Software-Defined Data Flow Detection and Control Approach for Industrial Modbus/TCP Communication

Ming Wan[1,3(✉)], Yan Song[2,3], Yuan Jing[1], Zhaowei Wang[3], Jianming Zhao[3], and Zhongshui Zhang[4]

[1] School of Information, Liaoning University, Shenyang 110036, China
wanming@lnu.edu.cn
[2] School of Physics, Liaoning University, Shenyang 110036, China
[3] Shenyang Institute of Automation Chinese Academy of Sciences, Shenyang 110016, China
[4] CNGC North Automatic Control Technology Institute, Taiyuan 030006, China

Abstract. There is an increasing consensus that software-defined networking may become a successful case to provide fine scalability and availability for industrial Internet, and it also brings new opportunities for the development of industrial cyber security. Aligning with the defense in depth strategy, this paper proposes a software-defined data flow detection and control approach for industrial Modbus/TCP communication. Furthermore, this approach designs a novel security strategy configuration service in SDN controllers to publish the flow control rules, and SDN switches match Modbus/TCP data flows with these flow control rules to detect and control abnormal communication behaviors. Specifically, a flow control rule database which stores all flow control rules of the entire control system is managed by SDN controllers, and a security flow table is maintained by each SDN switch according to different requirements of industrial communication. By using the DPI (Deep Packet Inspection) technology, this approach can run a deep analysis of Modbus/TCP packets according to the protocol specification, and block the improper control commands or undesired technology parameters. The qualitative analysis shows that the proposed approach possesses certain advantages and feasibilities.

Keywords: Modbus/TCP · SDN · Flow detection and control
Cyber security

1 Introduction

SDN (Software-Defined Networking) has been widely studied and discussed by both academia and industry, and its field is growing at a very fast pace [1]. In practice, SDN changes the limitations of current network infrastructures, and presents a new routing architecture of logic control and data forwarding separation [2]. Furthermore, the entire network architecture is separated into the control plane and data plane, and the critical OpenFlow technology is used to control and manage the network routing and forwarding [3]. In the data plane, SDN switches maintain fine-grained flow tables to

© Springer Nature Switzerland AG 2019
F. Xhafa et al. (Eds.): IISA 2018, AISC 885, pp. 722–729, 2019.
https://doi.org/10.1007/978-3-030-02804-6_94

forward the data traffic, and guarantee the end-to-end transmission. In the control plane, SDN controllers offer centralized network management, and simplify the policy enforcement and network configuration, such as flow table generation and configuration. By decoupling the control plane and data plane, SDN presents significant benefits, and has an excellent ability to apply in a wide variety of networked environments, including enterprise networks, data centers, infrastructure-based wireless access networks, et al. [1].

In recent years, with the deep integration of information technology and operational technology, ICT (Information Communication Technology) has been emphasized and developed in various network architectures, such as Industrial Internet [4], LTE network [5] and Internet of Things [6]. Due to the significant advantages of SDN, many researchers have developed some new industrial network architectures which are based on software-defined networking [3, 7–9]. In these researches, SDN not only can meet the real-time transmission requirements of industrial applications, but also can overcome the incompatibility problem causing by different network functions. However, although the application of SDN in industrial environments can bring many advantages, it cannot solve the industrial-oriented cyber security problems. Actually, industrial control systems are facing the escalating cyberattacks, and have caused great loss. Statistically, ICS-CERT reported that the number of industrial security incidents had reached 290 in 2016 [10]. Additionally, because these cyberattacks always skillfully steal industrial-oriented properties, the traditional IT security technologies cannot play an active role of industrial security protection [11]. Consequently, both academia and industry start to exploit the intrinsic system weaknesses [12] and develop novel industrial security solutions, including access control [13], vulnerability evaluation [14], intrusion detection [11, 15], et al.

