Intrusion Prevention System Evaluation for SDN-Enabled IoT Systems

Alexandru Stancu, Stefan-Ciprian Arseni, Alexandru Vulpe(✉), Octavian Fratu, and Sinoma Halunga

University Politehnica of Bucharest, 060042 Bucharest, Romania
{alex.stancu,stefan.arseni,alex.vulpe}@radio.pub.ro,
shalunga@elcom.pub.ro

Abstract. As the importance of communication networks increases in our lives, the limitations of traditional networks start to emerge. Software Defined Networking (SDN) is the most recent paradigm in the networking industry, its purpose being to mitigate traditional network limitations, such as complexity, the difficulty of introducing new services in the network, the inability of enforcing security policies while having a network-wide view. From a security point of view, the need for middleboxes in the network, such as firewalls or Intrusion Detection/Prevention Systems (IDS/IPS) is eliminated by implementing these functionalities in software applications. As SDN has the potential of becoming a key enabler for the Internet of Things (IoT), there are specific aspects of security for IoT that need to be taken into account, for example the lack of powerful computing resources or limited battery life, making securing IoT devices more challenging. This paper addresses one of these security issues, while evaluating a simple IPS application for an SDN controller. An emulated IoT network is controlled by the SDN controller, which also runs an IPS application. When a node becomes faulty or it is compromised and it sends too much traffic, that could cause a Denial of Service (DoS) in the network, it is blocked by the controller for a configurable amount of time.

Keywords: Security · Wireless Sensor Networks
Intrusion detection · Software Defined Networking
Internet of Things

1 Introduction

Software Defined Networking (SDN) and Internet of Things (IoT) are two of the most popular recent paradigms in the research community. IoT represents the interconnection of physical items (devices, vehicles, buildings, appliances) that are capable of network connectivity in order to collect and exchange data. SDN is an emerging architecture that decouples the network data plane from the control plane making the network control directly programmable through software

N. Oliver et al. (Eds.): MindCare 2016/Fabulous 2016/IIoT 2015, LNICST 207, pp. 145–150, 2018.
https://doi.org/10.1007/978-3-319-74935-8_21

applications and abstracting the underlying infrastructure for the network services and applications. It appeared as a solution for mitigating the limitations that traditional networks have proven, such as complexity, vendor dependency, network policies that are not consistent, difficult network management [1].

SDN is beginning to become a key enabler for new concepts, such as IoT, or Cloud Computing, because it satisfies their needs, such as dynamic network reconfiguration, demand of higher bandwidth or simplified network architectures that ease innovation [2].

Functions previously obtained through middle-boxes could be achieved in software applications that run on top of the SDN controller. This has been demonstrated in [3], where an IPS application was implemented for the POX SDN controller.

An example of architecture for security in SDN-enabled IoT networks is defined in [4]. The authors describe how the security of each domain can be enhanced and how to distribute the security rules in order not to compromise the security of one domain in the case of multiple interconnected domains. However they provide no experimental evaluation of their architecture.

Authors in [5] define a SDN architecture for IoT based on Object Management Group's data distribution service (DDS) middleware. They do not, however, study security aspects for this architecture. Finally, the combination of Software Defined Wireless Networking (SDWN) and Wireless Sensor Networks is evaluated against popular networks such as ZigBee and 6LoWPAN in [6]. Authors perform extensive campaign measurements on the EuWin platform, but they evaluate only the protocol stacks of the three solutions, and do not take security into account.

The paper is organized as follows: Sect. 2 presents security aspects that are specific to SDN, IoT and the combination of these two concepts. Section 3 presents the methodology that was used for deploying and evaluating an IPS application for an emulated SDN-enabled IoT system, while Sect. 4 presents and analyses the obtained results. Section 5 highlights the impact of the results and possible future research directions, drawing the conclusions.

2 Security Aspects in SDN and IoT

As far as security is concerned, Software Defined Networking has both advantages and disadvantages. A major advantage is that it enables enhanced network security by its ability to redirect or filter traffic flows based on content or network states. The major disadvantage is that SDN is more vulnerable to threats because of the existence of the logically centralized controller.

On the other hand, the rise of the Internet of Things brings about numerous security issues, caused by humans' ever increasing reliance on intelligent devices in most aspects of their lives. These become subject to attacks and intrusions that have the ability to compromise personal privacy or threaten public safety. Such concerns have been addressed in multiple scientific papers that present different views on how IoT security issues have been or are being resolved, but also on key problems that security for IoT needs to address for IoT to become a dependable concept [7–9].

Through the integration of SDN in IoT systems, a part of the security concerns can be addressed, as presented in [10]. By allowing a high level of customization, SDN has become a key concept in the implementation process and also in the evolution of IoT systems [11].

