machine learning techniques for anomaly detection; and Task II helps you better understand how to use gradient descent-based method to generate adversarial samples against **supervised** learning models beyond the computer vision domain. Specifically, two network traffic (NetFlow) datasets are provided, one for each task. Both datasets contain botnet traffic and normal traffic. You need to identify botnet IP addresses from both two datasets. In addition, for Task II you also need to choose a botnet IP address, and explain how to manipulate its network traffic in order to bypass detection.

Deliverables

1. **Task I** – Source code (Python) and SPL queries used to do the following:
   a. Generate/Select features from the packet capture files (training and test datasets) using Splunk.
   b. Use two alternative feature generation/selection methods to select features from packet capture files (training and test datasets).
   c. Build models from *two different* anomaly detection techniques on the generated/extracted features from 1.a. and 1.b. using Python/Splunk.
   d. Score the test data such that cyberattacks are assigned the highest (or lowest)[1] scores.
   e. Return the IP addresses of attackers and the timestamps of their first and last attempt for attacking the network service (per attack scenario).
   f. Compare and discuss the results from different feature extraction and different anomaly detection techniques.
2. **Task II** – Source code in Python, including:
   a. Building, training and testing a supervised learning model.

[1] Optionally anomalies may have lowest scores given the applied technique. Some anomaly detection techniques assign high scores (e.g., distance measure) to anomalies and some of them assign low scores (e.g., probability) to anomalies.

b. Generating adversarial samples for a chosen botnet IP address.

For each task:

3. A comma separated CSV file including the following fields **if applicable**: The time stamp of the start of the attack, the time stamp of the end of the attack, the IP address of attacker, the IP address of the victim, the source port, the destination port, the protocol, the stream ID.
4. A README that briefly details how your program(s)/queries work(s). You may use any external resources for your program(s) that you wish. You must indicate (cite) what are these external resources and where did you obtain them, in the README file.

*Note: need to submit a separate CSV file and a README file for each task.

5. A technical report, of 2000–2500 words addresses Deliverables 1–2.

Technical Report

A technical report, of 2000-2500 words, comprising:

Task I:

1. An overview of the test dataset using Splunk and explaining feature generation/selection using SPL queries and Splunk native functionalities.
2. Description of your methodology for generating features. **Briefly** explain your method for the first project, and discuss your modifications and new findings in Project 2.
3. Review of at least *two* anomaly detection methods that you have used.
4. Description of the experimental setup and evaluation of the (two) methods in detecting anomalies on the test datasets using features generated in Splunk and also features generated using alternative methods. Description should also comprise IP addresses of attacker(s) and victim(s), the attacked service(s), the timestamp, and the type of the attack per attack scenario identified.
5. Description of your final CSV file, the scoring and thresholding technique you used for detecting the reported anomalies[1].

Task II:

6. Explanation of the selected features and your choice of **supervised** learning model. Note that supervised learning is used here, and the mode is the target against which adversarial samples will be generated.
7. Choosing one botnet IP address, and explaining:
   a. How to perturb its features via gradient descent-based method to bypass the detection of your model;
   b. How to update its network traffic in order to satisfy the modified features.

[1] For example, you may choose the best model as your final model or make an ensemble of models.

For both tasks:

8. Conclusion and discussion: for Task I, describe anomaly detection method worked best given the attack scenario; For Task II, discuss the main difference between generating adversarial samples in computer vision and in network traffic.

You should include a bibliography and citations to relevant research papers and external resources and code you have used.

Assessment Criteria

Code quality and README: (2 marks)

Report (18 marks out of 20)

1. Methodology: (5 marks)
   You will describe your methodology in a manner that would make your work reproducible. You should describe in detail how

   Tasks I and II

   a. The features were generated and/or selected?
   b. The training data was used to learn the anomaly detection models? You should explain how the parameter settings for your methods were performed (e.g., setting the ν parameter in OCSVM[1]). **You should not use the test data for setting the parameters**.

   Task I

   c. The scoring was performed in each model to rank the data instances?
   d. The thresholding on the scores was performed in each model to label the attacks?
2. Accuracy of the results: (5 marks)
   Tasks I and II
   Your machine learning based technique should generate a report of detected attacks on the test datasets. This should be the output of your algorithm and you should not change it based on your analysis. It should indicate the IP address of the attacker and the victim, the attacked service, and the period (timestamps) for which the attack was happening. You are marked out of 5 based on the percentage of successfully detected attacks by your anomaly detection model[2].
3. Critical Analysis: (4 marks)
   Task I

   a. Use of Splunk for feature generation/selection from packet capture files (training and test datasets).

[1] For anomaly detection methods that require validation set for parameter tuning, you can combine a small amount of anomalies (about 5%) from the of the attack day dataset to your training set.
[2] $Mark = (5 + 0.1) * (TPR - FPR)$. TPR is true positive rate and FPR is false positive rate given your submitted CSV file explained in point 2 of the project deliverables. Since $FPR = 0$ is almost impossible, we added 0.1 bonus.

b. Discuss the differences in processes, scalability, and results identified using the Python code developed for anomaly detection.

Task II

c. Explain how you generate the adversarial sample, and how is the process different from the computer vision domain.

4. Report Quality: (4 marks)
   You will produce a formal report and express your methodology and findings concisely and clearly. The quality and description of figures, tables, and the README file should be acceptable.

Description of the Data

The two datasets for Project 2 (download link) contain the NetFlow data for a network under cyberattacks. Each line of the dataset includes the following 14 fields: (1) timestamp, (2) duration, (3) protocol, (4) source IP address, (5) source port, (6) direction, (7) destination IP address, (8) destination port, (9) state, (10) source type of service, (11) destination type of service, (12) the number of total packets, (13) the number of bytes transferred in both directions, (14) the number of bytes transferred from the source to the destination.

Changes/Updates to the Project Specifications

If we require any changes or clarifications to the project specifications, they will be posted on the LMS. Any addendums will supersede information included in this document.

Academic Misconduct

For most people, collaboration will form a natural part of the undertaking of this project. However, it is still an individual task, and so reuse of ideas or excessive influence in algorithm choice and development will be considered cheating. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy (http://academichonesty.unimelb.edu.au/policy.html) where inappropriate levels of collusion or plagiarism are deemed to have taken place.

Late Submission Policy

You are strongly encouraged to submit by the time and date specified above, however, if circumstances do not permit this, then the marks will be adjusted as follows. Each day (or part thereof) that this project is submitted after the due date (and time) specified above, 10% will be deducted from the marks available, up until 5 days has passed, after which regular submissions will no longer be accepted.

Extensions

If you require an extension, please email Yujing (yujing.jiang@unimelb.edu.au) using the subject 'COMP90073 Extension Request' at the earliest possible opportunity. We will then assess whether an extension is appropriate. If you have a medical reason for your request, you will be

asked to provide a medical certificate. Requests for extensions on medical grounds received after the deadline may be declined. Note that computer systems are often heavily loaded near project deadlines, and unexpected network or system downtime can occur. Generally, system downtime or failure will not be considered as grounds for an extension. You should plan ahead to avoid leaving things to the last minute, when unexpected problems may occur.