A Lightweight and Intelligent Intrusion Detection System for Integrated Electronic Systems

Daojing He, Xiaoxia Liu, Jiajia Zheng, Sammy Chan, Sencun Zhu, Weidong Min, and Nadra Guizani

Abstract

With the recent advancement in AI technology, IDSs reinforced by AI have been adopted to ensure system and network security. Unfortunately, computational and storage overheads of such systems prevent them from being deployed in IESs. To overcome the challenges that restrict the applicability of intrusion detection for IESs, we propose a lightweight and intelligent IDS. The proposed system first generates the behavioral specifications that characterize the normal communication of an IES. Based on specifications, Gini index and the generalized system attributes are exploited to detect abnormal communications. To reduce the false positive rate, the data that do not abide the behavioral specifications is passed to a Naive Bayes classifier for further classification. Our experimental results show that the proposed system can achieve 95.84 percent accuracy and thus holds great promise for deployment in IESs as a lightweight and efficient IDS.

Introduction

With the advancement of integration and generalization of electronic equipment, integrated electronic systems (IESs) emerged and were first applied to space projects, such as spacecrafts and communication satellites. Since then, they have been widely used in military fields, such as fighter jets, armored vehicles, radar, and so on. In these systems, computer networking technology is used to interconnect various independent modules, such as subsystems and central management units, to achieve information sharing, unified management and other functions.

Despite the critical importance and increasing popularity of IESs, the security of most IESs has not received enough attention. Due to the diversification of their electronic devices, the standardization of platforms, the openness of technology and the lack of encryption protection over internal communication, IESs are facing serious security threats. For IESs working on the ground, such as vehicle IESs, the attack source can be connected to them through physical attacks [1]. For IESs operating in space, such as spacecraft IESs, although they are impossible to be attacked physically, they can be compromised when their ground station is controlled by the Advanced Persistent Threat (APT) attack. The occurrences of related security incidents are frequent. For example, in civil aviation industry, on June 21, 2015, Polish Airlines was hacked, causing the system to crash for five hours and leaving more than 1,400 passengers stuck in the Frederic Chopin Airport [2]. In the area of communications satellites, Israel's main satellites were attacked, resulting in the transmission channel being controlled by malware [3]. The US-China Economic and Security Review Commission stated in the 2011 Congressional Report that "an attacker may use cyber activities to disrupt, deceive or damage space systems, or to exploit or attack ground infrastructure, space-based systems or communications links between them" [4].

Integrated electronic systems are usually deployed in physically isolated networks. Since the advent of the stuxnet virus, researchers have gradually conducted security research on physically isolated networks [5]. Intrusion detection technology is the key to security protection, which can effectively detect internal attacks, external attacks and malfunctions, fundamentally improving the security of systems. However, there are currently few intrusion detection systems (IDSs) for IESs. With the rapid development of artificial intelligence (AI), intelligent IDS using machine learning methods can learn system behavioral characteristics that are not recognized by traditional methods, thereby improving detection accuracy. However, due to the peculiarity of the IESs, the design of lightweight and intelligent IDSs faces the following challenges:

- IESs have limited hardware resources, and are limited by power consumption and size, which means that their computing and storage resources are also limited. These limited resources need to be first allocated to the management software of the system, before being made available for intrusion detection.
- IESs use different bus protocols that are designed according to the requirements of different application scenarios. Therefore, it is necessary to study the self-learning intrusion detection method suitable for IESs in different application scenarios, to avoid manual frequent redefinition of standards.

Daojing He is with East China Normal University and Nanchang University; Xiaoxia Liu and Jiajia Zheng are with East China Normal University; Sammy Chan is with City University of Hong Kong; Sencun Zhu is with Pennsylvania State University; Weidong Min is with Nanchang University; Nadra Guizani is with Purdue University.

Digital Object Identifier:
10.1109/MNET.001.1900480

0890-8044/20/$25.00 © 2020 IEEE

It is not practical to deploy deep learning-based intrusion detection methods on IESs. Due to the complexity of the model in deep learning, the time complexity of the algorithm increases sharply. In order to ensure the real-time performance of the algorithm, higher parallel programming skills and better hardware support are needed.