3 Methodology

In mininet, a simple tree-like IoT topology was emulated. It contains four Office Gateways, each having five types of sensors. The traffic from every other two Office Gateways is aggregated into a Floor Gateway and then every other two Floor Gateways are aggregated into a Company Gateway. In mininet, the sensors are represented as hosts, and the gateways are considered to be switches (emulated as Open Virtual Switches). ONOS was chosen as the SDN controller for the network, based on several reasons, as described in [12].

Next, an application for ONOS, representing a simple IPS was implemented. Every five seconds, the controller polls through the OpenFlow protocol, the port stats for every device and if traffic passed through a specific port, the IPS application will compute the amount of throughput it received from the host, in kbps. It will then compare that value with a chosen threshold value of 225 kbps, considering a normal traffic pattern of 125 kbps for each host. If the value exceeds that threshold, then a flow rule is installed on the device, dropping all traffic from that port, having a timeout of 60 s, giving the attacked server a good amount of time to process the traffic that was sent until the node was considered malicious. This behavior simulates an IPS.

The third step in the methodology was evaluating the application. Iperf3 was used for generating traffic between the sensors and the server. Three phases of evaluating the application were considered. The first phase consisted in running the mininet topology and connecting it to the ONOS controller, without the IPS application enabled. An iperf3 server was started on the host connected to the Company Gateway, referred to as “Server”. After that, an iperf3 client was started on each of the sensors, transmitting UDP traffic to the server, with a throughput of 125 kbps, for a period of 60 s. Also, ping was started from each of the hosts to the Server. Average RTT and jitter were measured by the ping, as well as the jitter and packet loss by the iperf3 server. These values were used to see the normal behavior of the network. The second phase of testing consisted in taking the same measurements, without the IPS application running on the ONOS controller. This time, eight of the sensors were considered to be malicious, and this situation was simulated by sending traffic with a rate of 250 kbps from those hosts. The third phase was identical to the second one, except for the IPS application, that was enabled in the SDN controller.

4 Experimental Results

Several network parameters were considered for evaluating the application: the average RTT of the ICMP packets from the sensors to the Server and the standard deviation of the latency for that type of traffic, as measured by the ping tool. Also, the jitter, as measured by the iperf3 client was taken into account.

The ping results from the compromised nodes reveal the amount of time needed by the IPS application to detect the malicious traffic and block it. In ten of the twelve cases, the ping stops after 10 s, and in the other two cases it stops after 15 s. This means an average value of 11.25 s until the faulty node is blocked from the network. The parameters measured with the iperf3 tool highlight other aspects of the traffic in the network. The jitter of the UDP traffic between the clients and the Server increases in 58% of the cases. Such increases of the jitter can drastically affect the performance of the network. After the IPS application is enabled in the ONOS controller and the same tests are conducted, an improvement is observed. In the case of the jitter, the affected nodes percentage decreases to 33%.

The RTT and jitter variations in time are presented in Figs. 1 and 2. For each graphic, three situations were presented: (a) normal traffic conditions, malicious traffic present in the network while the IPS application is disabled and malicious traffic while the IPS application is enabled.

