Network Intrusion Detection based on Deep Neural Networks for the SCADA system

Mustafa Altaha¹, Jae-Myeong Lee⁴, Muhammad Aslam², and Sugwon Hong⁴*

¹Department of Computer Engineering, ²Department of Electrical Engineering, Myongji University, Yongin, Gyeonggido 17058, R. of Korea

swhong@mju.ac.kr

Abstract. Security monitoring is one of the security strategies for the supervisory control and data acquisition (SCADA) systems, and the intrusion detection system (IDS) is a main tool to do security monitoring. Main task of security monitoring is to develop the SCADA-specific IDS, which reflects the semantics of the SCADA domain. In this paper, we work on developing IDS based on deep learning models for the SCADA system. The target SCADA communication protocol of the detection model is the DNP3, which is currently the most commonly utilized communication protocol in the power substation. The attack of major concern is data injection or modification attacks, which is most critical attack in the SCADA system. We extract 12 data features from distributed network protocol 3 (DNP3) packets, and use them to train the deep neural network. We measure the accuracy and loss of the detection system trained based on different deep learning algorithms, and show the comparison of the results.

1. Introduction
Supervisory control and data acquisition (SCADA) systems are process control systems which interconnect and monitor remote physical processes. SCADA systems collect data from remote facilities about the state of the physical process and send commands to control the physical process creating a feedback control loop. SCADA systems are used in power transmission and distribution systems for situational awareness and control. Security monitoring is one of the integral strategies for the SCADA systems, and the intrusion detection system (IDS) is a main tool of doing security monitoring. The anomaly IDS detection is the process to determine which observed events are to be identified as abnormal because it has significant deviation from normal behavior which is called ‘profile.’ The difficult part is how to decide or derive profiles which reflect all semantics of the system. Thus, the main task to do security monitoring is to design domain-specific IDS which is aware of the target domain semantics.

Deep learning and machine learning have shown some achievement in different fields, such as video recognition and image processing, audio processing, natural language processing and robotics, etc. Therefore, many researchers have begun to focus on constructing IDSs using machine learning and deep learning methods [1]. However, very few work have been done for developing the SCADA-specific IDS, which reflects the semantics of the SCADA domain [2-6].

In this paper, we work on developing IDS based on deep learning models for the SCADA system. The target of the detection model is the DNP3 protocol, which is currently the most commonly utilized communication protocol in the power substation [7]. The paper focuses on detecting the data injection or modification attacks, which are most critical attacks in the SCADA system [8, 9].
The rest of this paper is organized as follows. In section 2, we present the proposed SCADA-IDS model based on deep learning methods. In section 3, we present an experiment setup and evaluation of the proposed methods. Finally, we suggest conclusion and future work in section 4.

2. IDS MODEL

The IDS design is composed of two main phases: the extracting phase and the detection phase. The extracting phase is performed offline as it is somehow time consuming. In the extracting phase, the DNP3 packet is processed to extract data features that represent the behaviour of the network. Each processed DNP3 packet has a label indicating either normal or anomalous packet. This is used for the detection phase by adopting the deep neural network structure to train with the features. The detection phase uses the same features, which are extracted from the extracting phase, as input, and the neural network structure calculates trained parameters to predict which class packets belong to, either normal or abnormal.

2.1. Extracting Phase

The generated dataset has 500 normal instances, 10 disable unsolicited messages attack instances, 480 denial of service (DoS) attack instances, and 10 injected cold restart command attack instances. This gives a total of 1000 instances (almost 40 thousand packets). This dataset consists of 12 features, which are also used similarly by [10]. We choose these features because they have high correlation with possible anomalous behaviours.

- Destination bytes: Number of data bytes transferred from destination to source in a single connection.
- Source bytes: Number of data bytes transferred from source to destination in a single connection.
- Flag: Status of the connection, i.e. Normal or Error.
- Count: Number of connections to the same destination host as the current connection in the past two seconds.
- Service count: Number of connections to the same service (port number) as the current connection in the past two seconds.
- Destination host same source port rate: The percentage of connections that were to the same source port among the connections having the same port number.
- Same service rate: The percentage of connections that were to the same service, among the connections aggregated in Count.
- round trip time delay (Rttd): The length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgement of that signal to be received.