To this end, we propose a lightweight and intelligent intrusion detection method (named LI-IDS) for IESs. The method combines self-generating behavioral specification and Naive Bayes classifier to improve the accuracy of detection. Most of the existing intelligent IDSs, such as those using artificial neural networks, clustering, data mining and artificial immunity, require complex and advanced AI algorithms to process *all* the data, which is costly for resource-constrained IESs. In contrast, in the first step of our approach (i.e., with behavioral specification), most of the data can be easily processed by matching the behavioral specification, leaving very little processing load to the Naive Bayes classifier. Therefore, our approach requires less computing power than existing intelligent IDSs. In addition, compared to existing behavior-based methods, our method significantly reduces the false positive rate by adding a Naive Bayes classifier.

The rest of the article is organized as follows. The following section briefly overviews intrusion detection and existing work on intrusion detection for IESs. Then we introduce the background knowledge of IESs and preparation for self-generating specifications. Following that, we present the security model and our proposed LI-IDS. Then we present the evaluation and comparison results of LI-IDS. The final section concludes the article.

Related Work

Beginning with the first IDS proposed by Denning [6], IDSs have been developed for many years and can be divided into two categories: *misuse detection* and *anomaly detection*.

The misuse detection methods use a database of known signatures and patterns of intrusions to detect well known attacks. Network packet overload, high cost of signature matching, and large number of false alarms are three disadvantages of misuse-based IDSs [7]. In addition, the severe memory constraints in some types of networks, such as wireless sensor networks, result in low performance of misuse-based IDSs because of their need to store a large database of attack signatures. At present, there are many misuse detection methods using machine learning to find the effective attack characteristics [8].

Anomaly-based intrusion detection methods mainly focus on describing the normal behavior of the system, and detecting the abnormal behavior based on the deviation from the behavioral specification. It performs well in detecting unknown attacks, but may suffer from a higher false positive rate (FPR). Anomaly detection methods can be further classified as statistical-based [9], knowledge-based [10] and machine learning-based [11]. For example, machine learning-based methods mainly train and optimize the learning model to form the detection model according to the normal behavior or data of the system or network. Common machine learning methods include Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), Neural Network (NN), Hidden Markov Model (HMM), and so on. Deep learning is an advanced subset of machine learning. Thus far, initial deep learning research has demonstrated that its superior layer-wise feature learning can better or at least match the performance of shallow learning techniques. However, it is not practical to deploy deep learning-based intrusion detection methods on IESs. Due to the complexity of the model in deep learning, the time complexity of the algorithm increases sharply. In order to ensure the real-time performance of the algorithm, higher parallel programming skills and better hardware support are needed.

So far, much IDS research has been done in the Internet setting, but few on IESs. McGraw [12] demonstrated the effect of abnormal behavior on the operation of the satellite when subsystem communication is performed on the 1553B communication bus. They simulated the normal behavior of IESs, manually analyzing the interference of several malicious behaviors on the system. Nguyen [13] made a threat analysis on the 1553B bus in the IESs. The proposed attack scenarios are divided into timing attacks and storage attacks. Timing attacks deploy the time delay between messages defined by MIL-STD-1553, while storage attacks utilize word structures and programmer-defined functions. However, the proposed attack scenario is only hypothetical with certain assumptions, and may not be realistic. Stan *et al*. [14] proposed to detect intrusions by detecting the periodic characteristics of message frames in the bus of the electronic system, specifically using network layer message addressing and Markov chain to predict the probability that one message address will follow another message address. This method is very effective when detecting the insertion of an error message into the bus or a denial of service (DoS) attack. However, if an attacker masquerades as a subsystem on a bus and communicates on the bus at expected time intervals, the method will not detect such malicious behavior.

Background Knowledge and Security Model

IES Architecture

IESs typically use a distributed architecture of layered interconnections, as shown in Fig. 1. The central management unit (CMU) is responsible for managing the subsystems, and they can communicate with each other via the primary bus. The subsystem control center (SCC) manages the corresponding sensors, which are connected by a secondary bus. The subsystem can control the sensors through the bus. Typically, the primary bus is 1553B bus.