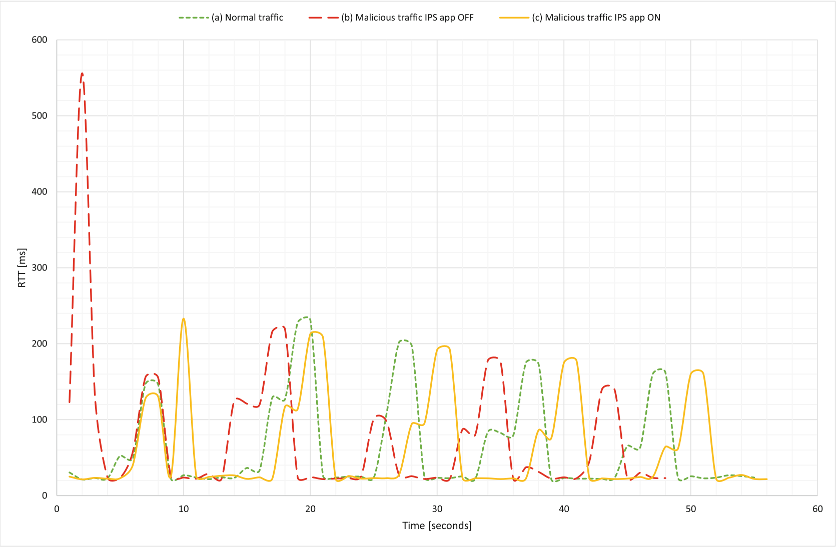

Fig. 1. RTT variation

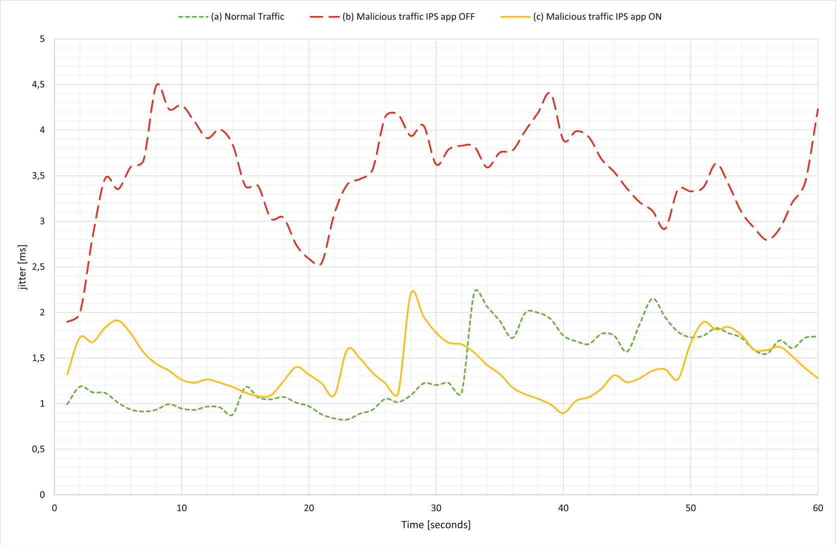

Fig. 2. Jitter variation

5 Conclusion

Software defined networking is proving to become an important enabler for a rapid and safe implementation of the Internet of Things paradigm. Although the flexibility that SDN brings improves the easiness of integrating dynamically configurable security solutions, there are still issues that need to be addressed.

Through this paper we made an assessment on the performance variation of an SDN-enabled IoT topology, when integrating an IPS application. The simple yet relevant implementation lead to some results that can be applied even for a more comprehensive simulation of a larger IoT system topology. We can state that the basic discovery and control information transmitted throughout the network was not affected by the occurence of some faulty nodes, but there was a drop in performance for the overall network, when faulty nodes were activated. After enabling the IPS application, the drop in performance lasted for a short period of time that would not create an accentuated ripple effect throughout the network.

In conclusion, even simple SDN security applications with a customizable implementation can ensure a minimum level of protection for a network. By integrating the SDN security principle, the internal network is assured with a sufficient level of confidentiality and integrity of data.

Acknowledgments. This work has been funded by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB – GEX. Identifier: UPB–EXCELENTA–2016 project "Platform for Studying Security in IoT", contract number 96/2016 (PaSS-IoT), by UEFISCDI Romania under grants no. 20/2012

"Scalable Radio Transceiver for Instrumental Wireless Sensor Networks - SaRaT-IWSN" and 262EU/2014 eWALL support project and by the European Commission by FP7 IP project no. 610658/2013 "eWALL for Active Long Living - eWALL".

References

1. Stancu, A., Halunga, S., Suciu, G., Vulpe, A.: An overview study of software defined networking. In: 2015 14th International Conference on Informatics in Economy (IE 2015), Bucharest, pp. 50–55, 30 April–3 May 2015
2. Vilata, R., Munoz, R., Casellas, R., Martinez, R.: Enabling internet of things with software defined networking. CTTC (2015)
3. Akin, G., Karaarslan, E., Bük, O., Uçar, E.: SDN architecture fundamentals and DOS prevention basics: a case study with openflow. In: International Scientific Conference, UNITECH 2015, Gabrovo (2015)
4. Flauzac, O., González, C., Hachani, A., Nolot, F.: SDN based architecture for IoT and improvement of the security. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Gwangiu, pp. 688–693 (2015). https://doi.org/10.1109/WAINA.2015.110
5. Hakiri, A., Berthou, P., Gokhale, A., Abdellatif, S.: Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications. IEEE Commun. Mag. **53**(9), 48–54 (2015). https://doi.org/10.1109/MCOM.2015.7263372
6. Buratti, C., et al.: Testing protocols for the internet of things on the EuWIn platform. IEEE Internet Things J. **3**(1), 124–133 (2016). https://doi.org/10.1109/JIOT.2015.2462030
7. Jing, Q., Vasilakos, A.V., Wen, J., Jingwei, L., Qiu, D.: Security of the Internet of Things: perspectives and challenges. Wirel. Netw. **20**(8), 2481–2501 (2014)
8. Sicaria, S., Rizzardia, A., Griecob, L.A., Coen-Porisinia, A.: Security, privacy and trust in Internet of Things: the road ahead. Comput. Netw. **76**, 146–164 (2015)
9. Nguyen, K.T., Laurent, M., Oualha, N.: Survey on secure communication protocols for the Internet of Things. Ad Hoc Netw. **32**, 17–31 (2015)
10. Olivier, F., Carlos, G., Florent, N.: New security architecture for IoT network. Procedia Comput. Sci. **52**, 1028–1033 (2015)
11. Martinez-Julia, P., Skarmeta, A.F.: Empowering the Internet of Things with software defined networking. In: White Paper, IoT6 - FP7 European research project (2014)
12. Stancu, A., Halunga, S., Vulpe, A., Suciu, G., Fratu, O., Popovici, E.C.: A comparison between several software defined networking controllers. In: 12th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS 2015), Niš, Serbia, pp. 223–226, 14–17 October 2015