International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 2, February 2020 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Novel SDN Architecture for IoT Security

M. Silambarasan, B. Michael Vinoline Rinoj, V. Karthik

Assistant Professor, Sethu Institute of Technology, Virudhunagar, Tamil Nadu, India

ABSTRACT
Describes the term Internet of Things (IoT) security architecture based on Software Defined Networking (SDN). In this context, building on SDN works with or without infrastructure. This is called the SDN domain. This work describes the mechanics of the proposed architecture and reduces the chances of using SDN to achieve more effective and flexible network security. It outlined the issues associated with current SDN security applications and introduced a new IoT system plan. This document has discussed the management of Internet access for specific networks and monitoring of global traffic. Finally, it describes the choice of architecture for SDN using OpenFlow and discusses the resulting results.

KEYWORDS: *SDN, ad-hoc networks, interconnect framework*

How to cite this paper: M. Silambarasan | B. Michael Vinoline Rinoj | V. Karthik "A Novel SDN Architecture for IoT Security" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-2, February 2020, pp.48-52, URL: www.ijtsrd.com/papers/ijtsrd29908.pdf

IJTSRD29908

1. INTRODUCTION

The Internet has grown rapidly over the past few decades and is still evolving in size and sophistication. By the end of 2014, 42.3% of the world's population was connected to the network [1]. However, Internet security threats increase with the development of the Internet. Internet of Things (IoT) has a security concern because it includes all objects or devices with network capabilities. Things include things that may be dangerous to human life, such as simple household sensors, medical equipment, cars, planes, and even nuclear reactors. The number of violations in 2013 was 62% higher than in 2012 [2].

Traditional security mechanisms, such as firewalls and intrusion detection and prevention systems, are deployed on the edge of the Internet. These mechanisms are used to protect your network from external attacks. These mechanisms are not enough to protect the next generation of the Internet. The limitless Internet of Things architecture raises additional concerns about controlling network access and checking software. In [4], the author provides details of a network access control application based on a named Internet of Things (PANATIKI) device.

Recent developments in computer networks have introduced a new technology model for future communications, software-defined networks (SDN). A central software program called the SDN Console manages the entire network. SDN separates control aircraft and data and focuses network intelligence logically. The console can add, update and delete flow entries in response to packages and use pre-defined rules. In addition, the SDN enables rapid response to security threats, precise traffic filtering and dynamic security policy deployment.

We suggest IoT security model based on the SDN architecture. First, the proposed security model is designed to create and secure a wired and wireless network infrastructure. Second, we have expanded the proposed structure to include dedicated networks and network objects such as sensors, tablets, and smartphones. The main contributions are:

As far as we know, this is the first tool to use the SDN architecture to address Internet of Things security issues.

- ➢ Design secure IoT based SDN designs, inspired by existing network access control and security technologies.
- ➢ Promote the exchange of security policy and deployment among SDN control areas based on the safety net model [23].

Security models are discussed later in this article. It concludes with an overview of an SDN-based security vision based on IoT solutions.

2. Software-defined network architecture

SDN has emerged as a strategy to improve network capabilities, reduce costs, reduce hardware complexity, and enable innovative research. The SDN architecture model has three layers [3], [5], [6]. The infrastructure layer consists of network devices (for example, adapters, routers, virtual adapters, and wireless access points), and the control layer consists of SDN controllers (such as Floodlight, Beacon, POX, NOX, MUL, Open Daylight etc.) and an application layer that includes Applications to configure SDN (access control, traffic / security control, energy saving networks, network management, etc.).

One of the main advantages of the SDN architecture is that the security perimeter can be expanded to include network access point devices (access keys, wireless access points, etc.) by setting security policy rules on network devices [7]. Via OpenFlow the SDN console creates a public network view by establishing a connection with the OpenFlow key.

In [14,15,16,17,18] some authors have proposed SDN frameworks and security applications. The main problem with their work is having one failure point with installing only one controller. In addition, security threats are another drawback, such as denial of service (DoS). If an attacker disables the SDN controller, it gains full control over the network, which creates potential risks for the entire network. In addition, hardware and software failures can occur on one control system. However, the presence of multiple consoles [10], [11] provides reliability and fault tolerance. If one of the controllers fails, another SDN can control and avoid system failure.

