International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

A Survey on Improving Security in Internet of Things (IoT) with SDN

Janani R, Siddique Ibrahim S. P,Kirubakaran R

Department of Computer Science and Engineering, Kumara guru College of Technology, Coimbatore, Tamilnadu, India

ABSTRACT

Internet of Things (IoT) connect to the Internet billions of heterogeneous smart devices with the ability of interacting with the environment. Software-defined Networking (SDN) use to automatically and dynamically managing network flows. SDN switches are basically powerful machines, that can be used as fog nodes accordingly. Therefore, SDN seems a good choice for IoT-Fog networks. IoT Environment that connected to numerous devices which are generating a large amount of data, because of that processing and transporting information becomes main challenge. Collecting information to help people could lead serious information leakage, and if IoT is combined with critical control system (e.g., nuclear system, Electric control board), security attack would cause loss of lives. Software-Defined Networking (SDN) framework for introducing security in IoT gateways.

Keywords: Internet of Things, Fog/edge computing, SDN, Security

I. INTRODUCTION

Internet of Things (IoT) is becoming a part of our daily lives ,that will ease our daily life. Managing such a huge amount of connections is a big challenge for network administrators[2]. Fog/edge computing is an architecture organized by the networking edge devices or clients to provide computing services for customers or applications in the space between networking central servers and end-users. fog/edge computing is organized as distributed architecture and can process data and store data in networking edge devices, which is close to end-users, fog/edge computing can provide services with faster response and greater quality, in comparison with cloud computing[4]. Software Defined Networking (SDN) is expected to be a key enabler for 5G (5th generation of wireless systems), which will need to integrate both IoT services with traditional human based services together. Through global orchestration of virtual network, memory and equipment for computing are provided and immediately delivering for the analysis of data. The technical solution proposed in this paper provides an agile answer on how to overcome these problems and improve the IoT security within software-defined networks.[1]. In SDN, the controller controls all the switches through "OpenFlow" channels. Commands and requests from the controller and status and result from the switches, are transmitted through the OpenFlow channels. That are intercepted, may bring disastrous circumstances to both the network providers and their customers. Encrypt the channel for security using cipher techniques after authentication. However, authentication and encryption alone cannot guarantee the safety of the OpenFlow channels. TLS, for example, is one of the most popular cryptographic protocols. However, there are still works exploiting vulnerabilities in its cipher suites and the protocol itself [2]. The attacker can compromise a TLS link by cautious installing a client certificate. Moreover, since smart embedded devices in IoT have limited resources, some safe but computing intensive protocols cannot be deployed on them. Secure communication is more important without these, devices are more vulnerable to be compromised, increasing the risks of attacks against OpenFlow channel. Even assuming it were perfectly safe,but implementing TLS is very troublesome. It indicates that most SSL implementations are partially implemented, and contain potential vulnerabilities.If the attacker were to find the credentials or passwords of the switches or controllers, there are limited approaches to detect and defend against the attacks. In general, we cannot only depend on cipher techniques. There should be other complimentary systems to secure Open Flow channels. To detect such attacks, it may be possible to use a packet monitor to investigate those packets in the

Open Flow channels. However, the attacker does not necessarily change all the packets passing through the channels. With only one or two packets inserted or dropped, the attacker can easily change a switch's behavior. Therefore, monitoring the channel is not efficient. Besides, developing another monitoring system could cost much time and money. IoT considered as multi-layer architectures, divided into the perception layer, networking layer, service layer, and application layer. Based on the multi-layer architecture, enabling technologies and open issues in each layer are then presented. After that, security vulnerabilities and challenges are discussed, and the security issues with respect to confidentiality, integrity, availability, as well as privacy issues in IoT are discussed. In addition, the integration of IoT and fog/edge computing and related issues are presented to enable the design and deployment of fog/edge computing based IoT. Finally, several applications (smart grid, smart transportation, and smart cities) are presented to illustrate how fog/edge computing-based IoT are to be implemented in real world

IoT-based systems. In summary ,the proposed work considerations are:

- To build demonstrations of these attacks to show how the attackers modify flow paths, collect sensitive information, and poison the controller's global view. Implementations are relatively simple scripts with a few lines.
- Based on SDN features, this paper propose a lightweight countermeasure to detect MitM attacks against Open Flow channel.
- The proposed w implement a prototype system to detect packet modification with Bloom filters based on SDN and extending the Open Flow protocol.

