sciendo

Carpathian Journal of Electronic and Computer Engineering 14/2 (2021) 15-19

DOI: 10.2478/cjece-2021-0009

A survey for Communication security of the embedded system

Yu Xie
Faculty of Informatics
University of Debrecen
Debrecen, Hungary
yu.xie@inf.unideb.hu

Attila Buchman
Faculty of Informatics
Universityy of Debrecen
Debrecen, Hungary
buchman.attila@inf.unideb.hu

***Abstract*— The embedded operating system is a task-oriented computing platform that can be tailored, low-cost, and has high requirements for reliability and real-time performance. It plays an extremely important role in engineering applications. With the gradual application of embedded systems in various fields, the shortcomings of its insufficient ability to respond to security threats have gradually emerged, and many hackers have turned their attack targets into embedded systems. The important reason for these attacks is that embedded systems lack sufficient multi-layer protection mechanisms. This article focuses on the threats embedded systems face in terms of communication security. Then analyze the existing communication security-related technologies from the perspectives of the network layer, the transport layer, and the application layer. Finally, it summarizes the research direction of embedded system security countermeasures.**

Keywords*— *embedded systems, communication security, software, security protocols

I. Introduction

The embedded operating system is a task-oriented computing platform that can be tailored, low-cost, and has high requirements for reliability and real-time performance. The difference from general-purpose computers is that embedded operating systems are generally designed for specific applications and work for specific applications, while communication computers are a general-purpose platform for applications [1].

The embedded operating system is widely used in people's daily life [1]. In the field of electronic communication, embedded system operating systems are applied to important communication node equipment such as mobile phones, base stations, and program-controlled switches. In the consumer electronics industry, embedded operating systems are used in consumer electronics devices such as music players, game consoles, cameras, wearable devices, and handheld POS machines. In household applications, embedded operating systems are used in various household products, such as refrigerators, ovens, and washing machines. In the medical industry, embedded operating systems are used in various advanced medical imaging systems. In public transportation systems, embedded systems are used in central control systems for avionics, automobiles, or hybrid electric vehicles. In urban public infrastructure, embedded operating systems are used in important facilities such as subways, traffic lights, and central control of water conservancy.

It can be seen that although the embedded operating system is not as versatile as the PC operating system, it can be widely used in people's lives because of its tailorable and lightweight advantages. It can be said that the embedded operating system plays an extremely important role in engineering applications. A growing number of physical objects are being connected to the Internet at an unprecedented rate realizing the idea of the Internet of Things (IoT) [2].

With the gradual application of embedded systems in various fields, the shortcomings of its insufficient ability to respond to security threats have gradually emerged, and many hackers have turned their attack targets into embedded systems. In 2009, the "Stuxnet virus" attacked the embedded industrial control system of the Bushehr nuclear power plant in Iran, directly destroying nearly a thousand centrifuges of the Natanz uranium enrichment plant, causing the delay of the nuclear power plant's start-up, which had a major impact on Iran's national nuclear program; At the security conference held in Oakland, California, researchers from the University of California, San Diego and the University of Washington demonstrated a technique for attacking on-board embedded systems [3], which can maliciously tamper with important embedded systems such as on-board adaptive brake controllers and speedometers. Modular control module poses a major security threat to vehicle embedded systems, and may even cause serious traffic accidents; at the Western Conference of the Design Annual Conference held in California in 2012, Vamosizhan, a senior analyst at Mocana, introduced attacks on printers and digital set-top boxes. [4]. These attacks often seriously threaten the security, availability, and reliability of the system, and will bring great security risks to many basic industries that rely on embedded systems to perform important control tasks.

The important reason for these attacks is that embedded systems lack sufficient multi-layer protection mechanisms. This article focuses on the threats embedded systems face in terms of communication security. Then analyze the existing communication security-related technologies from the perspectives of the network layer, the transport layer, and the application layer. Finally, it summarizes the research direction of embedded system security countermeasures.

II. The Vulnerability of Embedded Systems

Different from existing personal computers or servers, the functions, cost, power consumption, and size of embedded devices are completely different from traditional computers.