To detect attacks that require examination of the payload, the following DNP3 specific features are extracted.

- Contains DNP3 packet: A feature indicating if the connection contains a DNP3 packet.
- DNP3 payload length: The total length of the DNP3 payload contained within the connection.
- Min DNP3 payload length: The minimum DNP3 payload length in the connection.
- Cold restart in DNP 3 packet: A Boolean indicating if there exist a cold restart or disable unsolicited message command in the connection.

2.2. Detecting Phase

As deep learning structure is defined as a sequence of layers, we create a sequential model and add layers one at a time until we are satisfied with the network topology. The input layer represents the number of data features that are extracted from the extracting phase as shown in figure 1. Then after multiple hidden layers, the output layer indicates whether the packet is normal or anomalous (DoS, Unsolicited and Cold Restart attacks).

As for the activation function, we use the Rectified Linear Unit (ReLU) which has become very popular in the last few years for logistic/continuous output, since it speeds up detection training [11]. For the hidden layer, generally we need a network large enough to capture the structure of the problem. In this study, we adapt a fully-connected network structure with five layers. The deep neural network...
consists of an input layer with 12 data features, 3 hidden layer around 16 neurons for each, and an output layer that represents the multi-class classification (normal, DoS, Cold Restart and Unsolicited) as output classes.

After fetching the input and train the network, we use different deep learning algorithms: RNN, LSTM, CNN, GRU, and FNN. The Feedforward neural network (FNN) is an artificial neural network wherein connections between the nodes do not form a cycle [12]. By using these deep learning algorithms for SCADA-IDS system, we investigate which model can be an effective and flexible IDS that has higher accuracy rate for detecting anomaly attacks.

3. Experiment and result

3.1. Experiment Setup
We conduct an experiment, the network of which consists of one DNP3 master and one outstation, using GNS3, which is a network software emulator [13]. The experiment utilizes two Linux hosts, which are connected to a switch with negotiated speed of 1000 Mbps. One is working as a master and the other working as an outstation, both of them running OpenDNP3 [14]. We also add an attack host used for penetration testing and network security assessments, and we assume that an attacker already gained access to the DNP3 network and the Kali Linux node is used to perform all further malicious activities [15]. The experiment focuses on the following attacks: DoS, disable unsolicited messages, cold restart, and packet injection and modification. For the cold start and packet injection/modification attacks, we execute the man-in-the-middle (MITM) attack by ARP spoofing. After the success of the MITM attack, the packets are sniffed, and altered by using the scapy library. Figure 2 shows DNP3 network communication in the experiment setup.

3.2. Dataset for attacks
To generate the attack traffic, we assumed that an attacker (Kali Linux) has already compromised the network. For the DoS attack, hping3 is used to generate and sends DoS traffic to port 20000 (the DNP3 port) of the outstation node [16]. In order to perform disable unsolicited messages attack and cold restart attack, we need to set up the Man-in-the-Middle (MITM) attack by arpspoof [17]. After the success of the MITM attack, we write a Python TCP hijacking script which uses scapy library to manipulate the frame to simulate the attacks of packet injection and modification [18]. For this
experiment, the intercepted frame coming from the master node to the outstation was altered from being a read class frame to a cold restart packet or disable unsolicited messages command to compromise normal operation of the outstation node.

![Figure 2. DNP3 Communication](image)

3.3. Results

After training the neural network on the prepared dataset using Tensorflow and Keras, we evaluate the performance of the detection model on the same dataset, which provides the accuracy and the loss. The figures 3 and 4 show the results of accuracy and loss, depending on the different deep learning models.

As shown in figure 3, the accuracy is increasing as number of steps (epochs) is increasing until it reaches approximately 98.75% of accuracy, which means that there is a chance of around 98% of detecting any anomalous traffic inside the network by these deep learning algorithms. It is good to note that the neural network performed very well while training, judging from the fact that from 0 to 10 epochs the accuracy reached approximately 98% of detecting.

