Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 1028 – 1033

Procedia Computer Science

International Workshop on Big Data and Data Mining Challenges on IoT and Pervasive Systems (BigD2M 2015)

New Security Architecture for IoT Network

FLAUZAC Olivier, GONZALEZ Carlos, NOLOT Florent

University of Reims Champagne-Ardenne, Laboratory CReSTIC, 51100 Reims, France

Abstract

We explain the notion of security architecture for Internet of Things (IoT) based on software-defined networking (SDN). In this context, the SDN-based architecture works with or without infrastructure, that we call SDN-Domain. This work describes the operation of the proposed architecture and summarizes the opportunity to achieve network security in a more efficient and flexible with SDN. An overview of existing SDN security applications were discussed and tackles its issues, presenting a new IoT system's architecture. In this paper we considered the network access control and global traffic monitoring for ad-hoc networks. Finally, we point out architectural design choices for SDN using OpenFlow and discuss their performance implications.

Peer-review under responsibility of the Conference Program Chairs

Keywords: IoT, Software Defined Network, SDN, Security, Authenticity

1. Introduction

Internet is growing rapidly in the last decades and continues to develop in terms of dimension and complexity. At the end of 2014, 42.3% of the world population was connected to the network[1]. Nevertheless, the network security threats increase with internet evolution. A special concern will be dedicated to the security of the Internet of Things (IoT), since it will include every object or device with networking capabilities. Objects can include simple home sensors, medical devices, cars, airplanes and even nuclear reactors and other things, which can pose risks to human life. The number of breaches in 2013 was 62% percent greater than in 2012[2].

Traditional security mechanisms like Firewalling, Intrusion Detection and Prevention Systems are deployed at the Internet edge. Those mechanisms are used to protect the network from external attacks. Such mechanisms are no longer enough to secure the next generation Internet. The borderless architecture of the IoT raises additional concerns over network access control and software verification. In[4], the authors present details of network access control implementation based for IoT devices named as (PANATIKI).

* Corresponding author. Tel.: (+33) 3 26 91 32 15 ; fax: (+33) 3 26 91 33 97.
E-mail address: olivier.flauzac@univ-reims.fr, carlos.gonzalez-santamaria@etudiant.univ-reims.fr, florent.nolot@univ-reims.fr

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.05.099

Recent advances in computer networking have introduced a new technology paradigm for future communication, the Software Defined Networks (SDN). A central software program, called SDN controller, manages the overall network behavior. Into SDN the control and data planes are decoupled, network intelligence is logically centralized. The controller can add, update, and delete flow entries, both reactively in response to packets and proactively with predefined rules. In addition, SDN enables fast reaction to security threats, granular traffic filtering, and dynamic security policies deployment.

Based on the SDN architectures we propose a security model for the IoT. Firstly, the proposed security model was designed to establish and secure both wired and wireless network infrastructure. Secondly, we extended the proposed architecture in order to include Ad-Hoc networks and network object things such as: sensors, tablets, smart phones, etc. Our main contributions are as follows:

- To the best of our knowledge, this is the first effort that uses the SDN architecture to tackle security issues in the IoT.
- Inspired by existing Network Access Control and security techniques, we design a secure SDN-based architecture for the IoT.
- Based on a Grid of Security paradigm[23], we enhance security policies exchange and deployment between SDN control domains.

Our security model is discussed later in this article, and we conclude with the outline of our vision for the SDN based security on solutions for the IoT.

2. Software-Defined Networking Architecture

Software-Defined Networking (SDN) emerged as a strategy to increase the functionality of the network, reducing costs, reducing hardware complexity and enabling innovative research. SDN architecture models have three layers[3,5]: an infrastructure layer consists of network devices (e.g., switches, routers, virtual switches, wireless access point), a control layer consists of SDN controller(s) (e.g., Floodlight, Beacon, POX, NOX, MUL, Open daylight, etc.) and an application layer that includes the applications for configuring the SDN (e.g., Access control, traffic/security monitoring, energy-efficient networking, management of the network).