Moreover, embedded devices often carry a large amount of private information or are used in specific control areas. The data it carries can be easily stolen by attackers. The security problems exposed by embedded systems and their severity are closely related to their own characteristics. A typical embedded system includes a hardware layer, an operating system layer, and an application layer. From a technical point of view, the following reasons make embedded systems vulnerable to attacks:

- Physical exposure, simple hardware structure, lack of safety protection circuit, easy to be attacked by side-channel, etc. Many embedded devices are placed far away from the owner, making it easy for illegal users to physically touch the device, thereby illegally modifying the software and hardware of the device. For example, hackers can illegally spy on the system bus through hardware wiring, probes, etc., and analyze the communication data in the bus. Or directly replace the key components of the hardware system to bypass or destroy the original system function [5][6]. In addition, the non-volatile data stored in the embedded device can be easily accessed illegally. For example, a malicious user can simply solder down the memory chip and read the private data inside through the programmer. Many embedded devices are deployed as data collection products in inaccessible and poor conditions such as the wild. Even if security problems are discovered, it is difficult for maintenance personnel to patch and upgrade their software systems in time.

- The design of the operating system is too simple and can easily be tampered with maliciously. Because of cost and energy consumption considerations, embedded systems are often battery-powered and have low power supply capabilities. Based on the consideration of saving system energy consumption, it is often impossible to add excessive security encryption algorithms. This also leads to the extremely limited computing power of the embedded system CPU, and it is impossible to install anti-virus software and intrusion detection systems used on traditional computers in the embedded system. At the same time, the traditional cryptosystem cannot be used to strictly verify the integrity of its software system. Malicious users can use relevant tools and software to reprogram the firmware in the embedded device and modify it into firmware with malicious code or other illegal purposes. This brings great security risks to embedded devices.

- Various application interactions are complex. Devices are connected through the network and the communication protocol is too simple, which further leads to malicious users being able to launch attacks on the device anywhere on the Internet. It is easy to be implanted with malware such as Trojan horses. What's more serious is that because the embedded system did not consider these threats at the beginning of the design, the design considerations of the network protocol stack were too simple. Many devices use plain text or simply encrypt them and then send them over the network. Externally, it is extremely vulnerable to network intrusion attacks.

III. Software Security Issues of Embedded Systems

Compared with hardware attacks, software attacks are cheaper to implement. Due to its inherent complex characteristics, software systems have a larger attack surface and have become the main attack targets of hackers in recent years. Embedded software systems are faced with many attack threats. According to different attack purposes, these attacks can be subdivided into tampering (with the goal of modifying the code integrity), sabotage (by attacking the running software), and theft (to obtain confidential or private data as the goal).

- Code integrity attack. This kind of attack attempts to modify the relevant data or code of the embedded system. The focus of preventing this kind of attack is to ensure the integrity of the embedded system's own code, and it can detect whether the code has been tampered with by performing security measures on the relevant code of the embedded system before running. Kirovski et al. [7] and Chen et al. [8] used integrity transfer rules when the system was started to ensure that the upper-level program function modules were intact and protected from tampering. These solutions can ensure the static security of the embedded software at the beginning of the startup, but they lack protection for dynamic security. The AEGIS[9] and OASIS[10] architectures extend the instruction set that performs integrity verification in the processor and verifies the integrity of software instruction blocks and important data blocks during software operation. This scheme can protect the dynamic integrity of the embedded system during operation, but these schemes require major changes to the processor and cannot use the existing embedded multi-core processor architecture. There are insufficient feasibility and versatility.

- Application software attack. Attacks against embedded systems running software: such as viruses, Trojan horses, worms, etc., attacks on weak links in the terminal system structure through software agents [11]. This type of attack is a relatively low-cost and relatively common form. In June 2014, security manufacturer F-Secure first discovered the Havex virus [12]. The virus mainly targets the energy industry (hydroelectric dams, nuclear power plants, and power grids). It is used for industrial espionage activities and may disable hydroelectric dams and makes the nuclear power plant is overloaded. In September 2015, a virus called "Ghost Push" infected a large number of Android phones around the world [13], and more than 600,000 mobile phones were poisoned every day. The virus has its own root function, even if the user uses anti-virus software Remove the virus, it will still be installed automatically after restarting the phone. It is very difficult to completely remove the virus. In December 2015, the malware BlackEnergy [14] was implanted in Ukraine's national grid, which caused the power station to shut down unexpectedly, causing large-scale power paralysis and nearly 700,000 people suffering from power outages. In January 2016, Israel's National Electricity Agency network was attacked by ransomware [15]. The attacker sent phishing emails to trick recipients into executing malicious code and encrypting relevant content in the

computer. The power supply system was attacked by a major cyber attack. In August 2016, Apple announced a high-risk "zero-day" vulnerability in iOS [16]. As long as the victim clicks the link sent by the attacker, the phone will be remotely injected with code. The attacker can instantly obtain the highest authority of the victim's mobile phone and can remotely perform any operation on the victim's mobile phone.