Deserved to be mentioned, when a new network architecture or model comes to being, it may also bring new opportunities for the development of security services. In consequence, SDN-based security mechanisms in industrial control systems have started to be explored [16, 17]. In this paper, based on the defense in depth strategy emphasized by NIST (National Institute of Standards and Technology) [18], we propose a software-defined data flow detection and control approach for industrial Modbus/TCP communication, which is an improved approach based on our prior work [19]. Furthermore, this approach designs a novel security strategy configuration service in SDN controllers to publish the flow control rules, and SDN switches match Modbus/TCP data flows with these flow control rules to detect and control abnormal communication behaviors. More specifically, SDN controllers also manage a flow control rule database storing all flow control rules of the entire control system, and each SDN switch maintains a security flow table according to different requirements of industrial communication. Besides, in order to block the improper control commands or undesired technology parameters, our approach also uses the popular DPI (Deep Packet Inspection) technology to run a deep analysis of packets on the basis of Modbus/TCP protocol specification. Finally, we give the qualitative analysis to illustrate that the proposed approach possesses certain advantages and feasibilities.

2 Software-Defined Data Flow Detection and Control Approach

2.1 Basic Model and Architecture

As shown in Fig. 1, the basic model and architecture of our approach is also composed of two parts: control plane and data plane

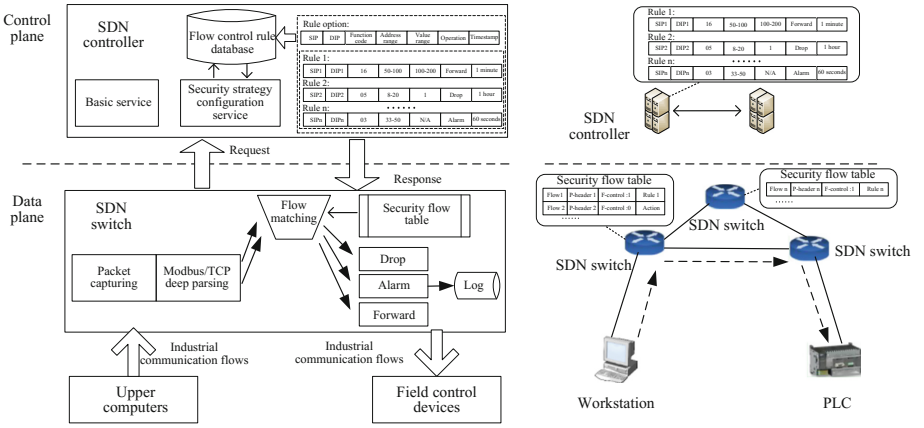

Fig. 1. Basic model and architecture of software-defined data flow detection and control approach

In the control plane, one or more distributed SDN controllers generate routing and forwarding strategies of all SDN switches according to the specific network status and user configuration. Moreover, SDN controllers possess the intrinsic basic service and the novel security strategy configuration service, and manage a flow control rule database which stores all user-defined flow control rules. As described by OpenFlow, the basic service provides multiple network management functions, including network topology management, device registration, routing computation, etc. By establishing the whole network view, SDN controllers can compute the routing path of each data flow, and decide the corresponding flow table for each SDN switch. Differently, the security strategy configuration service completes the security function which publishes the flow control rules for all industrial data flows, and sends these flow control rules to the need-related SDN switches.

In the data plane, all SDN switches constitute the entire transmission network, and each SDN switch holds one or more security flow tables. Furthermore, the structure of this table mainly including four parts: Flow ID, Packet header information, security control identification and flow control rule. Here, Packet header information covers source IP address, destination IP address, IP protocol, source port, destination port and other optional tuples defined in OpenFlow flow tables. The detailed terms can be defined in Table 1. By using the DPI technology, SDN switches run a deep analysis of all packets belonging to each Modbus/TCP data flow, and match the key contents with

the flow control rules to detect and block the improper control commands or undesired technology parameters in industrial Modbus/TCP communications.