One important feature of SDN architecture is its ability to extends the security perimeter to the network access end point devices (access switches, wireless access points, etc.), by setting up security policy rules to network devices[7]. via the OpenFlow protocol, the SDN controller builds a global network view by establishing connection with the OpenFlow switches.

In[15,16,17,18], framework and security applications for SDN have been proposed by some authors. The main issue of their works is the presence of a single point of failure with only one controller installed. Furthermore, security threats are another drawback such as a Denial of Service (DoS). If an attacker compromises the SDN controller, then he has full control over the network and this poses a potential risk to the entire network. Additionally, hardware and software failures can occur with a single controller system. However, having multiple controllers[19] provides trustworthiness and fault tolerance. If one of the controllers goes down, another SDN controller can take control to avoid system failure.

The Open Daylight Controller[19] supports a Cluster-based High Availability model. There is increased network performance with multiple controllers, because each controller has a partial view of the network, and the controllers have to collaborate and exchange information with each other. The interaction between the Controller(s) and the Open-Flow enabled switches is essentially to have one Openflow switch multi-homed to multiple controllers.

After version 1.2, Openflow includes two modes to interconnect multiple controllers in the network:

- Equal interaction: in this case all controllers have read/write access to the switch, which means they have to synchronize in order not to step on each other's feet.
- Master/Slave interaction: in this case there will be one master and multiple slaves (there could be still multiple equals as well).

3. SDN architecture for Ad-Hoc Network

The OpenDaylight Controller is set up by default on equal interaction. It has full access to the switch, and all controllers have the same rules. Based on this approach, we propose Multiple SDN Controller architecture for Ad-Hoc Networks. An SDN-based architecture involves the:

- legacy interfaces : the physical layer,
- programmable layer : SDN-compatible virtual switch and an SDN controller
- operating systems and their applications : the OS layer

All legacy interfaces are connected to a virtual switch, and this switch is controlled by an SDN controller, integrated on the node. Since all controllers of each node operate in equal interaction, they will have no need to be concerned about nodes liability for the misbehaving users connecting through them, as in[13]. Ad-Hoc users will connect with other nodes through their embedded SDN-compatible switch. At the same time, the SDN controller, in equal interaction, can enhance the security and connectivity between the nodes.

One of the advantages of this new SDN-based Ad-Hoc network architecture is its compatibility with SDN legacy network. Since each node in the Ad-Hoc network has an embedded SDN-compatible switch and an SDN controller, we can interconnect the Ad-Hoc network to the legacy network to construct an SDN-domain (Fig. a).

In a most recent work like[12], the SDN domain is limited to the network with infrastructure. In this configuration, Ad-Hoc users have to connect through other nodes (Network gateway) directly connected to the SDN domain. In our proposed architecture, the SDN domain is extended in order to include all Ad-Hoc devices. Our proposed solution consists in deploying an OpenFlow software switch, such as Open vSwitch[8] in each Ad-Hoc node. This configuration enables Ad-Hoc nodes to connect to the network as part of the SDN domain, so we can apply the same security rules as for the SDN domain users. As show in (Fig. a), the proposed architecture supports networks with or without infrastructure.

