Intelligent Anomaly Detection System for Critical Network Infrastructure

1st Boo Geum Jung
Defense ICT Convergence Research Sec.
ETRI
Daejeon, South Korea
bgjung@etri.re.kr

2nd Jinhyuk Yim
Defense ICT Convergence Research Sec.
ETRI
Daejeon, South Korea
jhyim@etri.re.kr

3rd Yoon-Sik Yoo
Defense ICT Convergence Research Sec.
ETRI
Daejeon, South Korea
ys5315@etri.re.kr

4th KangWoon Hong
Defense ICT Convergence Research Sec.
ETRI
Daejeon, South Korea
gwhong@etri.re.kr

5th Jongkuk Lee
Defense ICT Convergence Research Sec.
ETRI
Daejeon, South Korea
raphael@etri.re.kr

6th HeaSook Park
Defense Safety Convergence Div.
ETRI
Daejeon, South Korea
parkhs@etri.re.kr

***Abstract*—With the development of ICT technology, attempts to apply artificial intelligence to cyber security are increasing. The most relevant applications are intrusion detection systems. Traditionally, the intrusion detection system is based on known signatures. However, as attack techniques become more sophisticated, detecting unknown attacks is becoming an important issue. It is especially needed for critical network infrastructure where network reliability is crucial. Therefore, in this paper, we describe an Intelligent Anomaly Detection System(IADS) that learns the normal state to find and remove abnormalities different from the normal state. First, features are extracted from network traffic. Next, an anomaly detection function using an Autoencoder, an unsupervised learning-based algorithm learns without labels and discriminates the traffic. We validate the performance of developed systems by generating labeled datasets. Anomaly detection results are notified to the network management system so that the infrastructure can remain secure.**

***Index Terms*—network anomaly detection, mission-critical infrastructure, autoencoder, network feature extraction**

I. Introduction

As more and more devices connect to networks, the risk of exposure to cyberattacks increases. Network domains can be divided into three categories: Internet, Internet of Things, and Critical Infrastructure. The area where cybersecurity is most needed is Critical Infrastructure which is highly interconnected and has high demands on security [1].

As cyberattack techniques become increasingly sophisticated, existing cybersecurity solutions fall short of detecting and mitigating new cyberattacks. In particular, it can detect known attacks, but not unknown attacks. In order to solve these problems, attempts are being made to introduce artificial intelligence technology that predicts intrusion through learning in cyber security [2] [3].

Existing research has focused on improving the accuracy of learning by using features extracted from open datasets [4] [5]. In this paper, we propose a framework that can be used in field systems by extracting features from network traffic, discriminating normal/abnormal by applying a learning algorithm, and blocking abnormal traffic.

Following the introduction, Section 2 proposes the IADS system model, and Section 3 describes the design and implementation of the IADS system. Section 4 shows the verification results of the IADS and concludes in Section 5.

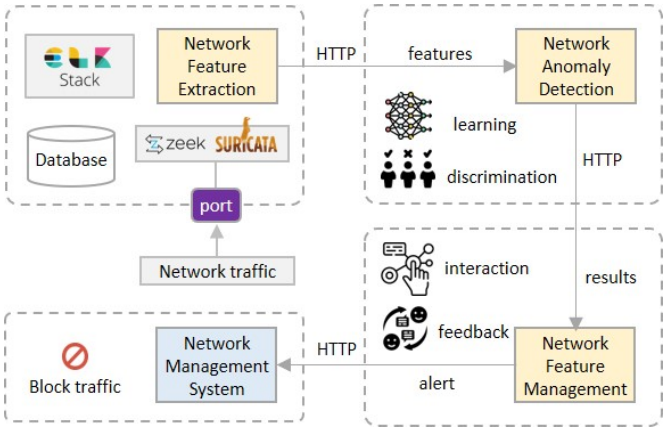

Fig. 1. Architecture of Intelligent Anomaly Detection System(IADS).

II. System Model

Section 2 presents the system model in Fig. 1. It consists of three functional blocks Network Feature Extraction, Network Anomaly Detection, and Network Feature Management. Each block sends and receives messages via HTTP. Zeek and Suricata are used for feature extraction and the ELK stack is used for log indexing and visualization.

A. Zeek

Zeek is an open-source network traffic analyzer. It records network activity at a high level and leaves log files. Extract the features of network traffic from the log files, which contain all logs for network connections.