Table 1. Term definition in flow control rules and security flow tables

Terms	Definition and description
SIP	Source IP address in one Modbus/TCP data flow or packet
DIP	Destination IP address in one Modbus/TCP data flow or packet
Function code	Detailed description in Reference [13]
Address range	Detailed description in Reference [13]
Value range	Detailed description in Reference [13]
Operation	Processing modes in flow control rules Forward: pass the corresponding Modbus/TCP flows or packets; Drop: drop the corresponding Modbus/TCP flows or packets; Alarm: generate an alarm and logging;
Timestamp	The lifetime of one flow control rule in SDN switches
Flow n	Representing ID of one data flow
P-header	Modbus/TCP packet header information
F-control	Security control identification 1: matching Modbus/TCP data flows with the following flow control rule; 0: processing Modbus/TCP data flows according the following actions
Action	Optional actions defined in OpenFlow, such as queuing or modifying

2.2 Detailed Executing Process

Figure 2 depicts the detailed executing process to detect and control Modbus/TCP data flow by using our approach. The main contents can be described as follows:

Step 1: when one workstation wants to send some control commands to one PLC, it first need establish the initial TCP connection with the PLC, and constructs the Modbus/TCP connection request packet *P1* which will be sent to the PLC.

Step 2: when the SDN switch receives this request packet *P1*, it analyzes this packet and get the key information, including source IP address, destination IP address, destination port, et al. After that, the SDN switch looks up its security flow table, and finds whether the corresponding flow ID exists. If it exists, the SDN switch will further process this packet according to the operation of this flow ID; if it does not exist, the SDN switch will construct one flow control rule request packet *R1* for this Modbus/TCP data flow, and send it to the SDN controller.

Step 3: after the SDN controller receives the request packet *R1*, it finds the corresponding flow control rule in the flow control rule database, and sends this rule to the SDN switch by using the response packet *R2*. When the SDN switch receives this packet, it generates new flow ID and stores this rule in its security flow table, and forwards the Modbus/TCP connection request packet *P1* to the PLC.

Step 4: after receiving the packet *P1*, the PLC sends the Modbus/TCP connection response packet *P2* which is forwarded by the SDN switch, and establishes the TCP connection with the workstation.

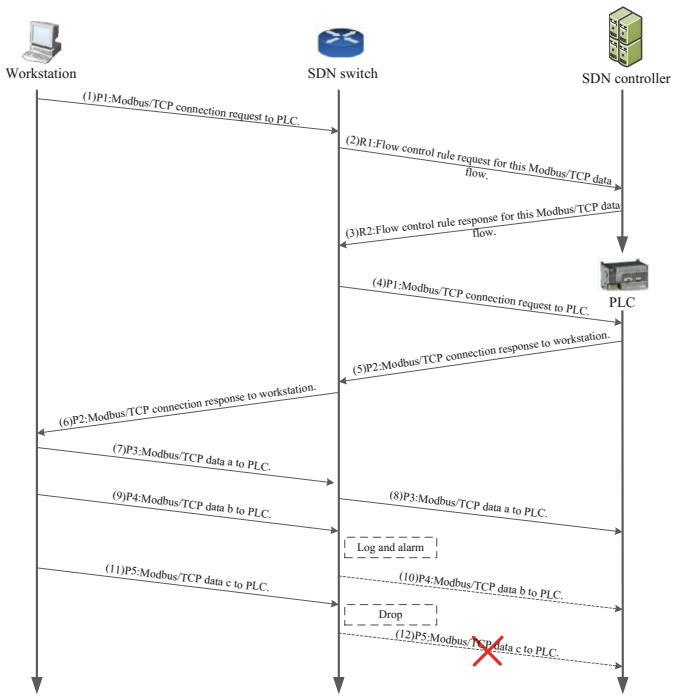

Fig. 2. Detailed executing process to detect and control Modbus/TCP data flow

Step 5: when the workstation sends the new Modbus/TCP data *a* to the PLC by using the packet *P3*, the SDN switch will match the parsed key contents with the rule belonging to this flow ID. If *F-control* is 1 and the corresponding operation is *Forward*, the SDN switch will forward this packet *P3* to the PLC.

Step 6: when the workstation sends the new Modbus/TCP data *b* to the PLC by using the packet *P4*, the SDN switch will match the parsed key contents with the rule belonging to this flow ID. If *F-control* is 1 and the corresponding operation is *Alarm*, the SDN switch will log these key contents and generate an alarm. However, it will also forward this packet *P4* to the PLC.