- Privacy data theft attack. The purpose of this attack is to obtain sensitive information data stored, transmitted, or manipulated in the embedded system. The main method to prevent this type of attack is to encrypt and protect sensitive information and data, but encryption protection requires keys. The creation, storage, use, and destruction of keys requires the introduction of a trusted key management mechanism to ensure its security. In addition, sensitive information and data can also be protected through access control. Bugiel of the Technical University of Darmstadt, Germany, etc., built a TrustDroid architecture [17] for the access and control of secure data in embedded devices. The architecture uses access restrictions and mandatory access policies at the middleware layer and the kernel layer respectively to ensure the security of embedded critical data. For the security protection of the software layer, the commonly used countermeasure is to learn from the virtualization (VMM) technology [18], which can provide an isolated execution environment without additional hardware overhead, so that security-sensitive software can be transplanted to the VMM environment, and Run protected. However, the security of the embedded virtual machine monitor itself in this scheme has not been effectively guaranteed, so the software security implementation method managed by its docker also needs to be improved.

At the network layer related to software implementation, traditional embedded software did not consider the security issues brought about by network connections at the beginning of its design. Most embedded hardware uses the TCPI/IP connection protocol. Based on the vulnerability analysis described above, embedded software is often connected in plaintext in the network communication design, and the software itself does not encrypt the communication data. At the same time, the identity authentication of the device is achieved by a simple user name and password mechanism. Due to the plaintext transmission characteristics of data, these passwords are easily captured and cracked by attackers. Based on the above reasons, embedded devices are extremely vulnerable to identity spoofing attacks during network access. At the same time, in the internal network composed of embedded devices, nodes are also vulnerable to attacks by spies and Trojan horses, and the data between nodes will also be intercepted, tampered and forged by attackers.

IV. Embedded System Communication Security Technology

From the previous threat analysis, it can be seen that due to the multiple components and complex levels of the embedded system, the exposed attack surface is huge, and security issues are becoming increasingly prominent. The coverage of embedded systems is extremely broad. This section only elaborates and analyzes the existing security protocols from the network layer, transport layer, and application layer.

A. Network layer

The IPSec protocol is generally used to reinforce the security communication at the network layer. The IPsec specification for 6LoWPAN has been proposed in [19]. Some scholars conducted security analysis on the IPSec protocol [20][21]. Considering the multi-hop nature and large message size in 6LoWPAN networks, IPsec provides more effective communication compared with IEEE 802.15.4 security [22]. These papers analyzed the security issues of IPSec and studied its ability to prevent replay attacks. In addition to studying the IPSec protocol specification itself, some scholars have tried to improve the IPSec protocol. Tan X.G, et al. try to apply the KPI system to the IPSec protocol [23]. Li X.F, et al. pointed out the deficiencies of the encryption algorithms (DES, AES, 3DES) needed in IPSec, and proposed the use of IDEA algorithm, SM4 algorithm, and other symmetric encryption algorithms in the IPSec protocol to solve the existence of traditional encryption algorithms Shortcomings [24]. It can be said that scholars have done a lot of work for the IPSec protocol, whether it is theoretical security research or improvement and optimization. Many applications use the IPSec protocol as an implementation of the VPN protocol [21]. In addition, Wang Jian, et al. have combined the Trusted Computing Module (TPM) with the IPsec protocol [25] to extend the IPSec protocol to achieve higher security requirements.

B. Transport layer

The most widely used protocol in the transport layer is the SSL/TLS protocol [26]. The SSL/TLS protocol proposed and designed by Netscape Corporation of the United States is an excellent secure communication protocol and has been successfully applied in computing engineering. Both versions of the SSL protocol 1.0 and 2.0 have serious vulnerabilities and thus have not been widely used. In 1996, the Internet Engineering Task Force (IETF) was responsible for writing SSL3.0 as a specification, which was published through RFC6101 [27]. Later, after IEIF standardization and improvement, the SSL protocol was replaced by the current TLS protocol. The TLS protocol version 1.0 is almost the same as the SSL protocol version 3.0. Later, after a series of updates and improvements, the TLS protocol finally released version 1.3 of the TLS protocol in August 2018 [26]. During the formal release and improvement of the SSL/TLS protocol, many scholars have also done a lot of research and application on it. Among them, in terms of research, Jia Fangshu has discussed the solutions to the situation of session timeout and session disconnection in the application of the client-side of the SSL/TLS protocol [28]. Z. Z. Wang et al. are also improving and optimizing the SSL/TLS protocol itself [29]. Wei Junlin et al. have analyzed that the SSL/TLS protocol may be attacked in the CBC encryption mode, and they have also proposed vulnerabilities in the implementation of SSL/TLS, which may be attacked by three-way handshake attacks, SLOTH attacks, and elastic degradation [30]. In terms of application, scholars have also successfully applied