The 1553B bus specifies three types of data units: command word, data word and status word. The format of the command word is described here since we will extract features from the command word in the experiment. The command word has 20 bits, which is composed of synchronous head, remote terminal address, R/T bit, subaddress/mode, data word size/mode code, and parity bit. The remote terminal address is the terminal identity which should receive and execute the command. The R/T bit indicates wheth-

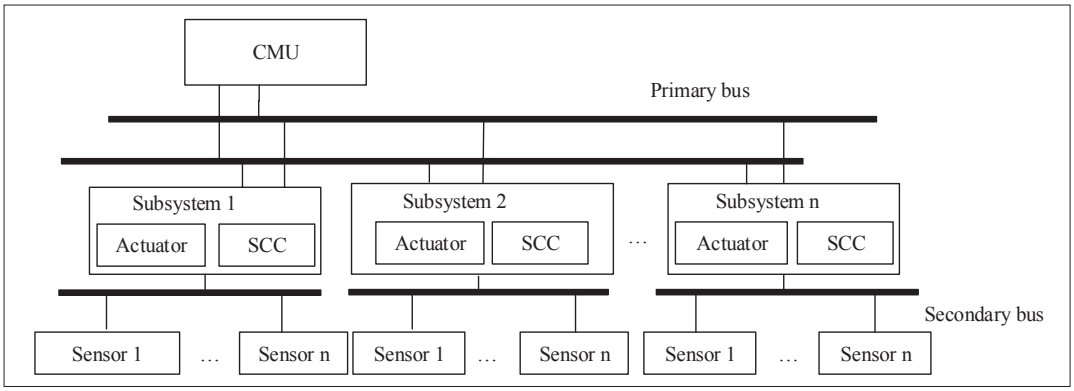

FIGURE 1. System architecture of an IES.

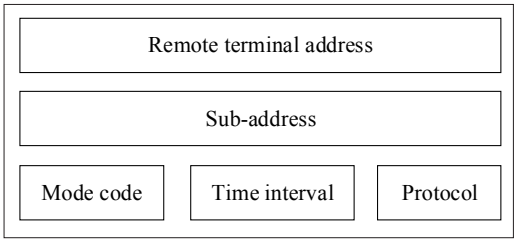

FIGURE 2. The attribute hierarchy of integrated electronic system.

er the terminal needs to receive (R) or transmit (T) data. Each terminal has 32 sub-addresses for storing data words; each address can store 64 bytes of data. The mode code is used to control the 1553B bus remote terminal. The data word size indicates how many data words the remote terminal needs to transmit or receive. The method code is used to indicate the specific control mode to control the remote terminal.

There are two communication modes in the 1553B bus, one between the Bus Controller (BC) (corresponding to the CMU in IES) and the Remote Terminal (RT) (corresponding to subsystem in IES), and the other between RT and RT.

Between BC and RT: There are two ways for BC and RT to communicate. First, when BC sends data to RT, BC transmits the configured "receive" command word through the bus to RT, which indicates RT to receive data, and then sends a data word. RT receives the message and responds with a status word. Second, when BC needs RT to send the data calculated by its subsystem, BC transmits the configured "send" command word through the bus to RT. Upon receiving the command, RT responds with a status word, followed by the data word.

Between RT and RT: Assuming RT2 sends data to RT1, BC first transmits a "receive" command word to RT1, and transmits a "send" command word to RT2. Upon receiving the command word, RT2 responds with a status word, followed by the data word. RT1 receives the data and responds with a status word.

Gini Index

For a sequence L, its elements are discretely unordered and can take the same value. Here L is a sequence of attribute values of the IES. Combining the same elements in L, $D = < (w_1, c_1), (w_2, c_2), \cdots, (w_M, c_M) >$ can be obtained, where M is the number of different values in the sequence L and w_m is the value that appears in L and c_m is the number of times the value appears. The Gini index of D, $Gini(D)$, is given by:

$$1 - \sum_{m=1}^{M} \left(\frac{|c_m|}{|D|} \right)^2.$$

The Gini index of the sequence reflects the randomness (uncertainty) of the element probability distribution in the sequence. The smaller the Gini index, the higher the purity of the sequence.

Generalization

A network IDS generally adopts either the state transition analysis method or the statistical method to generate specifications. For the state transition analysis method, it requires the generation of state sets, state variables, initial state, termination state, transition rules for different protocols, which incur a high consumption of storage and computational resources. For the statistical method, intrusion is identified by the quantity of specific characteristics. If the quantity exceeds a certain range or threshold, the abnormality is identified. Because of the periodicity and regularity of the commands in the IES, the statistical method is more appropriate to generate behavioral specification automatically and efficiently. If the bus protocol of the IES is different, the relevant attributes that represent the system behavior can be redefined according to the characteristics of the system and protocol specification.

