Intrusion detection system for critical infrastructures

Esha Sarkar
NetID: es4211

***Abstract*—Critical infrastructures, till recent times, have ensured security of their systems using air-gapped Networks. However, with the growing need to continuously monitor and control the remote plants, introduction of IT to critical infrastructures was inevitable. With IT, came the cyber-attacks as well. Threshold based security systems also fail because the faults which get wrongly tagged as attacks become a safety concern. Naturally Machine learning is a good candidate to learn to detect an attack and to differentiate it from a system fault. In this work, I present an intrusion detection system based on both network characteristics and physical characteristics that can detect attacks with high accuracy.**

I. Introduction

In 2011, Stuxnet [1], a worm which seemed to be designed specifically for industrial control systems surfaced. It attacked at least 14 sites of Ukrainian Nuclear Plants causing the centrifuges to run at dangerous speeds, thus tearing themselves apart. In addition to Stuxnet, there have been many attacks on critical infrastructures which have increased focus on security of critical infrastructure systems. Shodan [2], a 'Google' for IoTs, has demonstrated how to use common network parameters and intelligently query industrial IoTs to extract more information about them. Fig 1 shows an output of Shodan search query which lists devices from Schneider Electric Telecontrol (a company specializing in industrial control systems). When I click on the first link, I can see the login page of a PLC situated in France (Fig 2). Furthermore, a quick search on the product reveals the default username and password. Unfortunately, these credentials can give one access to a plant if the commissioning engineer forgot to remove it and I could log in many devices (screenshot is not attached in this report for security reasons). But it is clear that network is one of the primary entry points for attack.

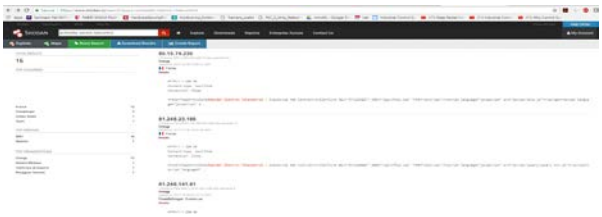

Fig 1: Shodan results for Schneider Electric

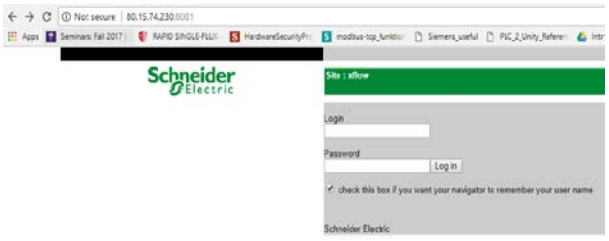

Fig 2: PLC login page

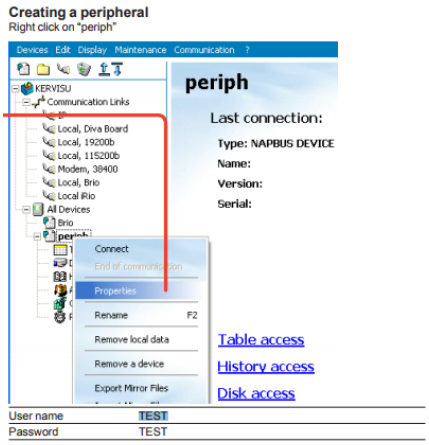

Fig 3: Default credentials of Schneider Electric device retrieved from Google search

Apart from directly logging into remote ICS, an attacker can choose to directly issue commands to a remote device or extract information because the network protocols used in critical infrastructures generally do not have authentication. In [3], the attacker querries for some specific register values using MODBUS and extracts vendor name of PLCs and firmwares running in it.

MODBUS is one of the most common protocols used in critical infrastructures. In this work I use a dataset of MODBUS packets which are relaying process information between Supervisory Control And Data Acquisition (SCADA) and field device. Many kinds of attacks (described in the next section) are simulated in this dataset. Also a preliminary analysis done on this dataset states that the maximum accuracy they get is 85.22% [4]. The primary aim of this work is to increase the accuracy of detection of attacks. Also, I want to decrease the dependence of intrusion detection algorithm on domain knowledge i.e. in this case the algorithm should not be fed gas pipeline specific background information before hand.

II. The Dataset

A. *Industrial Control System description*

The system is essentially a gas pipeline system monitoring different parameters of the system and relaying back the information to SCADA. Thus, if we look into the MODBUS packet we can easily figure out the value of the physical parameter (for example pressure). Now, depending on these values the SCADA may issue control commands. For example, let us suppose pressure of the pipeline should be X kPa and SCADA gets a value of X-Y kPa from the field. Then

SCADA issues a command to increase the pressure by Y amount.

B. *Threat model*

The threat model assumes that the attacker can snoop into the network to alter the values in the packets to make SCADA malfunction. For example, in the previous case, the data from the field is X kPa but SCADA receives a value of X-Y kPa. The SCADA will wrongly ask the field actuators to increase the pressure resulting in a gas pipeline pressure of X+Y. The attacks enlisted in Table 1 are launched during normal operation of gas pipeline and the dataset was generated.