Step 7: when the workstation sends the new Modbus/TCP data *c* to the PLC by using the packet *P4*, the SDN switch will match the parsed key contents with the rule belonging to this flow ID. If *F-control* is 1 and the corresponding operation is *Drop*, the SDN switch will drop this packet *P4*.

2.3 Security Flow Table Generating and Maintaining

In our approach, security flow table is an essential point to offer security network services, because it not only possesses the basic forwarding function defined by OpenFlow, but also implements an effective strategy of deep defense. In order to generate and maintain security flow tables in SDN switches, we suggest the general steps as follows:

Step 1: when the SDN controller receives the request packet from the SDN switch, it first parses this packet according to the basic service function. According to the parsed packet header, network topology, link state or other information, the SDN controller explores the initial flow items, including Flow ID, P-header, Action, et al. The detailed description of this step can refer to the OpenFlow protocol.

Step 2: the security strategy configuration service function in the SDN controller looks up its flow control rule database according to source and destination IP addresses. If some flow control rule matches with these information, go to Step 3; if no flow control rule matches with these information, the SDN controller sets *F-control* to 0 and send the initial flow items to the SDN switch.

Step 3: the SDN controller sets *F-control* to 1, and sends the initial flow items and the matched flow control rules to the SDN switch

Step 4: after the SDN switch receives these information, it generates a new security flow entry and stores it in its table. Additionally, the SDN switch maintains its security flow table by means of the timestamp in each flow control rule. Specifically, the timestamp represents the lifetime of one flow control rule in the SDN switch, and if the timer of one flow control rule changes from the timestamp to 0, this rule will be deleted by the SDN switch.

3 Qualitative Analysis

Compared with current industrial control network, the centralized and manageable SDN architecture can provide promising solutions to the problems of traditional industrial networks [3]. Based on SDN technology, our approach puts forward an additional security mechanism to detect and control abnormal Modbus/TCP communication behaviors, and further improves the security of SDN-based control systems. The qualitative analysis on the advantages and feasibilities of our approach is listed as follows:

1. Our approach meets the demands of defense in depth strategy, and can divide industrial control systems into different security enclaves by setting different flow control rules. Additionally, our approach supports the DPI technology according to Modbus/TCP protocol specification, and can dynamically adapt to industrial-oriented properties.
2. The flow control rules are based on the centralized management of all SDN controllers, and this situation can facilitate the reconfiguration of industrial network according to various industrial applications.
3. Our approach does not affect the scalability, because our approach can be successfully implemented without changing the basic characteristics of SDN.
4. Based on our prior work [13], the main security defense technologies in our approach are feasible. Additionally, the fine-grained security flow tables are fast and accurate for Modbus/TCP data flows, and the timestamp can avoid wasting resources which is caused by enormous number of entries in the security flow table.
5. Although the end-to-end transmission delay may be increased to some extent, we believe our approach still meets the real-time transmission requirements of

industrial networks. The causes are chiefly as follows: on the one hand, SDN switches can provide adequate levels of performance to perform deep packet parsing and matching; on the other hand, our prior work [13] has already proven the real-time capability even though the security defense technologies are implemented in the form of network middleware.

4 Conclusion

The SDN architecture in industrial control systems can bring new opportunities for the development of security services. From this point, this paper proposes a software-defined data flow detection and control approach for industrial Modbus/TCP communication. Furthermore, approach designs a novel security strategy configuration service in SDN controllers to publish the flow control rules, and SDN switches match Modbus/TCP data flows with these flow control rules to detect and control abnormal communication behaviors. Additionally, our approach uses the popular DPI technology to run a deep analysis of data flows according to Modbus/TCP protocol specification, and can support the defense in depth strategy in industrial control systems. Finally, with the help of the qualitative analysis, we show that the advantage and developing prospect of our approach is foreseen. In the future work, we will realize our approach and build the experimental platform, and quantitatively evaluate its performance and defense effect.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No. 61501447), and the General Project of Scientific Research of Liaoning Provincial Department of Education (LYB201616). The authors are grateful to the anonymous referees for their insightful comments and suggestions.