SSL/TLS in many scenarios. For example, the SSL/TLS protocol is used in the online banking system [31]. Qos perception optimization is based on the SSL/TLS protocol [32], and the SSL/TLS protocol is applied to the information management system [33]. In order to solve the problem of network plaintext transmission, while taking into account the characteristics of embedded hardware power consumption and limited computing power, Thapliyal H, et al. have designed a lightweight encryption scheme specifically for embedded devices [34]. The article proposes that in the production process of the device, a unique ID is generated for each device and stored in the device and the remote database to solve the problem of identity counterfeiting.

C. Application layer

The application layer is the uppermost layer of the network model, and the reinforcement measures taken will vary according to the protocol and application type. The HTTPS protocol is a commonly used security communication reinforcement protocol at the application layer to ensure information security in the communication process [35]. At present, situational awareness technology is mostly done at the application layer for secure communication. Liu Peng, et al. pointed out that situation awareness technology is mainly composed of situation awareness, situation understanding, and situation prediction [36]. Situational awareness mainly includes the collection, storage, analysis, and prediction of network communication traffic, abnormal events, user operation logs, attack information, and other important network communication key characteristic data to monitor the situation in network communication in real-time. Many scholars have researched situational awareness technology. Ma Long, et al. conduct situational awareness based on the network characteristics of traffic [37]. According to the state of the traffic response communication process, deep packet inspection (DPI) technology is used to extract traffic characteristics, and the user's behavior state is analyzed by collecting logs and abnormal information, and a situational awareness evaluation system based on traffic characteristics is also developed. Gao G.Z, et al. has established a data visualization system for the situational awareness system [38], which provides strong basic support for the data processing of the situational awareness system. With the development of big data technology, data mining, and machine learning, Zhu Yijie, et al. have begun to establish a network situation awareness platform under the big data environment [39]. With the technical support of the big data platform, some scholars have begun to use data to perform forecasting functions in the situation. Scholars try to use these data to fit various machine learning models or deep learning models to predict the situation [40][41].

There are also researchers from the perspective of security review and put forward related solutions for security enhancement. Tian G, et al. [42] designed and implemented a Dex file comparison tool, Dexdiff, which compares the compiled Android binary files in a structured manner, and can give contextual differences. In this way, potential malicious applications can be identified through application-level comparison and review. Based on similar design ideas, TaintDroid[43] and PiOS[44] can audit possible privacy leaks in applications through dynamic taint tracking and static data flow analysis.

V. CONCLUSION

After years of development of embedded systems, security issues have become increasingly prominent. The reasons are historical issues and challenges brought by the development of new technologies. With the continuous deepening of trusted computing research, current trusted research in the field of traditional PCs and servers will inevitably extend to embedded systems. Introducing the idea of trusted computing and its mechanism into the embedded system, making the embedded system also a trusted computing environment, will be a hot spot and a breakthrough in embedded security research in the future.

In the field of embedded software development, designers have in-depth cooperation with chip suppliers, using the inherent advantages of hardware to make up for the deficiencies in the software. For example, the HA2lloc hardware-assisted allocator proposed in [45] can use the extended memory management unit to detect memory errors in the heap. [46] proposed HAFIX (Hardware-Assisted Stream Integrity Extension), which constructs new instructions at the hardware layer to prevent code-reuse attacks at the software layer.

In the era of increasingly complex software design, expanding functional requirements, and frequent security attacks, how to ensure the security of all aspects of the software system from the bottom up is a huge challenge for embedded software designers. At the same time, when building a secure and credible embedded system architecture, researchers should combine the current results of artificial intelligence and deep learning to apply them to automatic software behavior measurement, active defense against attacks, and intelligent vulnerability mining. This is also a place worthy of continuous research and deepening in the future.

REFERENCES

[1] Jabeen, Qamar, et al. "A survey: Embedded systems supporting by different operating systems." arXiv preprint arXiv:1610.07899 (2016).

[2] Al-Fuqaha, Ala, et al. "Internet of things: A survey on enabling technologies, protocols, and applications." IEEE communications surveys & tutorials 17.4 (2015): 2347-2376.