B. Suricata

Suricata is an open-source intrusion detection system(IDS). Traffic is detected and logged if it matches a known signature. We can read intrusion information from the fast.log file and combine it with Zeek's conn.log file to determine which traffic corresponds to an intrusion.

C. ELK Stack

"ELK" is an acronym for three open-source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that collects data from multiple sources simultaneously, transforms it, and sends it to a "stash" like Elasticsearch. Kibana allows users to visualize data in Elasticsearch using charts and graphs. We can easily monitor Zeek's conn.log and Suricata's fast.log by indexing them in Elasticsearch and visualizing them in Kibana.

III. Design & Implementation

Chapter 3 describes the design and implementation of IADS. As a testbed environment, the IADS, network management system, and network gateway were connected through switches, the enterprise server was connected to the gateway, and the user device was connected as an attacker as shown in Fig. 2.

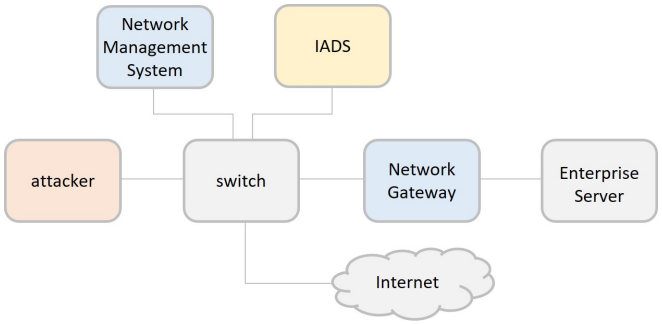

Fig. 2. Design & Implementation environment setup.

A. Network Feature Extraction

We index Zeek's conn.log in Elasticsearch to extract single traffic features and a certain number of packet window features. And we also index Suricata's fast.log file and compare it to the conn.log file to search the intrusive traffic. Finally, as shown in Table I, 8 single packet features, 5 packet window traffic, and 5 IDS features were extracted.

Since it is difficult to generate a lot of traffic in a real environment, a dataset was created by performing tcpreplay command on network ports using the MACCDC open dataset and extracting features from actual traffic. The size of the extracted dataset is shown in Table II.

TABLE I
Dataset Features: 15 numeric and 3 categorical

Category	No	Features	Type
Single Packet Features	1	protocol_type	object
	2	service	object
	3	flag	object
	4	origin_bytes	int64
	5	response_bytes	int64
	6	origin_packets	int64
	7	response_packets	int64
	8	duration	float64
Window Features	9	inbound	int64
	10	unique_destinations	int64
	11	unique_source	int64
	12	same_destinations	int64
	13	same_sources	int64
IDS Features	14	high	int64
	15	medium	int64
	16	low	int64
	17	ids_destination	int64
	18	ids_source	int64

TABLE II
Dataset Size

Label	Train data	Test data	Total
Normal	174,773	43,694	218,467
Abnormal	-	1,533	1,533
Total	174,773	45,227	220,000

B. Network Anomaly Detection

Network anomaly detection works best by learning normal states and predicting an anomaly when a state different from the learned normal state is detected. This is because not all threats can be learned, and new threats may emerge even if all known threats are learned. In addition, if an overfitting model is created by learning all known threats, it falls into the error of being able to detect only the learned threats.

Autoencoder takes input values and predicts input values. It is trained to approximate the identity function defined as

$$f_{W,b}(x) \approx x,\ f : neural\ network,\ W : weight,\ b : bias. \quad (1)$$

The learning process is shown in Fig. 3. Categorical features are one-hot encoded and numeric features are normalized with StandardScaler. Original data is reconstructed by applying a 3-layer encoder with 64, 32, and 16 neurons, a compressed data layer with 8 neurons, and corresponding decoder layers. Opti-

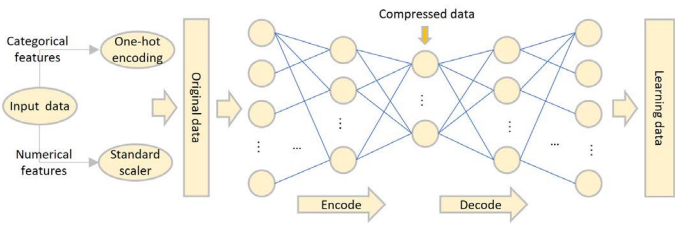

Fig. 3. Learning Process.

mize the parameters of the model to minimize reconstruction error defined as

$$L(x,\ x^{'}) = \|x - x^{'}\|^2. \tag{2}$$

C. Network Feature Management

While traditional intrusion detection systems only detect, the critical network infrastructure must quickly stop threats and stabilize the infrastructure. Therefore, Network Feature Management receives the learning result and alerts the network management system to block that traffic.