References

1. Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tutor. **16**(3), 1617–1634 (2014)
2. Kreutz, D., Ramos, F.M.V., Verissimo, P., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive Survey. Proc. IEEE **103**(1), 14–76 (2015)
3. Li, D., Zhou, M.T., Zeng, P., Yang, M., Zhang, Y., Yu, H.: Green and reliable software-defined industrial networks. IEEE Commun. Mag. **54**(10), 30–37 (2016)
4. Posada, J., Toro, C., Barandiaran, I., Oyarzun, D., Stricker, D., Amicis, R., Pinto, E.B., Eisert, P., Dollner, J., Vallarino, I.: Visual computing as a key enabling technology for Industrie 4.0 and Industrial Internet. IEEE Comput. Graph. Appl. **35**(2), 26–40 (2015)
5. Zhang, J., Deng, L., Li, X., Zhou, Y., Liang, Y., Liu, Y.: Novel device-to-device discovery scheme based on random backoff in LTE-advanced networks. IEEE Trans. Vech. Technol. **66**(12), 11404–11408 (2017)
6. Li, S., Zhang, N., Lin, S., Kong, L., Katangur, A., Khan, M.K., Ni, M.: Joint admission control and resource allocation in edge computing for internet of things. IEEE Network **32** (1), 72–79 (2018)

7. Hu, P.: A system architecture for software-defined industrial internet of things. In: Proceedings of 2015 IEEE International Conference on Ubiquitous Wireless Broadband, Montreal, Canada, October 2015, pp. 1–5 (2015)
8. Genge, B., Haller, P.: A hierarchical control plane for software-defined networks-based industrial control systems. In: Proceedings of 2016 IFIP Networking Conference and Workshops, Vienna, Austria, May 2016, pp. 73–81 (2016)
9. Gupta, A., MacDavid, R., Birkner, R.: An industrial-scale software defined internet exchange point. In: Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, CA, USA, March 2016, pp. 1–14 (2016)
10. NCCIC/ICS-CERT, NCCIC/ICS-CERT year in review (2016) https://ics-cert.us-cert.gov/Year-Review-2016 (2017)
11. Wan, M., Shang, W.L., Zeng, P.: Double behavior characteristics for one-class classification anomaly detection in networked control systems. IEEE Trans. Inf. Forensics Secur. **12**(12), 3011–3023 (2017)
12. Ly, K., Jin, Y.: Security challenges in CPS and IoT: from end-node to the system. In: Proceedings of 2016 IEEE Computer Society Annual Symposiumon VLSI, Pittsburgh, USA, Jul. 2016, pp. 63–68 (2016)
13. Wan, M., Shang, W.L., Kong, L.H., Zeng, P.: Content-based deep communication control for networked control system. Telecommun. Syst. **65**(1), 155–168 (2017)
14. Kim, S., Jo, W., Shon, T.: A novel vulnerability analysis approach to generate fuzzing test case in industrial control systems. In: Proceedings of 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, May 2016, pp. 566–570 (2016)
15. Huo, Y., Hu, C., Qi, X., Jing, T.: LoDPD: a location difference-based proximity detection protocol for fog computing. IEEE Internet Things J. **4**(5), 1117–1124 (2017)
16. Ndonda, G.K., Sadre, R.: A low-delay SDN-based countermeasure to eavesdropping attacks in industrial control systems. In: Proceedings of 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, Berlin, Germany, November 2017, pp. 1–7 (2017)
17. Genge, B., Graur, F., Haller, P.: Experimental assessment of network design approaches for protecting industrial control systems. Int. J. Crit. Infrastruct. Prot. **11**, 24–38 (2015)
18. Stouffer, K., Falco, J., Scarfone, K.: Guide to industrial control systems (ICS) security. National Institute of Standards and Technology (NIST). http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82.pdf (2013)
19. Zeng, P., Shang, W., Li, D., Wan, M., Zhao, J., Liu, J., Yang, M.: Method for controlling transmission security of industrial communications flow based on SDN architecture, USA, US20170339109A1, 23 November 2017