[3] MCMILLAN R. How Hackers Attack Cars[EB/OL]. [2010-05-14]. http://www.pcworld.com/article/196320/how_hackers_attack_cars.html.

[4] VAMOSIZHAN R. Embedded System Security[EB/OL]. [2012-10-01]. http://www.altera.com.cn/technology/systemdesign/articles/2012/embedded-sec-urity.html.

[5] KIM L W, VILLASENOR J D. Dynamic function replacement for system-on-chip security in the presence of hardware-based attacks[J]. IEEE Transactions on Reliability, 2014, 63(2): 661-675. DOI:10.1109/tr.2014.2316952

[6] BIDMESHKI M M, REDDY G R, ZHOU L, et al. Hardware-based Attacks to Compromise the Cryptographic Security of an Election System[C/OL]. [2017-03-02]. DOI:10.1109/iccd.2016.7753274.

[7] KIROVSKI D, DRINIĆ M, POTKONJAK M. Enabling Trusted Software Integrity[C/OL]. [2017-06-02]. DOI:10.1145/635508.605409.

[8] CHEN Y, VENKATESAN R, CARY M, et al. Oblivious hashing: A stealthy software integrity verification primitive[C]//Information Hiding. Heidelberg: Springer, 2003: 400-414.

[9] SUH G E, O'DONNELL C W, DEVADAS S. AEGIS: A single-chip secure processor[J]. Information Security Technical Report, 2005, 10(2): 63-73. DOI:10.1016/j.istr.2005.05.002

[10] OWUSU E, GUAJARDO J, MCCUNE J, et al. OASIS: on achieving a sanctuary for integrity and secrecy on untrusted platforms[C]//Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. New York: ACM, 2013: 13-24. DOI: 10.1145/2508859.2516678.

[11] GHOSH A K, SWAMINATHA T M. Software security and privacy risks in mobile e-commerce[J]. Communications of the ACM, 2001, 44(2): 51-57. DOI:10.1145/359205.359227

[12] CHEN P, DESMET L, HUYGENS C. A study on advanced persistent threats[C]//IFIP International Conference on Communications and Multimedia Security. Heidelberg: Springer, 2014: 63-72. DOI:10.1007/978-3-662-44885-4_5.

[13] EDGAR C. Ghost Push Malware Can Root Devices and Install Unwanted Apps[EB/OL]. [2015-10-13]. https://www.androidauthority.com/ghost-push-malware-root-apps-fix-648735/.

[14] KOVACS E. BlackEnergy Malware Used in Ukraine Power Grid Attacks[EB/OL]. http://www.securityweek.com/blackenergy-group-uses-destructive-plugin-ukraine-attacks.

[15] LIU Y T, FAN R, TERZIJA V. Power system restoration: A literature review from 2006 to 2016[J]. Journal of Modern Power Systems and Clean Energy, 2016, 4(3): 332-341. DOI:10.1007/s40565-016-0219-2

[16] CVE.CVE-2016-4657[EB/OL].[2016-08-25]. https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2016-4657.

[17] BUGIEL S, DAVI L, DMITRIENKO A, et al. Practical and lightweight domain isolation on Android[C]//Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices. Chicago: ACM, 2011: 51-62. DOI: 10.1145/2046614.2046624.

[18] BALDIN D, KERSTAN T. Proteus, a hybrid virtualization platform for embedded systems[C]//Analysis, Architectures and Modelling of Embedded Systems. Berlin: Springer, 2009: 185-194.

[19] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt and U. Roedig, "Securing Communication in 6LoWPAN with Compressed IPsec," in Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference On, 2011, pp. 1-8.

[20] Doraswamy, Naganand, and Dan Harkins. IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall Professional, 2003.

[21] Hamed, Hazem, Ehab Al-Shaer, and Will Marrero. "Modeling and verification of IPSec and VPN security policies." 13th IEEE International Conference on Network Protocols (ICNP'05). IEEE, 2005.

[22] S. Raza, S. Duquennoy, J. Höglund, U. Roedig and T. Voigt, "Secure Communication for the Internet of Things—a comparison of link‐layer security and IPsec for 6LoWPAN," Security and Communication Networks, 2012.

[23] Tan Xinglie, et al. "The Application of PKI Technology in IPSec Series Protocols." Computer Science 30.5 (2003): 160-163.