First, the attributes of the IES are defined. In this article, it is believed that the sub-addresses under each remote terminal address have their own behavioral patterns. For example, the on-off state telemetry subsystem is accessed every cycle, while the thermal control subsystem is accessed every two cycles. Therefore, the attributes of each sub-address can be defined. Figure 2 shows the defined attribute hierarchy of an IES. The remote terminal address is the parent node of the sub-address, which is the parent node of the three attributes including mode code, time interval and communication protocol. Figure 3 shows the generalization process of the self-generating behavioral specification, where the first four lines correspond to normal system data, and the fifth line is the generalization result.

Different generalization conditions may be set for different attributes. Here, we adopt the Gini index of sequence as generalization conditions.

Remote terminal address	Sub-address	Mode code	Time interval	Protocol
02	03	01001	0.8ms	1553B
02	03	01001	0.8ms	1553B
02	03	01001	1.0ms	1553B
02	03	01001	1.1ms	1553B
↓				
02	03	01001	0.8-1.1ms	1553B

FIGURE 3. Generalization process of self-generating behavioral specification.

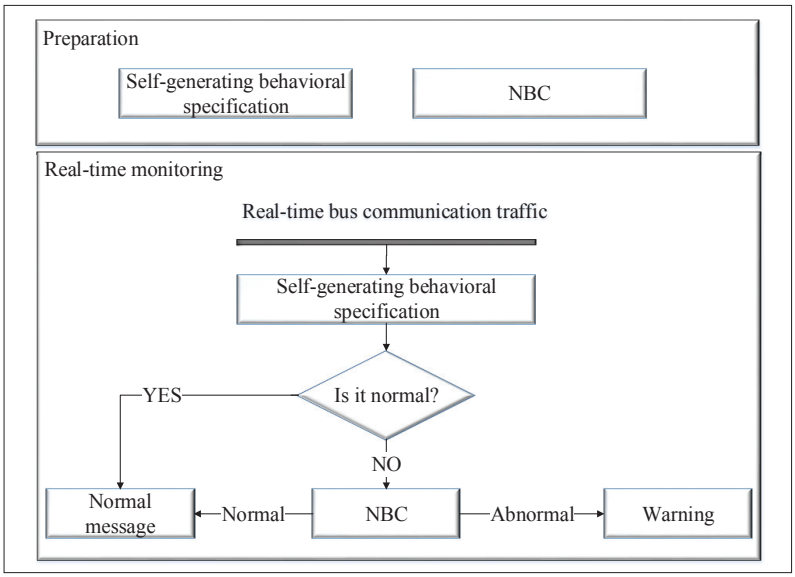

FIGURE 4. The two phases of LI-IDS.

When the Gini index of a sequence exceeds a predefined threshold, the generalization will be performed to the parent node; otherwise, no generalization will be performed. For example, if the predefined threshold of Gini index for the sequence is 0.7, the Gini index of the sequence $L = < 01, 02, 03, 04 >$ is 0.75, generalization will be performed. On the other hand, if the Gini index of $L = < 01, 01, 01, 01 >$ is 0, no generalization is performed. Generalization requires threshold values, which can be chosen empirically.

Security Model

The main security objectives of IESs are confidentiality, availability and integrity. Intrusion detection focuses only on availability and integrity. Confidentiality can be accomplished by access control and encryption techniques. Our work mainly aims at destructive attacks against integrity and availability. The purpose of physical attack, APT attack, backdoor or logic bomb is to change the normal behavior of a system, which can be reflected from data. Thus, our method only focuses on detecting two kinds of attacks:

- The attacks that compromise the data integrity, including tampering attacks, forgery attacks, replay attacks.
- The attacks that compromise the data availability are DoS attacks.

The following examples illustrate the manifestations of forgery attacks, tamper attacks, replay attacks and DoS attacks.

Forgery Attacks: Aim to achieve the attacker's purpose by forging data as the BC or a RT. For example, on a spacecraft, an attacker can forge a heater turn-on command word and send it to a remote terminal that manages the thermal control subsystem. With this command, the heater that should be turned off will be put into the on state, leaving the spacecraft in a dangerous state.