Open Day Controller Controller [19] supports high availability block-based model. The use of multiple consoles improves network performance. This is because each controller has partial visibility of the network and the controllers need to work together to exchange information. The interaction between the console and the open flow switch is basically a multi-key Openflow to multiple controllers.

Since version 1.2, the open flow has two conduction modes

Equal interaction: in this case all controllers have read / write access to the switch. That is, they must be synchronized in order not to step on one another.

Interaction between master/slave: In this case, there is one master and multiple slaves (there may be equal lenses).

3. SDN architecture for ad hoc networks

By default, the console is set in open daylight with an equivalent reaction. You have full access to the switch and all controllers have the same rules. Based on this approach, we suggest multiple SDN control structures for ad hoc networks. SDN-based structures include:

The old interface: physical layer,

Programmable Layer: SDN-compatible virtual switch and SDN controller.

OS and its applications: OS layer.

All old interfaces are connected to a default adapter, controlled by an SDN controller built into the node. You do not need to worry about node responsibilities if there is an unauthorized user connected, as in [13], because all the controllers in each node work the same way. At the same time, SDN controllers can improve security and communication between nodes with equivalent interactions.

One advantage of the new SDN-based dedicated network architecture is compatibility with legacy SDNs. Since each node of the ad hoc network contains an integrated SDN and an SDN controller, the ad hoc network can be linked to an outdated network to create an SDN domain (Figure 1).

In modern works like [12], the SDN domain is limited to infrastructure networks. In this configuration, dedicated users must connect through another node (network gateway) directly connected to the SDN domain. The proposed structure extends to the SDN to cover all custom devices. The suggested solution is to deploy an OpenFlow key like Open Switch [8] on each Ad-Hoc node. With this configuration, the Ad-Hoc node can connect to the network as part of the SDN domain, and can apply the same security rules that SDN domain users use. As shown in Figure 1, the proposed architecture supports networks with or without infrastructure.

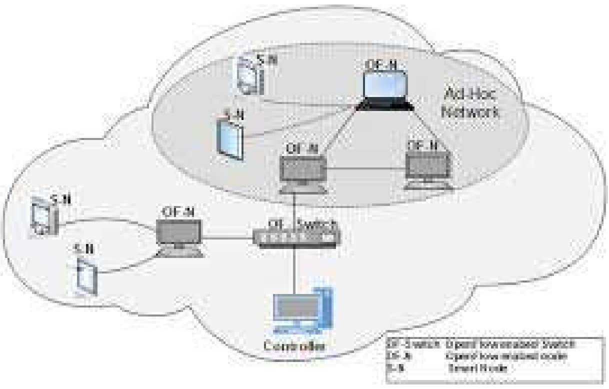

Fig1. SDN Domain interconnection - Extended SDN Domain

Since each Ad-Hoc node has its own SDN controller, the SDN Control Plane must control the evolution of each SDN virtual switch on each Ad-Hoc. When new custom devices call themselves or leave the network, they can exchange many messages to synchronize all the rules. For durability and tolerance, it is recommended to use an SDN architecture distributed with multiple controllers, [13]. To ensure this, new consoles are added dynamically to the Ad-Hoc network region, allowing nodes to perform controls. New consoles share the same network overview. However, its functionality and area of SDN management is limited to a small assigned area. Additionally, since the user is deploying these controllers, it is necessary to monitor the operation of the program key.

The proposed distributed distributed access control architecture allows you to quickly respond to network changes. It also responds to attacks in the SDN while sharing traffic management with roads.

Foreman. As mentioned earlier, custom control console functionality is limited and included in the resources available for the dedicated hosting device. By developing a framework that integrates OpenFlow adapters into these devices, we plan to extend the SDN to smart objects such as tablets, smartphones, and portable vehicles.