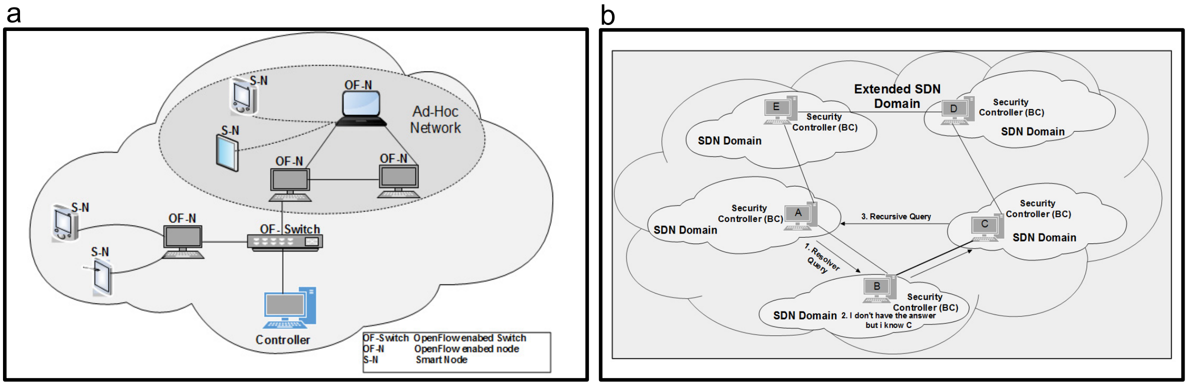

Fig. 1. (a) Domain-SDN; (b) SDN Domain interconnection - Extended SDN Domain.

As each Ad-Hoc node has its own SDN controller, the SDN control plane has to manage the evolution of each SDN virtual switch on each Ad-Hoc device. When a new Ad-Hoc device connects itself or leaves the network, we can have many exchanged messages in order to synchronize all the rules. In order to ensure scalability and fault tolerance, a distributed SDN architecture is preferred, with multiple controllers as in[13]. To ensure that, we dynamically add new controllers to the Ad-Hoc network area and authorize special nodes to run control operation. The new controllers will share the same network global view. However, their functions and SDN management domain will be limited to a small Ad-Hoc area. Furthermore those controllers will be responsible for monitoring the behavior of the software switches, since they are deployed at the user side.

Our proposed distributed network access control architecture enables faster response to network changes. Moreover, it reacts to attacks occurring in the SDN domain, while sharing the traffic load management with the root controller. As mentioned earlier, control functions of Ad-Hoc controllers will be limited and adapted to the available resources of the hosting Ad-Hoc device. We intend to extend the SDN domain even more, to include smart object such as: tablets, smart phones, mobile vehicles, etc. by developing a framework that integrates OpenFlow software switches into those devices.

4. SDN based architecture for IoT

The traditional network protocols and equipment are not designed to support high levels of scalability, high amounts of traffic and mobility. Diogo et al.[20] have proposed an new architecture model for IoT. The authors argue about the possibility of exploiting ETSIs M2M Architecture, by allowing the device to negotiate QOS and security parameters with the Gateway. Also the authors discuss the idea of real-time configuration of Cloud Service connectivity that provide information about the device connected to it. It's proposal is to have an interoperable, scalable and adaptable IoT. Besides, there exist some papers[25,26] of software-defined approach for the IoT environment. These papers had the focus of determining the integration of SDN and IoT, it's do not propose a security mechanism. Our system proposed a secure SDN-based architecture for IoT and Ad-hoc networks.

4.1. SDN Domain

In IoT or in sensor networks, each device cannot have an embedded SDN-compatible switch and an SDN controller as we have proposed in the previous section. But, we can assume that each device, with low resource can be associated to one neighbor node which has the SDN capability. In a heterogeneous network as in (Fig. a) we have two types of nodes in a domain. If the node has enough resources, we called it an OF node, and if not, we called it a sensor or a smart object. Each domain has its SDN controller which controls all traffic in its domain. The edge controller of the SDN domain is in equal interaction, and all rules will be synchronized.

4.2. SDN Domain interconnection

In the proposed architecture with multiple SDN domains, we assume that in each domain, we have one SDN controller or multiple SDN controllers. These controllers manage only the devices in its domain. A domain represents an enterprise network or a datacenter.