Tampering Attack: Refers to the purpose of tampering with the system data through the forged command word. For example, in the IES, there is a timing command to synchronize the clocks of the respective RTs. If the attacker forges the timing command, the time clock in the system can be changed.

Replay Attack: Means that past data is re-injected into the system. For example, when the battery is not charging, the attacker could inject an earlier battery charging command, which will change the state of the battery, causing potential serious consequences. To some extent, a replay attack shares certain similarity with a forgery attack, but in the latter, an attacker may forge data freely, while replay attacks only use the past data of the system.

DoS Attack: Refers to disable normal operation of BC, RT, or the entire bus. For example, if a forged command word is sent to an RT at a relatively high frequency, the RT will not be able to execute the command from the legitimate BC, hence being denial of service.

Our Proposed Method

System Overview

Figure 4 depicts an overview of our detection system. It is divided into two phases: *preparation phase* and *real-time detection phase*. In the preparation phase, two modules are trained in advance. First, based on statistical generalization, the system uses a large amount of normal traffic to self-generate normal behavioral patterns. Second, normal data and abnormal data are used to train a Naive Bayes classifier. In the real-time detection phase, the real-time communication traffic on the bus is obtained for traffic analysis. The traffic analysis mainly derives the characteristics required for the detection, which is then matched against the normal behavioral specification. If the matching is successful, the traffic data enters the normal communication; otherwise, the Naive Bayes classifier is further employed to classify it. If the result of classification indicates that the traffic is abnormal, our detection system will generate a warning message and log a record of the intrusion.

Detailed Design

LI-IDS comprises two modules: the Self-generated Behavior Specifications Generator (SBSG) and Naive Bayes Classifier (NBC). The SBSG generalizes the statistics from the normal network traffic as a concise presentation. If traffic fails to match the self-generated behavior specifications, it will be sent to the NBC, which serves as the second-layer of defense. The NBC module aims to reduce the false positive rate given the reduced amount of data.

The generation of behavior specification is based on the following assumption: the normal traffic of the system can be collected to generate behavioral specification when the IES is running in a safe state (i.e., the system is free from any attack, such as backdoor and logic bomb).

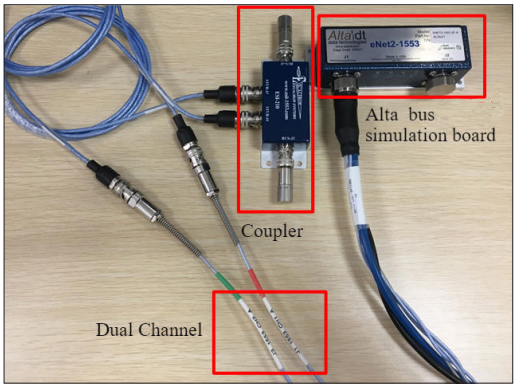

FIGURE 5. Experiment platform.

The assumption is based on the following two points: first, attacks such as backdoors and logic bombs generally have a longer latency, so attackers generally do not trigger these attacks on the ground; second, if those attacks are triggered on the ground, some abnormalities can be discovered and dealt with in time through manual intervention.

Specifically, the process of producing the self-generating behavior specifications is as follows.

Define the Properties of the IES: First, select a set of system attributes of the IES based on the system's configuration file, bus protocol, or system-specific attributes. To optimize the self-generated specification, the attribute level and its corresponding Gini index threshold are specified.

Generate Behavioral Specifications: For each attribute $Attr_i$, obtain its normal data, add into the sequence L, combine the same elements in L and calculate the Gini index of L. If the Gini index of L is greater than the predefined threshold value, the sequence value will be generalized into an interval; otherwise, L will be converted into a data set. The interval or set is the behavior specification of the attribute $Attr_i$.

Naive Bayes Classifier: The Naive Bayes classifier [15] is a classification method based on Bayesian theorem and feature condition independence hypothesis. First, the joint distribution of the input or output is learned based on the hypothesis of feature independence, and then the output y of the posterior probability maximization is solved by the Bayesian theorem according to the model. The entire Naive Bayes classification is divided into three phases:

- First phase — preparation. This phase identifies feature attributes that can help distinguish attack traffic from normal traffic, such as the three feature attributes mentioned earlier, and then form a training set.
- Second phase — classifier training. The purpose of this phase is to generate a classifier. The main task is to calculate the frequency of occurrence of each category in the training samples and the conditional probability estimates for each category of feature attributes. When the feature attribute is a continuous value, it is assumed that the value obeys a Gaussian distribution. The required estimated value is obtained by calculating the mean and standard deviation of the feature items in each category in the training sample.
- Third phase — application. The purpose of this stage is to classify the items using the classifier. The output is the mapping relationship between the input item and the category.