4. SDN-based architecture for IoT

Conventional network protocols and equipment are not designed to support high levels of scalability, heavy traffic, and portability. Diogo et al. Twenty people are proposing new architectural models for the Internet of Things. The author discusses the possibility of exploiting the ETSI M2M architecture by allowing devices to negotiate QOS and security standards with gateways. The author also discusses the idea of configuring in real time a cloud service connection that provides information about the connected device. The suggestion is that the Internet of Things be interoperable, scalable and adaptive. There are also papers [25] and [26] on software-defined approaches to the IoT environment. These papers focused on the SDN and IoT integration report, but they did not propose a security mechanism. Our system has proposed an SDN-based secure architecture for IoT and ad hoc networks.

4.1. SDN domain

In IoT or sensor networks, as suggested in the previous section, each device cannot have an SDN compatible key and an SDN controller. However, it can be assumed that each resource-deficient device can be associated with a single neighbor with SDN capability. In a heterogeneous network, as in Figure A, there are two types of nodes in the field. If the node has sufficient resources, it is called the node OF; otherwise, it is called a sensor or smart object. Each domain has an SDN controller that controls all traffic in this domain. Edge controllers in the SDN field interact equally and all rules are synchronized.

4.2. SDN connect domain

The proposed architecture with multiple SDNs assumes that each domain has one SDN controller or multiple SDN controllers. These controllers only manage devices in the domain. Domains represent enterprise networks or data centers.

SDN-based structures for the Internet of Things require heterogeneous links with many SDN domains. In order to achieve this interdependence on a large scale, a new type of controller is introduced in each field. This is the root controller, also called the boundary controller. Some authors [20], [21] and [22] have proposed an SDN hierarchy to improve control functions and their distribution. Instead of distributing control functions to multiple controllers, we suggest distributing routing functions and security rules to each edge controller. In addition, these controllers establish connections and exchange information with other SDN limit controllers (Figure 1).

The development of this architecture is based on demonstrating equal interaction between controllers using current safety mechanisms. Each SDN field has its own security policy and management strategy. To solve potential problems arising from failure to standardize security policies for each coherent SDN, Flauzac et al. The concept of the proposed security network is used. In [23]. Safety net is an intermediary program to implement network security in a distributed manner.

5. Distributed SDN security solution

Many studies have implemented SDN architectures by implementing firewalls [9, 10, 29, 30, 31,] IPS [11], NAC [7], and IDS modules [24, 27, 28] on top of SDN controllers or installing security policies. Researching the network security used. For OpenFlow switch. The advent of the next generation of Internet architecture requires higher levels of security, including authentication of network devices, users, and objects that connect to users using both wired and wireless technologies. In addition, you need to monitor the behavior of both users and objects, establish trust boundaries, and use accounting techniques with software validation. However, existing security mechanisms [7,9,10,11] do not provide these security levels to meet the security needs of the next generation Internet architecture.

Inspired by existing network access control and security techniques [3], it designs an extended secure SDN-based architecture for IoT. To illustrate the architecture, we first present a simple solution where the controller manages the security of one SDN domain. Second, this first solution can be extended to include multiple controllers for the resources available on each control platform. It also extends the distributed control architecture by interconnecting all SDN domains via border controllers. This allows you to approach a secure model of the IoT.

Traditional ad hoc architectures do not provide network access control or global traffic monitoring because there is no network infrastructure. The architecture proposed in this article overcomes these security limitations and allows for dynamic network configuration and security policy deployment.

To protect network access and network resources, SDN controllers begin by authenticating network devices. Once the device is authenticated, the controller pushes the appropriate flow entry to the software or hardware access switch.

The overall concept of the security network grid is to extend the concept of the SDN domain to multiple domains (Figure 2). Each controller in each domain exchanges security rules with controllers in other domains. To ensure network security, there is an SDN controller that acts as a security guard at the edge of the SDN domain.

We have begun implementing the solution with an Open daylight controller and LXC virtual machine connected to open switch.