II. STATE OF THE ART

A. Internet of Things

This section analyzes security requirements based on 3 typical IoT characteristics that have been researched in other researches. These security requirements are commonly applied in IoT security. Therefore, it is important to understand and take advantage of it to design security mechanisms in IoT environment.

a. Heterogeneity

In IoT, heterogeneity means diversity of hardware performances (e.g. CPU computation, memory footprint), protocols, platforms, policies, etc. The biggest problem of heterogeneity is absence of common security service heterogeneity weakens interoperability and causes extra fees about performance and money to interpret each other [10]. Besides, making security-related policies and updates are more complex. In order to solve these problems, we can use some technologies (e.g., meta data registry (MDR), middleware); however, it is not a fundamental solution. For providing common security service, unified IoT security standard has to be established. Then, developer who are related to IoT development should follow standards strictly. Recently, standards organizations to develop some standards for the security in the IoT.

B. Resource Constraint

The constraints IoT devices, such as personal device ,sensors are lack of performance and battery capacity. The legacy security services are TLS for Transport layer security, using AES (advanced encryption standard) which cannot be applied to IoT devices directly . So, these services or algorithms should be designed to be lightweight and straightforward to increase efficiency of CPU processor, memory and battery of IoT devices. In addition, scalability has to be considered, where it should handle devices if it increase also. For lack of performance of IoT device and network bandwidth are low, so that multicast is more effective than unicast .Note that, CoAP (constrained application protocol) supports multicast in RFC 7252 .

C. Dynamic Environment

Because of mobility and bad network connections, IoT has a dynamic network topology, which should rely on network connections. Now a trending case (e.g., smart city), numerous devices may send a large number of requests. Hence, scalability and flexibility is more important requirement in IoT communication protocols. Cisco estimates that 50 billion devices will be created by 2020, and after that, more and more devices will be made. Consequently, flexibility and scalability will be key security requirements of IoT.

D. Software-Defined Networking:

Software-Defined Networking (SDN) is an network architecture that is dynamic, and adaptable, making it ideal for the high-bandwidth. This architecture underlying infrastructure which seperates the network control and forwarding functions that will enable the network control to become directly programmable which is abstracted for applications and network services.SDN was commonly associated with the Open Flow protocol (for remote communication with network plane elements for the purpose of determining the path of network packets across network switches) since the latter's emergence in 2011. Since 2012, however, many companies have moved away from Open Flow, and have embraced different techniques. It is assumption that the SDN paradigm to solve the resource management needs for network environments for the following reasons, which are,

SDN allows for a clear separation between services in the control and the data. The decoupling of the control plane from the forwarding plane encourages abstractions of low level network functionalities. Virtually centralized view of the network, it allows to perform network optimization techniques. Replication mitigation techniques may help to avoid single points of failure.

Programmability in network provides the dynamic and fast creation of new network services. The Open Flow protocol is an open source protocol that is a fundamental element for building SDN solutions. Open Flow protocol allows network controllers to determine the flow paths across a network of switches thus enabling the easy traffic management through the separation of the control plane from the forwarding plane. Tables in Open Flow Switch that perform packet lookups and forwarding, and an Open Flow channel to an external controller. In SDN, the controller controls all the switches through "Open Flow" channels. Commands and requests from the controller ,as well as status and statistics from the switches, are transmitted through the Open Flow channels. Therefore, the security and reliability of Open Flow channels between the controller and switches are critical for proper SDN operation, configuration, and management. The SDN controller manages the switch via the Open Flow protocol. Using this protocol, the controller can add, update, and delete flow entries, both reactively (in response to packets) and proactively [2] The SDN controller is a software entity that has exclusive control over an abstract set of data plane resources. Several open source implementations of an SDN controller are available. Applications can run on top of SDN controller in order to offer advanced network services.

E. SDN-Enabled Security Framework

In this paper ,proposed architecture consists of three parts: SDN/NFV edge node, SDN Controller, End to End Security connection.

The SDN/NFV Edge Node, which is handling services network node itself. This will improve efficiency and compress the data that needs to be transported to the cloud process the data, analysis and memory. This is often done for improving the efficiency of the network, but it may also be implemented for security and compliance reasons. This distributed network node has the ability to act as:
a) An Open Flow-enabled switch in order to switch the different IoT gateways that are connected. Moreover connectivity to aggregation and transport networks is also provided.

b) Virtual Machines running different services, such as an IoT database, where the measurements of the different sensors are stored for local processing.

Second, The SDN controller supports Open Flow control of flow tables, meters and actions. A clear North Bound Interface (NBI) of the SDN controller used to the integration of higher level applications.

Third, The End-to-end security Application will do monitoring of the current flows, and through different anomaly detection mechanism it will be able to identify malicious flows. Finally, this application provide security policies used to the detected anomalies in order to reduce its severity of them.