Normal Operation
Naïve Malicious Response Injection
Complex Malicious Response injection
Malicious State Command Injection
Malicious Parameter Command injection
Malicious Function Code Injection
Denial of Service
Reconnaissance

Table 1: Types of attacks on Gas pipeline

III. TRAINING THE DATASET

A. *Extraction of features*

The information relayed by the Modbus packet are as follows:

1. Address
2. Function Code
3. Length
4. Setpoint
5. Gain
6. Reset rate
7. Dead band
8. Cycle time
9. Rate
10. System mode
11. Control Scheme
12. Pump
13. Solenoid
14. Pressure Measurement
15. CRC rate
16. Command response
17. Time

Potentially all of the information can be used as features to derive whether a packet reflects an attack or not.

B. *Machine Learning Algorithms*

The goals of selecting algorithms are listed below:

1. It should be able to differentiate between fault and attack. It should also be able to detect the type of attack (Table 1)
2. The algorithm should be light-weight: it should not interfere with the primary function of industrial computer
3. It should not require any prior domain knowledge: in this, no networking knowledge or gas pipeline data should be available

I did not want to use neural network because of point 2. My initial intuition was to use logistic regression to predict if there is an attack. Changing the value of 'C' did not have any effect on the accuracy. The following figure shows the accuracy does not change for a wide variety of values of C.

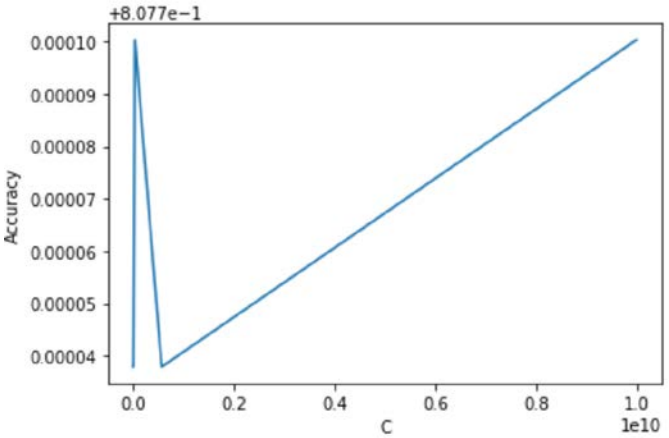

Fig: Change in accuracy with value of C

IV. SUPPORT VECTOR MACHINE

The next natural step is to use SVM. Now, I tried for C values from 10^{-3} to 10^{3}. But the accuracy did not improve. Then I tried for improving accuracy by changing values of gamma 0.001-10 and accuracy improved but again not to the level of my goal.

Also, I wanted to decrease the values of C and gamma to values which I have used in labs.

A. *Feature selection*

I found the weights of the features from logistic regression and removed the features which had the lowest two weights. Removing Address and Function code increased accuracy but it did not go above 95% which I wanted. This is intuitive from the nature of dataset. When we see the type of attacks, we can see that the attacks are a combination of malicious manipulation of all the features. For example, although address does not seem important, but one way for Malicious Response attack is to change the address. So, although, all the data are correct and pertinent to the request from the SCADA, the address does not match with request. The SCADA, thus, does not interpret the response correctly and sends the request again and does it till it gets a response. This leads wasting of bandwidth.

B. *Intuitive division of dataset*

I noticed that the request and response are different in terms of features they use i.e. certain fields are not used in request and the others are not used in response. So, I need the algorithm to understand the difference between request and response and automatically understand which features it should operate on.
Thus, the learning finally occurs in 2 steps:

1. First, the algorithm sees the feature called 'Command Response'
2. According to the value of the field, it decides what features to select and the selection is not based on weights

C. *Final algorithm*

Even after this division of dataset, the accuracy did not improve with logistic regression. But when I ran SVM with C=5 and gamma=0.09, the accuracy improved to 98.8%.

D. *Discussion with other algorithms*

To decide on the randomness and non-learnability of data, the creator of this dataset ran the following machine learning algorithms. The corresponding accuracy were as follows:

1. Naïve Baeysian Network-80.39%
2. PART-78.1%
3. Multilayer Perceptron-85.22%

All these are very heavy algorithms for Industrial Control systems and with low accuracy. Thus, my way of learning has accuracy more than all of them

V. DISCUSSION

The main aim of the project was to increase the accuracy, i.e. to detect if there is an attack or not. That aim is satisfied. Moreover, SVM for the dataset is not as heavy as Neural Network based algorithms but heavier than Logistic regression. I checked that using CPU usage, when the algorithms were running. However, the third aim is not satisfied. Here, I wanted the algorithm to be oblivious to the domain of system but in my method, I had to make the algorithm understand whether it is a response or a request. This needs networking knowledge which the algorithm could not learn itself.

Finally, I have to note that the attack accuracy has substantially increased which would ensure the first level of security.

This report is complementary to the code I have attached. The aim of this report is to justify the motivation and the quirks of Industrial Control System which are different from conventional application of machine learning.
Please note, when I was coding, I deleted most of my trials and kept the one which worked. Later, when project description was updated and it was required to show the trials where accuracy was less, I just had the time to update one trial (Logistic Regression).

REFERENCES

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier," Whitepaper, Symantec Corp., Security Response, vol. 5, 2011.
[2] https://www.shodan.io/
[3] A. Keliris, M. Maniatakos, "Remote Field Device Fingerprinting Using Device-Specific Modbus Information", IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2016
[4] Ian's Defense
[5] https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets