Network Traffic Analyzing and Modeling for Cybersecurity Threat Detection using AI.

Saber Mhiri

***Abstract*—nowadays, there is a serious need for a tool capable of analyzing industrial networks and detecting abnormal behavior in it.**
In order to create such a tool, we have to face the following obstacles:

First, the complicated nature of the OT environment: Although the components in an industrial network aren't as dynamic as other networks (such as IT network or cellular networks. . .) OT networks are highly complicated and, in most of the cases, even the network administrator have no idea about the relationships between the network components, neither the complete asset list in their network.

Second, the lack of data related to OT environment communications: Access to industrial network communications traffic is usually restricted due to the sensitivity of these data.

Faced with these obstacles, the need for a data-set containing traffic communication in an industrial network is increasing. Having such data-set would help researchers develop better network protection systems.
Considering the success of AI in cybersecurity threat detection [1], AI technologies should be used to analyze the behavior in the OT networks, especially when talking into considering the amount and the complicated nature of data in OT networks in terms of quality and quantities.

***Index Terms*—OT network, industry 4.0, monitorization, threat detection, threat alerting.**

I. Introduction

Due to OT networks important impact on the physical and economical world, the need for smarter approaches for traffic analyzing and threat detection in the industrial networks became a major need for companies today.
Having an AI module that can detect threat in the network is a highly needed capacity specially with the increase of AI based cybersecurity attacks [2]. Having a well-trained AI module would tremendously help industrial SME's in keeping their systems protected during their transition into the industry 4.0. To this need, it is important extracting useful data from traffic that help deduce the state of the network as well as the threat detection.
It is also important to have enough data that are well modeled in order to train an AI system to learn behavior and detect threats in the network.
In this paper we focus on the type of data found in an industrial network that can be used for training an AI module, as well as presenting a data model that we designed in order to extract these needed data.
Our objective is preparing a module for the extracted data in order to train an AI module capable of detecting threats in the network through learning normal behavior in the network and detecting abnormalities. In this work we will start by presenting the proposed approach in chapter 2, then evaluating our approach and presenting the results in chapter 3 and finally we represent the conclusions.

II. Proposed approach

A. Objective

In this work we focus on identifying the most useful data that can be extracted from the traffic in an OT network. Identifying the relevant information in the network depends on the security threat that the system would be protecting against. We identify 4 main threats that our system will be identifying it:

- **1- Unauthorized communication:**
  Our final AI system should be able to detect any new element in the network. It should also detect any change in the protocols used in the network. In the case of a new device appearing in the network, our system should be able to identify such threat and launch an alert in order to identify if this new device is authorized.
- **2- Communication map change:**
  Our final AI system should be able to detect any change in communication between network devices, for example if an a control PC is usually communicating to a PLC1 in order to program it, then one day it start communicating with a PLC2, our system should be able to identify such threat and launch an alert in order to identify if the user of that PC have the required authorization or if it is a malicious script running on that PC.
- **3- Network communication change:**
  Our final AI system should be able to detect whenever a device start sending or receiving an abnormal amount of traffic, for example, if a web-server of a PLC start receiving a huge amount of traffic, or traffic at a suspicious hour of the day, our system should be able to identify such threat and launch an alert in order to identify the possibility of a denial of service attack or an unauthorized access.
- **4- PLC state change:**
  Our final AI system should be able to learn the state of PLCs in the network and detect any change in the PLC states that isn't usual. A PLC that start turning on and off an engine in the industrial platform risk the explosion of such engine. A PLC that increase the pressure of a container can cause it's destruction. Our system should

be able to identify such threat and launch an alert in order to rapidly deal with such threat.

B. Approach description

In order to collect the data in the network, we need to start by sniffing the industrial network communication and identifying the main data that will affect our decision making and then modeling these data into a more dedicated easy to implement data form for an AI module.

C. Identifying the needed data

After analyzing the industrial network, we managed to identify the following data as the main indicator of the state of the network. The data can be distributed into the following groups:
Data related to nodes in the network: such as Firmware version, IP and MAC address, OS, ...
Data related to the link in the network: source and destination of each communication, the ports used, the protocol implemented, ...
Through focusing on these data, we believe we will be able to have the minimum required network identifiers to identify the threads in the network.

D. Modeling the data

Through monitoring the traffic in the network, we realized that in order to detect the different threats mentioned previously, we need to:

- First collect the logs to learn the activities in the network.
- Second create a matrix of connection. Through this matrix we can see which machines are communicating together and through which ports and protocols. Having such matrix will ensure the detection of threats (1) and (2), unauthorized communications and the connection of a unauthorized machine in the network which will lunch the alerts.
- Third, create a matrix of communications, where we can know how many packets have been exchanged between machines. Having these data collected will identify any attack related to the threat (3).
- Finally collecting the state of the PLCs in the network periodically.

All the collected data would be saved in a BD in order to be analyzed and studied later. The reason for analyzing these data is to be able to deduct the normal behavior in the network and extract rules that would oversee lunching alerts in case of threat detection.
The choice for modeling most of the data as matrices is in order to help the AI module with the learning process, knowing that NN have shown a great success rate in the field of image processing and recognition. For that reason, and just like an image is treated as a matrix, we opted to model most of the extracted data as matrices.

E. Scripts and tools

In order to monitor and analyses the traffic in the network, we started by sniffing the network using the library pyshark. This library allows us to monitor the traffic in the network. Pyshark have a rich collection of industrial protocols dissectors available which help us analyze the industrial network state.

We developed an algorithm in charge of monitoring and extracting the different useful data in the network. The script in charge of monitoring and collecting data in the network called OT_integrator will periodically collect and analyze the network traffic, extracting the different threat indicators previously mentioned in section 2.1.
The OT_integrator collect the logs, monitor then create and save in a DB the matrix of communication and connection. It also collects and saves the PLC state in term of active and inactive Inputs and Outputs.

III. Evaluation and results

A. Testing environment

In order to test our OT_Integrator capability of detecting the network state as well as the state of the network components such as PLC, Lab PC, ⁚. we decided to create a test environment capable of simulating the varayity of network protocols and components in the network.
To that end, we will be using different PCAPS containing traffic with a variety of industrial protocols focused on these main protocols:

- **s7Comm [3]**: the protocol used by the Siemens products.
- **ModBus [4]**: a basic but certainly still popular protocol that is still in use by most of the industrial plants appliances. Used by multiple manufacturers specially Schneider and Allen-Bradley.

These PCAPS are collected from industrial cybersecurity workshops such as 4SICS [5] where a real industrial network components where used to create a testing ICS lab. The 4SICS [5] ICS lab contained 5 rack detailed as follow:

- Rack 1
  - DirectLogic 205 (PLC)
  - Phoenix Contact FL IL 24 BK-PAC (Ethernet bus coupler) Open Ports: 80 (HTTP), 502 (ModBus [4]/TCP), 1962
  - Advantech ADAM-5500 (Ethernet I/O module) Open ports: 21 (FTP), 80 (HTTP), 81
  - AXIS 206 Network Camera Open ports: 21 (FTP), 80 (HTTP), 49152 (UPnP)
  - Hirchmann EAGLE 20 Tofino (Firewall) Open ports: 22 (SSH), 443 (HTTPS)
  - Allen-Bradley Stratix 6000 (managed switch)
- Rack 2
  - Siemens SIMATIC S7-1200 (PLC) Open Ports: 102 (S7 protocol), 5001
  - RUGGEDCOM RS910 (Serial device server and ethernet switch)

  - RUGGEDCOM RS910
  - RUGGEDCOM RS910
  - RUGGEDCOM RS910 Open ports: 22 (SSH), 23 (Telnet), 80 (HTTP), 443 (HTTPS), 502 (ModBus [4]), 514 (RSH), 20000 (DNP3)
  - RUGGEDCOM RX1100 (Router)
- Rack 3
  - Red Lion DSP (protocol converter) Open ports: 80 (HTTP), 502 (ModBus [4]/TCP)
  - Moxa EDS-508A (Switch) Open ports: 22 (SSH), 23 (Telnet), 80 (HTTP), 443 (HTTPS), 502 (ModBus [4]), 4000, 4001
  - Moxa EDS-508A (Switch) Open ports: 22 (SSH), 23 (Telnet), 80 (HTTP), 443 (HTTPS), 502 (ModBus [4]), 4000, 4001, 44818
  - Cisco Catalyst 2955 (Switch)
  - Moxa UC-7112 (Embedded computer)
  - HOST Engineering MB-gateway (ModBus [4] gateway) Open ports: 80 (HTTP), 502 (ModBus [4])
- Rack 4
  - Beckhoff CX1010 (Win CE, open port: 1234) Open ports: 23 (Telnet), 80 (HTTP), 135, 443, 1234, 5120
  - xLogic x-Messenger EXM-12DC-DA-RT-WiFi (WiFi PLC)
  - TCP/IP to RS-232/422/484 Converter
- Rack 5
  - and 192.168.87.1 Westermo Lynx 3643-0105 (Router)
  - Westermo Lynx 3643-0105 (Router)

these PCAPS from 4SICS [5] contain both ModBus and s7Comm traffic that would be useful for our tests and evaluations of our OT_Integrator. As for the PLC state change, we programmed the PLC to randomly change it's state, these changes are saved in a file that will be used for verification purposes for the accuracy of out OT_Integrator results. Due to the variety of protocols used in the industrial compounds, we opted to limit our OT_Integrator system to 2 commonly used protocols, the s7Comm [3] from Siemens [6] and ModBus [4] used by Schneider products.

adapting our algorithm to support these 2 protocols needed the use of dissectors in order to extract and model the data in the network traffic.

Theoretically,in case of an abnormal behavior detected, our monitoring system will alert the engineering are via the monitoring PC screen and will try to alert the workers in the industrial layer through igniting the orange light.

To test our data collection and modelling module the OT_Integrator, we used different PCAPs that contain s7Comm [3], ModBus [4] traffic

B. Evaluation scenario

Scenario

- **First step:** Analysing the state of the PCAPs:
  In order to test the efficiency of our algorithm, we need to analyse the state of the network in order to compare the accuracy of the data provided by our algorithm compared to the state of the real state of the network.

- **Second Step:** monitoring the network and extracting the data:
  We start by running the OT_integrator in order to collect the network traffic and save it in the DB.

- **Third Step:** testing the efficiency of the OT_Integrator:
  After running our OT_integrator for 1 hour, we managed to create a database containing the different information's needed about the network in terms of connection and PLC states and the logs.

Evaluation We compared the results with the real state of the network, we decided on the following indicators as a reference to the success level of our solution

- α: Percentage of hosts detected.

$$\alpha = \frac{nb_{discovered\ hosts}}{nb_{real\ hosts}}$$

- β: Percentage of links detected (s7Comm [3]and ModBus [4]).

$$\beta = \frac{nb_{discovered\ links}}{nb_{real\ links}}$$

- ζ: Percentage of times PLC state change detected.

$$\zeta = \frac{nb_{discovered\ PLC\ statechange}}{nb_{real\ PLC\ state\ change}}$$

C. Results

After dividing the PCAPs of 4SICS [5] into 13 smaller PCAP files, we run our OT_Integrator on a randomly selected PCAP files from these previously mentioned PCAPS focusing on collecting data related to s7comm and ModBus [4] traffic. This random selection and test where repeated 5 times and the results of our algorithm where compared to the real data from the PCAP according to the previously mentioned metrics α, β, ζ.

Figure 1 shows the α, β, ζ resulting from testing our OT_Integrator.

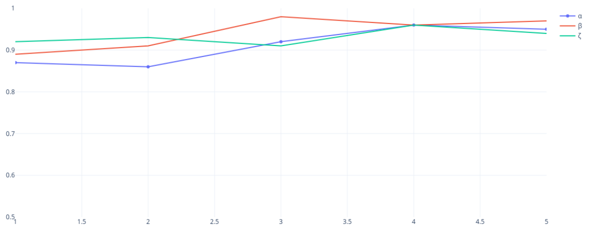

Fig. 1. OT_Integrator accuracy

These results show that our OT_Integrator manage to provide results with accuracy Superior to 85% up to 98%. After looking through the results in details we realized that the reason for not arriving to a 100% is that the PCAPS and the testing envirement have components that don't use neither s7Comm nor Modbus, that's why these applience where not discovered neither their links.

IV. Conclusion

In this work, we identified some needs for the industrial network protection and developed a solution that can analyze the network and lunching an alert when identifying a threat in the network. the proposed solution where tested and proven success as the results in shown in chapter 3.

References

[1] DNV, GL. "Global opportunity report 2017." Høvik, Oslo: DNV GL AS (2015).
[2] https://www.symantec.com/blogs/feature-stories/cyber-security-predictions-2019-and-beyond
[3] https://wiki.wireshark.org/S7comm
[4] http://www.bb-elec.com/Learning-Center/All-White-Papers/Modbus/The-Answer-to-the-14-Most-Frequently-Asked-Modbus.aspx
[5] https://www.netresec.com/?page=PCAP4SICS
[6] https://new.siemens.com/global/en.html
[7] https://builtin.com/artificial-intelligence