Evaluation

We ran several experiments with an IES to demonstrate the effectiveness of LI-IDS. Figure 5 shows the experiment platform. The simulation of an IES is based on the Alta (Alta Data Technologies) interface card. The Alta board is connected to a PC equipped with an i7-7500 3.41GHZ processor, 8GB RAM, and the 64-bits Windows 10 Operating System. The Alta bus simulation board uses a two-channel structure, in which one channel is used for normal system communication, while the other plays the role of an attacker. A coupler is used to connect two channels so that the source of an attacker is connected to the normal communication channel. The two channels of the 1553B bus simulated by Alta are double redundant channels.

First, based on the specifications of the IES, the system attributes are defined as: remote terminal address, sub-address, mode code, time interval between command words, packet length, time interval between command words and state words of the same remote terminal address within a period. Since the actual attack data against the IES could not be obtained at present, we also carried out an attack experiment on the simulation platform by using the AltaView Bus Analyzer software through the attack channel. The attacks launched in our experiment are forgery attack, replay attack and DoS attack according to the attack scenario described above. We obtained 105,624 normal communication data records of the system through the BM module in AltaView Bus Analyzer, which was then used to self-generate behavioral specification. In the testing phase, after conducting the attack experiment, 223,865 data records were extracted, including 110,523 records of normal periodic data, 3,776 records of normal non-periodic data, 103,422 records of DoS attacks and 6,144 records of other types of attacks (such as forgery attack, tampering attack, replay attack).

We used a Python program to extract the defined attributes of IES from the acquired data, and divided them into the training set and the test set in the ratio of 7:3. The training set was used to train the Naive Bayes model and the test set was used to verify it. For a given test data set, *accuracy* refers to the proportion of the samples correctly classified by the classifier; *precision* is the ratio of the number of correctly predicted attacks to the total number of predicted attacks; *recall* is the ratio of the number of correctly predicted attacks to the actual number of attacks; *false positive rate (FPR)* is the ratio of the number of normal data predicted as attacks to the total number of normal data.

Concerning computing resources, the smaller the threshold and the larger the reasonable range of system attribute generalization, the less additional computing resources are required. In contrast, if the threshold value is too high and the value range of system attribute generalization is too small, more data need to be processed by the Naive Bayes classifier, leading to a waste of computing resources. Through many experiments,

Number	Method	Accuracy	Precision	FPR	Recall	Time	Amount of Data
1	LR	91.67 %	89.7 %	10.9 %	93.8 %	17.531ms	100 %
2	RF	94.74 %	93.5 %	5.5 %	97.4 %	89.965ms	100 %
3	SVM	93.73 %	91.6 %	4.7 %	97.3 %	215.6ms	100 %
4	NB	92.8 %	91.8 %	6.18 %	96.0 %	38.26ms	100 %
5	Self-generated behavior specification match	95.84 %	92.69 %	2.1 %	98.2 %	25.35ms	100 %
	NB					11.28ms	30 %

TABLE 1. Experimental results comparing four existing machine learning methods and our proposed method.

we found that when Gini index threshold was set to 0.9, the accuracy was the highest, but about 52 percent of additional data would need to be processed by the classifier. When Gini index threshold was 0.7, our method only needed to process 30 percent of additional data while reaching the accuracy of 95.84 percent and FPR of 2.1 percent. Compared to the standalone self-generated specification method, the accuracy was improved by about 11.4 percent and the FPR was reduced by about 17 percent. When Gini index threshold was further reduced to 0.5, the accuracy decreased more, but the additional data to be processed by the classifier almost remained the same. From this comparison, we chose 0.7 as the optimal threshold value.