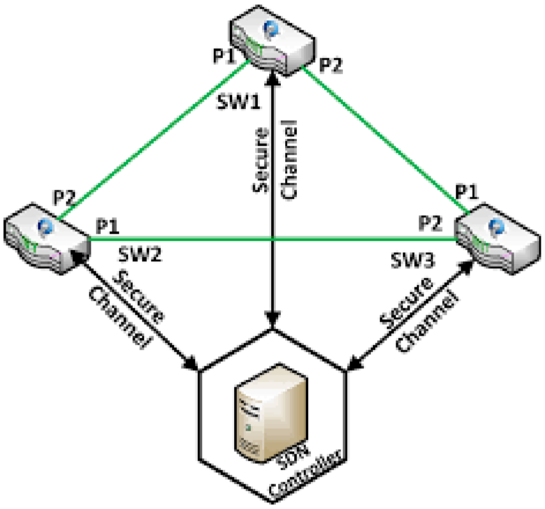

Fig2. Grid of Security in SDN Domain-LLDP

6. Conclusion

This paper has outlined a new SDN-based network architecture with distributed controllers. In addition, our solutions can be used in the context of ad hoc networks and

IoT. First, we introduced a new architecture where multiple SDN controllers interact equally. Next, we proposed a scalable architecture with multiple SDN domains. Each domain can be configured with or without infrastructure, and each controller is responsible for that domain only. Communication between domains is performed by special controllers called border controllers. These edge controllers need to work with new distributed interactions to ensure the independence of each domain in the event of a failure. It employs an architecture that guarantees the security of the entire network, and incorporates a security grid concept into each controller to prevent attacks.

Future work will explore the characteristics of the extended SDN domain further, explore more security mechanisms, and explore the potential for using them in the context of SDN. In addition, the system will be tested on a larger scale to leverage the architecture framework of Open daylight and optimize the system design. Build this architecture and work to test it in a real environment.

References

[1] Internet World Stats, Internet Usage Statistics. 2014;*[Online]. Available: http://www.internetworldstats.com/stats.htm/*.

[2] Internet Security Threat Report 2014. *[Online]. Available: http://www.symantec.com/*.

[3] Open Networking Foundation. *[Online]. Available: https://www.opennetworking.org/*.

[4] Moreno Sanchez P, Marin Lopez R, Gomez Skarmeta AF. A Network Access Control Implementation Based on PANA for IoT Devices. *Sensors* 2013. p. 14888-14917.

[5] Sezer S, Scott-Hayward S, Chouhan PK and Fraser B, Lake D, Finnegan J, and Viljoen N, Miller M, Rao N. Are we ready for SDN? Imple-mentation challenges for software-defined networks. *Communications Magazine, IEEE* 2013. p. 36-43.

[6] Tootoonchian and A, Ganjali Y. Hyperflow: A distributed control plane for openflow. *Internet Network Management Conference on Research on Enterprise Networking*. 2010. p. 3.

[7] Nunes B, Santos M, de Oliveira B, Margi C, Obraczka K, Turletti T. Software defined networking enabled capacity sharing in user-centric network. *IEEE Communications Magazine*. vol. 52, 2014. p. 28-36.

[8] Scott-Hayward S, OCallaghan G, Sezer S. SSDN security: A survey. *IEEE SDN for Future Networks and Services*. 2013. p. 1-7.

[9] Son S, Shin S, Yegneswaran V, Porras P, Gu G. Model checking invariant security properties in openflow. *IEEE International Conference on Communications*. 2013. p. 19741979.

[10] Hu H, Han W, Ahn J, Zhao Z. Flowguard: Building robust firewalls for software-defined networks. *Third Workshop on Hot Topics in Software Defined Networking*. 2014. p. 97-102.

[11] Jin R, Wang B. Malware detection for mobile devices using software-defined networking. *Workshop of Research and Educational*. 2013. p. 81-88.

[12] Skowyra R, Bahargam S, Bestavros A. Software-defined ids for securing embedded mobile devices. *High Performance Extreme Computing Conference* 2013. p. 1-7.

[13] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. Openflow: Enabling innovation in campus networks. *SIGCOMM Computer Communication*. 38, 2008. p. 69-74.

[14] De Rubertis A, Mainetti L, Mighali V, Patrono L, Sergi I, Ste- fanizzi M, Pascali S. Performance evaluation of end-to-end security protocols in an internet of things. 21st International Conference on Software, Telecommunications and Computer Networks. 2013. p. 1-6.

[15] Braga R, Mota E, Passito A. Lightweight DDoS flooding attack detection using NOXIOpenFlow, in Local Computer Networks (LCN). 2010 IEEE 35th Conference on. IEEE 2010. p. 408-415.