An SDN-based architecture for the IoT requires heterogeneous interconnection with larger numbers of SDN domains. In order to achieve such large scale interconnection, we introduce a new type of controller in each domain : the root controller, that we can also call a Border controller. Some authors[21,22] propose hierarchical architecture for SDN, to optimize and distribute control functions. We propose not to distribute control functions on multiple controllers but to distribute routing functions and security rules on each edge controller. Moreover these controllers are responsible for establishing connections and exchanging information with other SDN border controllers (Fig. b).

The development of this architecture is based on the perspective of equal interaction between controllers, using the existing security mechanisms. Each SDN domain has its own security policies and management strategy. To solve potential problems raised by the heterogeneity of the security policies respective to the interconnected SDN domains, we use the Grid of Security concept proposed by Flauzac et al. in[23]. The Grid of Security is a middleware for decentralized enforcement of the network security.

5. Distributed SDN Security Solution

Many works have studied network security using the SDN architecture, either by implementing firewalls[9,10,29,30,31], IPS[11], NAC[7] and IDS modules[24,27,28] on top of the SDN controller, or by installing security policies into OpenFlow switches. The emergence of the next generation Internet architecture, requires even higher security levels, such as authenticating network devices, users and objects connecting to users using both wired and wireless technologies. Furthermore, we need to monitor the behavior of both the users and the objects, establish trust boundaries, and use accounting methods along with software verification. However, existing security mechanisms[7,9,10,11] do not provide these security levels to meet the security needs of next generation Internet architecture.

Inspired by existing Network Access Control and security techniques[13], we design an extended secure SDN-based architecture for the IoT. To explain our architecture, we first present a simple solution in which a controller manages the security of one SDN domain. Second, we can extend this first solution to include multiple controllers with regard to the available resources on each control platform. In addition, we extend the distributed control architecture by interconnecting all SDN domains via border controllers, which leads us close to a secure model for the IoT.

Traditional Ad-Hoc architecture does not provide network access control or global traffic monitoring, due to the absence of the network infrastructure. The architecture proposed in this article overcomes those security limitations and enables dynamic network configuration and security policies deployment.

In order to secure network access and network resources, the SDN controllers begin by authenticating the network devices. Once the device is authenticated, the controller will push the appropriate flow entries to the software or the hardware access switch.

The whole concept of the grid of security network is to extend the SDN domain concept to multiple domains (Fig. 2), and each controller of each domain exchanges its security rules with controllers of other domains. There are SDN controllers which behave as security guards on the edge of the SDN Domain to ensure the network safety.

We have started to implement our solution with Opendaylight controller and LXC virtual machines connected to an openvswitch.

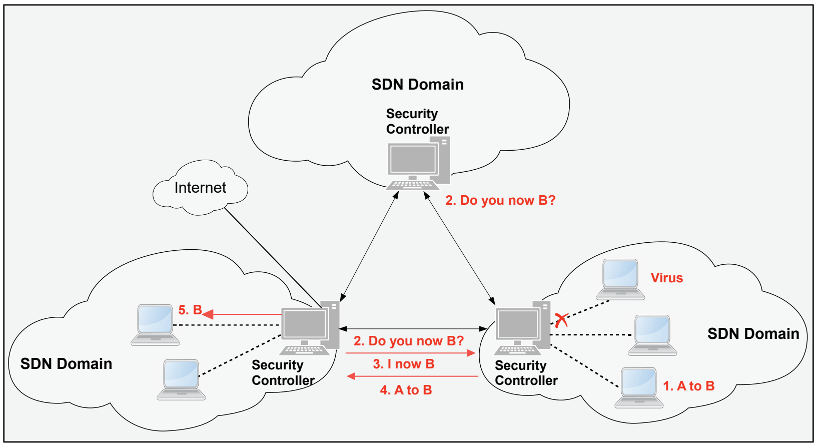

Fig. 2. Grid of Security in SDN Domain.

6. Conclusion

In this paper, we have provided an overview of a new SDN-based network architectures with distributed controllers. Moreover, our solution can be used in the context of Ad-Hoc networks and IoT.

