C3i Hub, Indian Institute of Technology Kanpur

HCL HACK IITK 2021

Submission Round: Programming Challenge

16th Jan 2022 – 23rd January 2022

Question 1 / Problem 1: Develop an IDS suitable for CPS

Instructions:

1. This round is consisting of two problems, each carrying 100 marks. Question 1 is given below, Question 2 will be given on Monday.
2. Please read the Questions very carefully
3. The questions contain links to download the dataset provided to develop ML-based IDS
4. The test dataset (without level) will be provided 24hrs prior to the final submission deadline.
5. You can make multiple submissions, only the last submission will be considered for evaluation, the previous versions will be automatically erased
6. Last date for submission is 23rd January 2022

Question 1 / Problem 1: Develop an IDS suitable for CPS

Maximum Marks 100

Description: Critical infrastructures (CIs) are cyber-physical systems that facilitate and boost societal and economical operations. Some examples of CIs include infrastructure supporting supply of natural gas, water treatment and supply, food production and distribution, transportation, healthcare, banking and finance, telecommunication, and goods and services. A CI has a layered architecture. It is supported by an industrial control system (ICS), Supervisory Control and Data Acquisition systems (SCADA), and Process Control Systems (PCS) that monitor and control the infrastructure. These high-level architectures of supervisory systems are typically connected to Programmable Logic Controllers (PLCs) directly or with the help of Remote Terminal Units (RTUs) in case of a large network. The PLCs are industrial computing devices linked to the sensors and the actuators that control the physical processes by communicating with the supervisory control system, usually SCADA. The SCADA is often augmented with an intrusion detection system (IDSs) that regularly monitor the physical processes or network data to raise the alarm when the system operates abnormally.

There are multiple methods are used to secure CIs. These methods include securing network architecture by adhering to the policies such as network segmentation and segregation, the use of boundary protection devices, and firewall filters between each network segment. However, network security is constantly being breached due to exploitation of vulnerabilities that also include zero-day vulnerabilities. Assuming that network security is foolproof and no attacker will break it to cause harm to the ICS is not correct. Only by bypassing the network security, attackers do not always harm the CI immediately but they wait

for an opportune moment while slowly affecting the devices. When an attacker performs such malicious activity on the CI, its effect gets reflected in the dynamics of the physical processes. The sensors and control behavior associated with attack-targeted devices start to show structural and behavioral changes. Such changes can be identified by an IDS to detect an attack and save the CIs from lasting damage.

Dataset: You are provided two subdatasets:

Subdataset 1: The data consists of measurements (real-values) of 41 sensors periodically (period of 1hr), generated for the duration of around one year. This data represents the behavior of 41 sensors collected periodically over a duration when there is no attack on the system. So they represent **nominal behavior** of the system**.** However, there is a chance that this data may be subject to data poisoning or false data injection attacks. Thus, while using this data to learn the nominal behavior of the system, you may have to use some robust machine learning technique.

Link of Subdataset 1:
https://drive.google.com/file/d/1P9SvJBBgkm0GZCBS8eMQ_Z5JsAplXelQ/view?usp=sharing

Subdataset 2: The dataset is collected from the same architecture but during seven different attacks. Each attack is active for a duration, as given below. This dataset aims to be useful to validate your model's performance. You may or may not use this dataset to do the training of your model.

Link of Subdataset 2:
https://drive.google.com/file/d/1mDjswuClMc7CrXW5BgYyVUeeb8NQdWpO/view?usp=sharing

ID	Starting time dd/mm/yy hh	Ending time dd/mm/yy hh	Duration (hours)
1	13/09/20 23	16/09/20 00	50
2	26/09/20 11	27/09/20 10	24
3	09/10/20 09	11/10/20 20	60
4	29/10/20 19	02/11/20 16	94
5	26/11/20 17	29/11/20 04	60
6	06/12/20 07	10/12/20 04	94
7	14/12/2020 15	19/12/20 04	110

Steps to follow:

- **Data collection**: Collect the two CSV files provided through the links.
- **Develop model:** Train an ML model which takes inputs from the sensor measurements at a time stamp ‘t’ and classifies it either in “ATTACK” or “NORMAL” categories. That means that you want to determine if at time stamp t, the system is under attack or under normal operation. Developing an unsupervised or semi-supervised time-series ML model for better accuracy is suggested.

Note: The model is not allowed to take input from the future. For example, if you classify the timestamp 't,' you can pass only up to 't'th measurement as input.

- **Develop tool:** Develop a command-line tool that accepts a CSV file (test.csv) of the same format as input and outputs a CSV file (result.csv) of two columns: "DATETIME" and "ATTACK/NORMAL." The order of DATETIME should be the same as the provided test CSV file.

Project must fulfill these requirements as mentioned below:

The developed tool must have good accuracy, precision, recall, and F1-score for the machine learning model with low false positive and low false negative rates. In case of close competition, tool having low testing computational cost will be awarded additional marks.

Deliverable:

This is a tentative deliverable plan. As mentioned before, the test data to generate the result.csv will be provided 24 hours prior to your final submission deadline.

- A result.csv file (a two column csv output file) which is the final result collected by you. We will use this file to compute accuracy by comparing it with the actual labels. The following deliverable will be used to validate the result.csv file.
- Your IDS tool must be named IDS_test.py. This tool takes as input a CSV file (test.csv) and saves a two-column CSV output file (result.csv). We may consider the computational efficiency of IDS_test.py to award additional marks. The result.csv file provided by you and generated by us using your submitted IDS_test.py must be the same.

  python IDS_test.py absolute_path_for_test.csv_file
- You have to provide the entire code in a well-structured and commented format which you developed for data processing, model training, and testing (whatever you have done) with comments and a README file. This will allow us to verify that you have not done any hard coding of outputs or your model is not biased for the test data.
- All the above three files/folders must be zipped within a single folder named "P1 _Teamname."

Cheating policy: We will conder the following activities as cheating and any team engaging in any of them will be disqualified.

- Collaboration with other teams
- Intentionally training model to be biased for the test data
- Random output or only one class output
- Anything else that is explicitly not allowed