IV. Verification Results

This chapter describes the verification results of our IADS system.

A. Experimental Environment

Table III shows the hardware and software specifications for systems equipped with IADS. Network Feature Extraction and Network Anomaly Detection function were developed with Python Language, and the Network Feature Management function was developed with Java Language.

TABLE III
Test System Environment

Items	Specification
Processor	12th Gen Intel Core i9-12900HX, 14 Core
OS	Ubuntu 22.04LTS
Graphic	NVIDIA GeForceRTX 3080Ti GPU
Memory	32GB DDR5-4800MHz
SW	Python 3.7, Tensorflow 2.7, Cuda 12.0, JAVA

B. Test Results

Accuracy, precision, and recall were measured as the main performance indicators of anomaly detection.

Accuracy: shows the ratio of true detection over total transactions and calculated as

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}. \tag{3}$$

Precision: shows how many predicted intrusions by AIDS are actual intrusions,

Recall: it is the ratio of predicted intrusions versus all intrusions presented and calculated as

$$Precision = \frac{TP}{(TP + FP)},\ Recall = \frac{TP}{(TP + FN)}. \tag{4}$$

F1-score: is a harmonic mean of Recall and Precision and gives a better measure of the accuracy and is calculated as

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{5}$$

As shown in Table IV, each performance metric was measured for three feature categories: Single Packet features, Single Packet + Packet Window + IDS features, and Single Packet + IDS features. Fig. 4 shows one of the training results.

TABLE IV
Test Results using Different Features

Features	Accuracy	Precision	Recall	F1 score
Single	99.3%	90.3%	90.3%	90.3%
Single+Window+IDS	99.6%	94.8%	94.7%	94.7%
Single+IDS	99.7%	95.2%	95.2%	95.2%

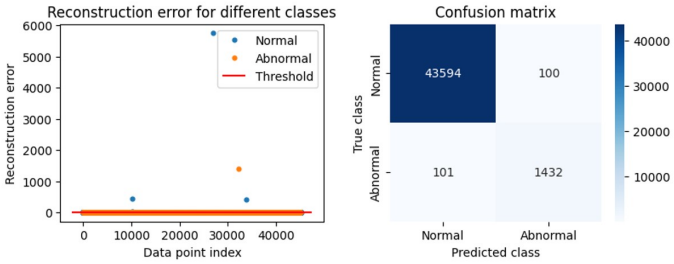

Fig. 4. Test results using Autoencoder Unsupervised Learning.

Conclusion

This paper discussed an intelligent anomaly detection system for critical infrastructure. All functions for anomaly detection - feature extraction, learning, and alert to block malicious traffic are developed for practical use. It can be seen that the single packet feature alone gives a good performance of over 95% of single+IDS features. In the future, we plan to build various datasets and leverage other open datasets to further improve performance.

Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korean government(MSIT) (2021-0-00040, Development of intelligent stealth technology for information and communication resources for public affairs and missions)

References

[1] Zeadally, Sherali; Adi, Erwin; Baig, Zubair; Khan, Imran (2020): Harnessing artificial intelligence capabilities to improve cybersecurity. Deakin University. Journal contribution. https://hdl.handle.net/10536/DRO/DU:30134262.

[2] J. Ish, Daniel, Jared Ettinger, and Christopher Ferris, Evaluating the Effectiveness of Artificial Intelligence Systems in Intelligence Analysis. Santa Monica, CA: RAND Corporation, 2021. https://www.rand.org/pubs/research_reports/RRA4641.html. Also available in print form.

[3] Wang, Song & Balarezo Serrano, Juan Fernando & Sithamparanathan, Kandeepan & Al-Hourani, Akram & Gomez Chavez, Karina & Rubinstein, Ben. (2021). Machine Learning in Network Anomaly Detection: A Survey. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3126834.

[4] Mohammad Kazim Hooshmand, Doreswamy Hosahalli (2022): Network anomaly detection using deep learning techniques. CAAI Transactions on Intelligence Technology 7 (2), 228-243. https://doi.org/10.1049/cit2.12078.

[5] Xu, Wen; Jang-Jaccard, Julian; Singh, Amardeep; Wei, Yuanyuan; Sabrina, Fariza (2021): Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset. CQUniversity. Journal contribution. https://hdl.handle.net/10779/cqu.17072780.v1.