Leveraging SDN for Detection and Mitigation SMTP Flood Attack through Deep Learning Analysis Techniques

Mohd Zafran Abdul Aziz[1], Koji Okamura[2]

[1]Department of Advanced Information Technology, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan
Faculty of Electrical Engineering,Universiti Teknologi MARA, 40450, Shah Alam, Selangor, Malaysia
[2]Research Institute for Information Technology,
Kyushu University, Japan

Abstract
This manuscript presents a mitigation of SMTP Flood attacks on SDN-based platforms. We have revisited the SMTP security issues and SDN related works to deal with the SMTP Flood attacks. We have proposed FlowIDS as a framework that can be used to detect anomaly on SMTP traffic flows. The novelty of the FlowIDS is the detection method, whereby this work has introduced a flow based attack detection of SMTP traffic flows. Decision tree (DT) classification and deep learning (DL) algorithms were used for attack metric computations and decision making. Both algorithms were tested by simulations using SDN for DT and DL . Based on the simulation results, FlowIDS has provided significant contributions in detecting and preventing SMTP flow attacks on SDN. It also provides a quick detection and mitigation capability by reducing the network bandwidth consumption since the attack traffic flows can be dropped at the early stage of attacks.

Index Terms:
SDN, SMTP, OpenFlow, Security, ONOS, Anomaly Detection, SMTP Flood Attack, Decision Tree, Deep Learning

1. Introduction

Software-defined networking (SDN) is a platform for multi devices controlling and monitoring in vast network topologies such as IoT and cloud computing. It provides manageable network infrastructures for various computing devices and software stacks. Referring the most recently work to mitigate the SMTP Flood attacks using push back[21] method ,the mechanism in which routers upstream of the server under attack are asked to start dropping packets to the server under attack all packet will drop include legitimate SMTP packet flow. In this work, we present a framework to detect SMTP attack using FlowIDS in SDN. We have revisited the existing works [1] on SMTP security in SDN environment such as SDN, OpenFlow, Distributed Denial of Service (DDoS), botnet, spam etc. Later, we discuss the FlowIDS with simulations on SMTP Flood attacks using Suricata Network Intrusion Detection System (NIDS). The primary objective of this work is to simulate FlowIDS framework in SDN environment networks. The outcome of the simulation is to enhance anomaly detections of SMTP flood attacks over SDN. This work has implemented two algorithms as an engine for FlowIDS to detect the SMTP flood attacks, namely decision tree (DT) classification and deep learning (DL) algorithms can classify malicious traffic and normal traffic with high accurancy.In our proposed work we have used decision tree (DT) classification and deep learning (DL) algorithm to identify legitimate SMTP traffic flow which can be used to detect the SMTP flood attacks on the same malicious dataset.

We divided this work into six sections. The Introduction section provides an introduction as well the objective of this work. It follows by Related Work section that discusses SDN, and SMTP attacks. We show the proposed FlowIDS framework and detection algorithms in Section 3. After that, we show the experimental setup for the FlowIDS in section 4. Then, we discuss the experiment results in Results section which covers SMTP attack detection and prevention. Based on the simulation results, we present the comparison performance FlowIDS using DT and DL on single site. We also perform a performance comparison between FlowIDS with another precedent work, Dossy [2]. Finally, we conclude this work and propose a suggestion in the Conclusion section.

2. Related Work

1.1 Software-defined Networking (SDN)

SDN is an architecture for multi devices communication in integrated networks. Open Networking Foundation (ONF) develops OpenFlow for SDN [3]. The ONF provides SDN

Manuscript received October 5, 2017
Manuscript revised October 20, 2017

resources (e.g. switch specification) for product manufacturer and software developer to implement SDN using the OpenFlow standard and protocol [4]. Figure 1 shows SDN architecture and stacks. In SDN topology, all network nodes or devices are controlled using a control plane. The architecture splits the control plane from actual network data and routing process (data plane). The infrastructure layer communicates with SDN Controller using Control Data Plane (CDP) API (e.g. OpenFlow). All nodes or routers in the SDN network will use the CDP API for all control plane communication. The control layer consists of SDN Control Software or Controller, which extract information from the infrastructure layer such as a list of all devices in the SDN network and its states. It does not provide the entire information of all connected devices, but it provides an abstract view of the SDN network and topology. The application layer uses information from the control layer for a network abstraction administrative such as network analytics; network, system and topology managements etc. [5], [6].