[24] Li X.F, Zhao Y.J, Quan C.B. Application of symmetric key encryption algorithm in IPsec protocol [J]. Journal of Electronic Measurement and Instrument, 2014, 28(1): 75-83

[25] Wang Jian, Wang Haihang, Yang Jian. The remote attestation extension of IPsec protocol [J]. Computer Science, 2011, 38(6): 49-53

[26] Internet Engineering Task Force (IETF). RFC8446. The transport layer security (tls) protocol version 1.3[S]. USA: Internet Engineering Task Force (IETF), 2018.

[27] Internet Engineering Task Force (IETF). RFC6101. The secure sockets layer (ssl) protocol version3.0[S]. USA: Internet Engineering Task Force (IETF), 2011.

[28] Jia Fangshu. Research on Security Technology Issues in the Application of Ssl Protocol [J]. Digital User, 2018, 36: 18-19

[29] Z. Z. Wang, Y. Wang. An improvement ssl protocol application research[J]. International Conference on Electronic Mechanical Engineering and Information Technology, 2011, 12(14): 4010- 4012

[30] Wei Junlin, Duan Haixin, Wan Tao. Overview of Security Defects in https/tls Protocol Design and Implementation [J]. Journal of Information Security, 2018, 3(2): 1

[31] Tang Yi, Wang Zhishuang. Handshake data analysis of personal online banking ssl/tls protocol [J]. Software Guide, 2017, 16(6): 159-162

[32] Q. Fang, T. Zhe. Qos-aware optimization strategy for security ranking in ssl protocol[J]. IEEE, 2009, 9: 842-847

[33] Wu Zhijun, Jiang Yuanchun, Niu Fangchao. SSL protocol interactive authentication scheme for wide area information management system [J]. Information Security Research, 2017, 3(8): 718-726

[34] THAPLIYAL H, VARUN T S S, KUMAR S D. Adiabatic computing based low-power and DPA-resistant lightweight cryptography for IoT devices[C]//VLSI (ISVLSI), 2017 IEEE Computer Society Annual Symposium on. Washington DC: IEEE, 2017: 621-626. DOI:10.1109/isvlsi.2017.115.

[35] Wang Kai, Chen Liyun, Wang Zengguang. Refined fingerprint attack method against https protocol website [J]. Journal of Academy of Armored Forces Engineering, 2018, 32(4): 99-104.

[36] Liu Peng, Chen Houwu, Fang Xiao. Research on Network Security Situation Monitoring Mechanism and Model [J]. Information Network Security, 2018, 9: 66-69.

[37] Ma Long, Sun Jianghui, Du Cheng. Research on Network Situation Awareness System Based on Traffic Analysis [J]. Information Technology, 2016, 9: 97-100.

[38] Gao Feng Zhan, Yong Jiang. Research on Network Security Situational Awareness Visualization System [N]. Global Market Information Herald, September 2018.

[39] Zhu Yijie, Yang Yulong, Li Shuai. Research on Network Security Situation Awareness Platform for Big Data Environment [J]. Network Security Technology and Application, 2018, 11: 52-54

[40] X. F. Liu. Machine learning based ddos attack detection from source side in cloud[C]. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing, American, 2017, 114-120.

[41] Jiang Yibo, Wang Yuchen, Wang Wanliang. Research on Manet Network Intrusion Detection Performance Evaluation Method Based on Machine Learning [J].Computer Science, 2013, 40: 170-174.

[42] MITCHELL M, TIAN G, WANG Z. Systematic audit of third-party android phones[C]//Proceedings of the 4th ACM Conference on Data and Application Security and Privacy. New York: ACM, 2014: 175-186. DOI:10.1145/2557547.2557557.

[43] ENCK W, GILBERT P, HAN S, et al. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones[J]. ACM Transactions on Computer Systems (TOCS), 2014, 32(2): 5. DOI:10.1145/2494522

[44] EGELE M, KRUEGEL C, KIRDA E, et al. PiOS: Detecting privacy leaks in iOS applications[J]. Network & Distributed System Security Symposium, 2011, 1: 280-291.

[45] ARIAS O, SULLIVAN D, JIN Y. HA2lloc: Hardware-assisted secure allocator[C]//Proceedings of the Hardware and Architectural Support for Security and Privacy. New York: ACM, 2017: 8. DOI:10.1145/3092627.3092635.

[46] DAVI L, HANREICH M, PAUL D, et al. HAFIX: Hardware-assisted flow integrity extension[C]//Proceedings of the 52nd Annual Design Automation Conference. New York: ACM, 2015: 74. DOI:10.1145/2744769.2744847.