First, we presented a new architecture with multiple SDN controllers in equal interaction. Second, we proposed an architecture which is scalable with multiple SDN domains. In each domain we can have networks with or without infrastructure and each controller is responsible only for its domain. The communications between domains is made with special controllers called Border Controllers. These edge Controllers have to work in a new distributed interaction in order to guarantee the independence of each domain in case of failure. We adopt an architecture to guarantee the security of the entire network with the concept of grid of security embedded in each controller to prevent attacks.

As future work, we will further study the characteristics of the extended SDN-Domain, investigate more security mechanisms and explore the possibilities of employing them in the context of SDN. In addition, we plan to take better advantage of the architectural framework of Opendaylight and test our system at an even larger scale in order to optimize our system design. We shall work to build this architecture and test it in a real environments.

Acknowledgements

This work was supported by SENACYT-Panama, Secretaría Nacional de Ciencia, Tecnología e Innovación.

References

1. InternetWorldStats, 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? Implementation challenges for software-defined networks. *Communications Magazine, IEEE* 2013. p. 36-43.
6. Tootoonchianand 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. Shin S, Porras P, Yegneswaran V, Fong M, Gu G, Tyson M. FRESCO: Modular composable security services for software-defined networks, *Network and Distributed Security Symposium* 2013.
18. Shin S, Gu G. "CloudWatcher, Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?), in Network Protocols (lCNP), *20th IEEE International Conference on. IEEE*, 2012. p. 1-6.
19. Network Functions Virtualization (NFV). *OpenDaylight*, http://www.opendaylight.org/. [Online].
20. Diogo P, Reis LP, Vasco Lopes N. Internet of Things: A system's architecture proposal. *9th Iberian Conference on Information Systems and Technologies (CISTI)*. 2014. p. 18-21.
21. Shuai Gao, Yujing Zeng, Hongbin Luo, Hongke Zhang, Scalable area-based hierarchical control plane for software defined information centric networking. *23rd International Conference on Computer Communication and Networks (ICCCN)*. 2014. p. 4-7.
22. Xu Li, Djukic P, Hang Zhang. Zoning for hierarchical network optimization in software defined networks. *IEEE Network Operations and Management Symposium (NOMS)* 2014. p. 5-9.
23. Flauzac O, Nolot F, Rabat C, Steffenel L. Grid of security: A new approach of the network security *Third International Conference on Network and System Security*. 2009. p. 67-72.
24. Kreutz D, Ramos F, Verissimo P. Towards secure and dependable software-defined networks, *second ACM SIGCOMM workshop on Hot topics in software defined networking*. 2013. p. 55-60.
25. Zhijing Qin, Denker G, Giannelli C, Bellavista P, Venkatasubramanian N. Software Defined Networking architecture for the Internet-of-Things. *Network Operations and Management Symposium (NOMS)*. 2014. p. 1-9.
26. Martinez P, Skarmeta A. Empowering the Internet of Things with Software Defined Networking. *FP7 European research project on the future Internet of Things*.
27. Song S, Hong S, Guan X, Choi BY, Choi C. Neod: Network embedded on-line disaster management framework for software defined networking. *Integrated Network Management IFIP/IEEE International Symposium* 2013. p. 492-498.
28. Oktian YE, Lee S, Lee H. Mitigating denial of service (dos) attacks in open flow network. *Information and Communication Technology Convergence (ICTC)* 2014. p. 325-330.
29. Pena J, Yu W. Development of a distributed firewall using software defined networking technology. in *Information Science and Technology (ICIST), 4th IEEE International Conference 2014*. p. 81-88.
30. Suh M, Park SH, Lee B, Yang S. Building firewall over the software-defined network controller. *16th Advanced Communication Technology (ICACT)*. 2014. p. 744-748.
31. Javid T, Riaz T, Rasheed A. A layer2 firewall for software defined network. *Information Assurance and Cyber Security (CIACS)Conference* 2014. p. 39-42.