Furthermore, we used various machine learning methods, including SVM, Random Forest (RF), Logistic Regression (LR) and Naive Bayes (NB) in the sklearn library to conduct the experiments, and we compared them with ours in terms of running time, accuracy, precision, FPR, recall and amount of data. From Table 1, it is clear that the classifiers of LR, RF, SVM and NB all used 100 percent of the data, RF and SVM had higher recall rate and accuracy, but took more time. RF can evaluate the importance of variables while determining the category, and hence balance the errors for unbalanced classified data sets. Therefore, its accuracy and recall are relatively high. However, because multiple decision trees need to be generated for evaluation, the running time and computational resources are also much more. SVM uses quadratic programming to solve support vectors, and involves the calculation of m-order matrix (m is the number of samples). When m is large, the storage and calculation of this matrix will consume a lot of memory and computation time. Our simulation system simulated only a few RTs and generated low complex data. When multiple highly complex data of information interaction is generated in the real system, the resource consumption of these two methods will be tremendous. LR and NB used less time, but the accuracy and precision are lower than RF and SVM. LR is suitable for dealing with near-linearly separable classification problems. Due to the burst of aperiodic messages, attacks were more difficult to distinguish, and bursty aperiodic messages affected the calculation of interval of periodic messages, which in turn affected the classification effect of LR. NB has a higher FPR although other indicators are satisfactory. The time interval feature involves continuous values, which LR and NB handled poorly, so there were false alarms. NB performed better than LR, because NB is a generation model, which can fit the data better according to prior probability, while LR is a decision model, which directly predicts output through training data and does not model the joint probability. If the size of data set increases and the feature dimension increases, the performance of LR will be improved.

Since the processing of data in our approach leverages the behavioral specification, which is a comparison of range of extracted feature values, the resource consumption is very low. Our evaluation also showed that the amount of data passed to the Naive Bayes classifier could be reduced by 70 percent, while the accuracy of detection could still reach 95.84 percent. Note that the detection rate of our method is higher than of all other methods. However, some attacks were not detected. Especially, for replay attacks, because the attack frequency in the training set is low, the attack detection rate is also relatively low.

Conclusion

In this work, we have presented a lightweight and intelligent IDS. Our approach uses Gini index and system attribute generalization to generate the behavioral specifications that characterize the normal communication behavior of the system. To achieve high accuracy, the traffic that does not conform to the behavioral specifications is passed to a Naive Bayes classifier for a second round of detection. Experimental results have shown that our approach is lightweight and efficient in detecting various attacks such as DoS attack, forgery attack and tampering attack. Our method is self-adaptive, which is also applicable to the resource-constrained devices in cyber-physical systems.

Acknowledgment

This research is supported by the National Key R&D Program of China (2017YFB0802805 and 2017YFB0801701); the National Natural Science Foundation of China (Grants: U1936120, U1636216, and 61762061); the Natural Science Foundation of Jiangxi Province, China (Grant No. 20161ACB20004); Jiangxi Key Laboratory of Smart City (Grant No. 20192BCD40002); Joint Fund of the Ministry of Education of China for Equipment Preresearch (No. 6141A020333); the Shanghai Knowledge Service Platform for Trustworthy Internet of Things (No. ZF1213); and the

Fundamental Research Funds for the Central Universities. Daojing He is the corresponding author of this article.