III. ATTACK IN OPENFLOW CHANNEL

Assuming switch and controller are secured, they work well in open flow channel. Every IoT Local Area Network has a gateway switch and a fog node. This gateway and the fog node are usually combined together for efficiency. ISP controller controls gateway. Since ISP Cloud is more secure that it is safer to put the

controller in ISP cloud rather than IoT LAN. ISP offers its customers virtual machines with controller software installed, giving them rights to control their gateway switches and fog nodes. Usually, the gateway switch and controller in ISP cloud communicates in TLS. The goal of the attacker is to intercept this encrypted communication channel. As introduced in [5], the attacker can launch KCI attack to intercept the communication channel between a client and a server by stealthily installing a client certificate at the client side. In order to successfully install client certificate at the gateway switch, the attacker needs a helper inside the LAN. There are a large amount of embedded smart devices are vulnerable to firmware updating attack, in which the attacker compromise a smart device's firmware through legitimate updating processes. If there is such a device inside the IoT LAN, the outside attacker can take control of it by launching firmware modification attack. Then the smart device, ordered by the attacker, installs a client certificate at the gateway, claiming that the fog node needs to use this certificate to identify itself in their future communications. After the gateway installs the client certificate, the outside attacker breaks the connection between the controller and the gateway and performs KCI attack [5] to achieve MitM attack on the OpenFlow control channel .After these steps, the attacker has successfully intercepted the OpenFlow channel and take control of the gateway.

IV. EVALUATION

In this paper evaluating the performance of the technique Bloom filter method and it introduces delay. Then, test the accuracy (false positive rate) of this method. Obtaining the additional technique use the number of switches and hosts. Mininet is used to simulate more switches and hosts to generate more traffic flows. The Floodlight controller connects with all the switches remotely. An attacker stealthily injects commands in the OpenFlow channels. There is no flow path in the data plane that consists solely of compromised switches. A conformant flow is created with the following properties: a) Using the base of 20 packets/second, which shall match for example the readings of 20 sensors each second, we introduce a small uniform variance; and b) The number of bytes per second is linearly obtained from the packets per second (using a standard 100 bytes per packet). Instead, a bad flow is created with a probability of 10%. A bad flow has different properties: a) The number of packets per second are duplicated (in comparison with a conformant flow); and b) The packet size is also 50% increased.

V. CONCLUSION

In this paper, focusing on the potential threat of Man in the Middle attacks considering on Open Flow channels in IoT-Fog basis. This paper, introduce an attack model to show how to perform such attack on proposed SDN architecture. To detect such attacks, here propose a countermeasure using bloom filter to detect MitM attack. A prototype of this Bloom Filter Monitor is implemented by extending the Open Flow protocol. The evaluation result shows that the Bloom filter method is both lightweight and efficient. A simple algorithm has been implemented to analyze the feasibility of statistical analysis for anomaly detection and introduced the proposed SDN security application for IoT by interconnecting the ADRENALINE and IOTWORLD test beds, running on top of an SDN/NFV-enabled edge node. Finally, the flow interruption anomaly mitigation technique. Further research on security for IoT needs to be performed, but using the powerful framework on SDN, has been demonstrated as a useful weapon against security threads.

VI. REFERENCES

[1]. Cheng Li, Zhengrui Qin, Ed Novak, Qun Li, "Securing SDN Infrastructure of IoT-Fog Networks from MitM Attacks"

[2]. Ricard Vilalta1, Raluca Ciungu2, Arturo Mayoral1, Ramon Casellas1, Ricardo Martínez1, David Pubill1, Jordi Serra1, Raul Muñoz1, and Christos Verikoukis1 "Improving Security in Internet of Things with Software Defined Networking"

[3]. Giotis, K., et al. "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments." Pp 122-136, Computer Networks 62 (2014)

[4]. S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," IEEE Communications Surveys & Tutorials, 2015.

[5]. Se-Ra Oh, Young-Gab Kim, "Security Requirements Analysis for the IoT"

[6]. S. Scott-Hayward, S. Natarajan, and S. Sezer, “A survey of security in software defined networks,” IEEE Communications Surveys & Tutorials,2015.

[7]. Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications”.

[8]. J.-Y. Lee, W.-C. Lin, Y.-H. Huang, “A lightweight authentication protocol for internet of things”, International Symposium on Next-Generation Electronics Taiwan, pp. 1-2, May 2014

[9]. P. Porras, S. Cheung, M. Fong, K. Skinner, and V. Yegneswaran, “Securing the software-defined network control layer,” in NDSS, 2015.