[16] Jafarian H, Al-Shaer E, Duan Q. Open flow random host mutation: transparent moving target defense using software defined networking, First workshop on Hot topics in software defined networks. ACM, 2012. p. 127-132.

[17] P. Meenalochini and S. P. Umayal ,Comparison of Current Controllers on Photo Voltaic Inverters Operating as VAR Compensators, Journal of Electrical Engineering The Institution of Engineers, Bangladesh Vol. EE 38, No. I, June, 2012.

[18] Karthick, R and Sundararajan, M: "A Reconfigurable Method for TimeCorrelatedMimo Channels with a Decision Feedback Receiver," International Journal of Applied Engineering Research 12 (2017) 5234.

[19] Karthick, R and Sundararajan, M: "PSO based out-of-order (ooo) execution scheme for HT-MPSOC"Journal of Advanced Research in Dynamical and Control Systems 9 (2017) 1969.

[20] Karthick, R and Sundararajan, M: "Design and Implementation of Low Power Testing Using Advanced Razor Based Processor," International Journal of Applied Engineering Research 12 (2017) 6384.

[21] Karthick, R and Sundararajan, M: "A novel 3-D-IC test architecture-a review," International Journal of Engineering and Technology (UAE)7 (2018) 582.

[22] R.Karthick, P Selvaprasanth, A ManojPrabaharan, "Integrated System For Regional Navigator And Seasons Management," Journal of Global Research in Computer Science 9(4),2018(11-15).

[23] Karthick, R and Prabaharan, A.Manoj and Selvaprasanth, P. and Sathiyanathan, N. and Nagaraj, A., High Resolution Image Scaling Using Fuzzy Based FPGA Implementation (March 15, 2019). Asian Journal of Applied Science and Technology (AJAST), Volume 3, Issue 1, Pages 215-221, Jan-March 2019 . Available at SSRN: https://ssrn.com/abstract=3353627

[24] Karthick, R and Sundararajan, M., Hardware Evaluation of Second Round SHA-3 Candidates Using FPGA (April 2, 2014). International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014), Vol. 2, Issue 2, Ver. 3 (April - June

2014). Available at SSRN: https://ssrn.com/abstract=3345417.

[25] Karthick, R and and Prabaharan, A.Manoj and Selvaprasanth, P.,Internet of Things based High Security Border Surveillance Strategy (May 24, 2019). Asian Journal of Applied Science and Technology (AJAST), Volume 3, Issue 2, Pages 94-100, Apr-June 2019. Available at SSRN: https://ssrn.com/abstract= 3394082.

[26] Karthick, R and Sundararajan, M: "SPIDER based out-of-order (ooo) execution scheme for HT-MPSOC" International Journal of Advanced Intelligence paradigms, In Press.

[27] Karthick, R and John Pragasam, D "Design of Low Power MPSoC Architecture using DR Method" Asian Journal of Applied Science and Technology (AJAST) Volume 3, Issue 2, Pages 101-104, April -June 2019.

[28] Karthick, R and Sundararajan, M., Optimization of MIMO Channels Using an Adaptive LPC Method (February 2, 2018). International Journal of Pure and Applied Mathematics, Volume 118 No. 10 2018, 131-135. Available at SSRN: https://ssrn.com/abstract=3392104

[29] Karthick, R and Rinoj, B. Micheal Vinoline and Kumar, T. Venish and Prabaharan, A.Manoj and Selvaprasanth, P., Automated Health Monitoring System for Premature Fetus (July 27, 2019). Asian Journal of Applied Science and Technology (AJAST) (Peer Reviewed Quarterly International Journal) Volume 3, Issue 3, Pages 17-23, July -September 2019. Available at SSRN: https://ssrn.com/abstract=3427756

[30] Karthick, R., Deep Learning For Age Group Classification System, International Journal Of Advances In Signal And Image Sciences. Volume 4, Issue 2, Pages 16-22, 2018.

[31] R. Karthick, N. Sathiyanathan, "Medical Image Compression Using View Compensated Wavelet Transform" Journal of Global Research in Computer Science 9(9), 2018(1-4).