References

[1] A. Palanca *et al.*, "A Stealth, Selective, Link-Layer Denial-of-Service Attack Against Automotive Networks," *Proc. Int'l. Conf. Detection of Intrusions and Malware, and Vulnerability Assessment*, Bonn, Germany, June 4, 2017, pp. 185–206.
[2] R. Abeyratne, "Aviation Cyber Security: A Constructive Look at the Work of ICAO," *Air and Space Law*, vol. 41, no. 1, 2016, pp. 25–39.
[3] Y. Tang, Q. Chen, and M. Li, "Challenge and Evolution of Cyber Attacks in Cyber Physical Power System" *Proc. 2016 IEEE PES Asia-Pacic Power and Energy Engineering Conference (APPEEC)*, Xi'an, China, Oct. 28, 2016, pp. 857–62.
[4] US-China Economic and Security Review Commission, "2010 Report to Congress of the US-China Economic and Security Review Commission," U.S. Government Printing Office, 2010.
[5] I. Ghar *et al.*, "Detection of Advanced Persistent Threat Using Machine-Learning Correlation Analysis," *Future Generation Computer System*, vol. 89, no. 9, July 2018, pp. 349–59.
[6] D. E. Denning, "An Intrusion-Detection Model," *IEEE Trans. Software Engineering*, no. 2, 1987, pp. 222–32.
[7] W. Meng, W. Li, and L.F. Kwok, "EFM: Enhancing the Performance of Signature-Based Network Intrusion Detection Systems Using Enhanced Filter Mechanism," *Computers & Security*, vol. 43, 2014, pp. 189–204.
[8] A. Nagaraja and S. Aljawarneh, "PAREEKSHA: A Machine Learning Approach for Intrusion and Anomaly Detection," *Proc. First Int'l. Conf. on Data Science, E-learning and Information Systems*, Madrid, Spain, Oct. 01-02, 2018, pp. 36.
[9] S. Jose *et al.*, "A Survey on Anomaly Based Host Intrusion Detection System," *J. Physics: Conf. Series*, vol. 1000, no. 1, Apr. 2018, pp. 012049.
[10] K. A. Al-Utaibi and E. S. M. El-Alfy, "Intrusion Detection Taxonomy and Data Preprocessing Mechanisms," *J. Intelligent & Fuzzy Systems*, vol. 34, no. 3, 2018, pp. 1369–83.
[11] M. Z. Abedin *et al.*, "Performance Analysis of Anomaly Based Network Intrusion Detection Systems," *Proc. 2018 IEEE 43rd Conf. Local Computer Networks Workshops (LCN Workshops)*, Chicago, Oct. 01–04, 2018, pp. 1–7.
[12] R. McGraw *et al.*, "Cyber Threat Impact Assessment and Analysis for Space Vehicle Architectures," *Sensors and Systems for Space Applications VII*, vol. 9085, June 2014.
[13] D. Nguyen, "Towards MIL-STD-1553B covert channel analysis," Naval Postgraduate School Monterey CA, Jan. 2015.
[14] O. Stan *et al.*, "Protecting Military Avionics Platforms from Attacks on MIL-STD-1553 communication bus," ArXiv Preprint, July 2017.
[15] K. P. Murphy, "Naive Bayes Classifiers," Technical Report, Oct. 2006, available: http://www.cs.ubc.ca/murphyk/Teaching/CS340.Fall06/reading/NB.pdf.

Biographies

Daojing He [()S'07, M'13] received the B.Eng. (2007) and M. Eng. (2009) degrees from Harbin Institute of Technology (China) and the Ph.D. degree (2012) from Zhejiang University (China), all in computer science. He is currently a professor in the School of Software Engineering, East China Normal University, P.R. China. His research interests include network and systems security. He is on the editorial board of several international journals such as *IEEE Communications Magazine*.

Xiaoxia Liu was born in 1996. She is currently a master student in the School of Software Engineering, East China Normal University, P.R. China.

Jiajia Zheng is currently a master student in the School of Software Engineering, East China Normal University, P.R. China.

Sammy Chan (S'87-M'89) received his B.E. and M.Eng.Sc. degrees in electrical engineering from the University of Melbourne, Australia, in 1988 and 1990, respectively, and a Ph.D. degree in communication engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. Since December 1994 he has been with the Department of Electrical Engineering, City University of Hong Kong, where he is currently an associate professor.

Sencun Zhu is an associate professor in the Department of Computer Science and Engineering at Penn State University (PSU). He received the B.S. degree from Tsinghua University, the M.S. degree from the University of Science and Technology of China, and the Ph.D. degree from George Mason University in 1996, 1999, and 2004, respectively. His research interests include wireless and mobile security, software and network security, and fraud detection. He is the editor-in-chief of *EAI Transactions on Security and Safety*, and associate editor of *IEEE Transactions on Mobile Computing*.

Weidong Min (M'12) received the B.E., M.E. and Ph.D. degrees in computer application from Tsinghua University, China in 1989, 1991 and 1995, respectively. He is currently a professor and the Dean, School of Software, Nanchang University, China. He is Executive Director of the China Society of Image and Graphics. His current research interests include image and video processing, artificial intelligence, big data, distributed system and smart city information technology.

Nadra Guizani is a Ph.D. student and graduate lecturer at Purdue University, completing a thesis on prediction and access control of disease spread data on dynamic network topologies. Her research interests include machine learning, mobile networking, large data analysis, and prediction techniques. She is an active member in both the Women in Engineering program and the Computing Research Association for Women.