Another metric to be considered is the loss. Lower the loss means the better the model is. Figure 4 shows the loss upon training the models. Similar to accuracy, loss will decrease as number of epochs increase till it reaches a value of 0.1, which is almost negligible loss at the end of the training.

![Figure 3. Model accuracy during the training of the network.](image)

![Figure 4. Model loss during the training of the network.](image)

4. CONCLUSION AND FUTURE WORK

We propose a deep learning-based approach to build an effective and flexible IDS. The main contribution of this work is to show comparison of performance of detection models based on different deep learning models. The attack of major concern is data injection and modification attack using the man-in-the-middle attack, in addition to other types of attack. As a further work, we need to expand
the detection model, which can detect unseen and unexpected attacks. In addition, we need to validate and evaluate IDS based on deep learning models by comparing with the traditional rule-based or protocol-based IDSs.

5. References

[1] Liu H and Lang B 2019 Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey, applied sciences, Vol 9, No 4396
[2] Perez R, Adamsky F, Soua R and Engel T, 2019 Forget the Myth of the Air Gap: Machine Learning for Reliable Intrusion Detection in SCADA, EAI Endorsed Transactions on Security and Safety
[3] Alrawashdeh K and Purdy C 2016 Toward an online anomaly intrusion detection system based on deep learning, Proc. IEEE International Conference on Machine Learning and Applications (ICMLA) pp 195-200
[4] Yang, H, Cheng, L and Chuah, M C 2019 Deep-Learning-Based Network Intrusion Detection for SCADA Systems, Proc. IEEE Conf. on Communications and Network Security (CNS)
[5] Gao J, Gan L, Buschendorf F, Zhang L, Liu H, Li P, Dong X and Lu T 2019. Omni SCADA Intrusion Detection Using Deep Learning Algorithms, arXiv:1908.01974v1
[6] Kwon S, Yoo H and Shon T 2019 RNN-based Anomaly Detection in DNP3 Transport Layer, Proc. IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)
[7] IEEE Standard for Electric Power Systems Communications Distributed Network Protocol (DNP3), IEEE Standard Association, IEEE Std 1815-2012
[8] Volkova A, 2019 Security Challenges in Control Network Protocols: A Survey, IEEE Com. Surveys & Tutorials, Vol 21, No 1, pp 619-639
[9] East S, Butts J, Papa M, and Shenoi S, 2009 A Taxonomy of Attacks on the DNP3 Protocol, Critical Infrastructure Protection III, IFIP AICT 311 chap 5 (Springer, Berlin, Heidelberg), pp 67-81
[10] Igbe O, Darwish I and Saadawi T 2017 Deterministic Dendritic Cell Algorithm Application to Smart Grid Cyber-Attack Detection, Proc. IEEE Int. Conf. on Cyber Security and Cloud Computing, pp 199-204
[11] Karlijn Willems 2017 Keras Tutorial: Deep Learning in Python, [Online]. Available: https://www.datacamp.com/community/tutorials/deeplearning-python
[12] Aslam, M, Lee, J.-M, Kim, H.-S, Lee, S.-J and Hong, S, 2020 Deep Learning Models for Long-Term Solar Radiation Forecasting Considering Microgrid Installation: A Comparative Study, Energies, Vol 13, No 147
[13] Welsh C, GNS3 network simulation guide. Packt Publ. 2013
[14] OpenDNP3, [Online]. Available: https://www.automatak.com/opendnp3
[15] Linux, Kali, [Online]. Available: https://www.kali.org
[16] hping3, [Online]. Available: http://www.secdev.org/projects/scapy
[17] arpspoof, [Online]. Available: http://www.secdev.org/projects/scapy
[18] Scapy, [Online]. Available: http://www.secdev.org/projects/scapy

Acknowledgments

This work was supported by “Human Resources Program in Energy Technology” of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea. (No. 20174030201790). And this research was also supported by Korea Electric Power Corporation. (Grant number: R18XA01).