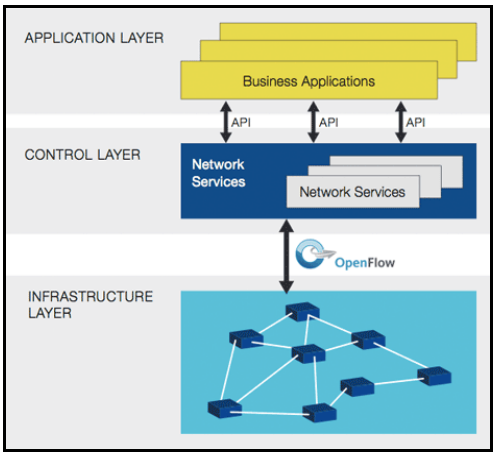

Fig. 1. SDN's stacks [7]

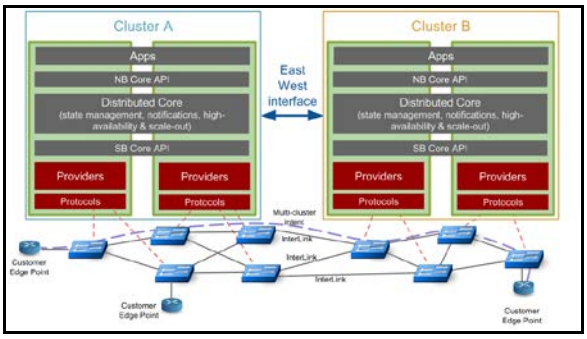

Fig. 2. ONOS architecture [6]

To implement SDN architecture and its APIs (e.g. OpenFlow), ONOS [8] is developed as an open source network OS for the SDN implementation. ONOS is a distributed SDN control platform that allows various SDN functionalities such as a global network view of network abstraction, fault tolerance, improving network performance and monitoring [9]. Figure 2 shows the ONOS architecture that provides the global network view of network infrastructure. It allows numerous network devices and systems in network clusters to share its states via ONOS. ONOS allows research, developer and vendor communities to collaborate in contributing, developing, testing as well as distributing this open source network OS. In this work, we have explored ONOS as a platform for FlowIDS implementation.

1.2 Simple Mail Transfer Protocol (SMTP)

SMTP can be implemented in a centralized network security by the SDN architecture [10], [11]. The SDN architecture allows an abstraction of network security monitoring and control in providing a central authority for clustered networks. This allows various security parameters such as firewall, IDS, antivirus and malware tools to be integrated by SDN control planes. The following paragraphs will discuss some related works on SMTP security threats and countermeasures.

N. Hoque et al. (2014) [12] discuss tools used by attackers and security admin in SDN. The authors revisit machine learning algorithm, flow-based features for botnet detection using a predefined dataset. The dataset consists of SMPT Spam and UDP Storm and it successfully detected with rate 75%. S. Lim et al. (2014) [13] propose to utilize SDN for DDoS attack detection and prevention. The authors discuss a method to block the DDoS attack using OpenFlow in SDN controller. T. Sochor (2014) [14] revisited the existing methods to detect and prevent spam messages. H. Chen et al. (2015) [9] integrate entropy measurement for flooding detections in mail systems. It studies an entropy in round-trip time (RTT) and retransmission timeout (RTO) to detect dangerous traffics. Y. Yan et al. (2015) [19] review DDoS attack on cloud computing and then how to prevent the DDoS attack by implementing SDN in the cloud computing. P. Holl (2015) [20] discusses multiple methods to detect and prevent DDoS attack in SDN such proactive and reactive defenses, and post-attack analysis. Q. Yan et al. (2016) [22] present a survey on SDN, DDoS in cloud computing. T. Bakhshi [15] (2017) has reviewed the SDN paradigm which started at the history of SDN, and later discussed the SDN platforms as well as the challenges to secure SDN platforms. S. Fernandes [16] (2017) has explored the performance of networking protocols using modeling and analysis techniques. The purpose of their work is to collect various networking protocol including SMTP for Internet traffic profiling.

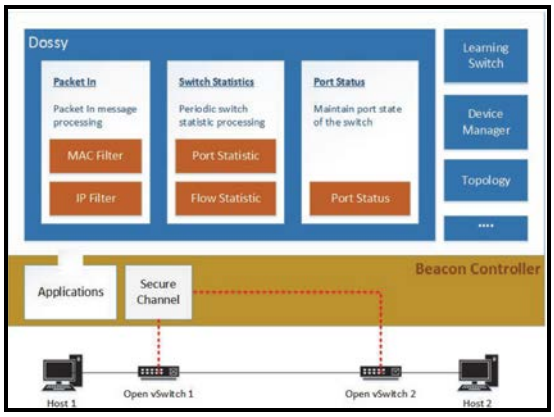

Fig. 3. Dossy framework for mitigating DoS attacks

The nearest preceding work to this manuscript was done by Y.E. Oktian et al. [2] (2014). The authors had proposed Dossy framework for mitigation against DoS attacks in SDN domain. Referring to Figure 3, Dossy relies on six network parameter for DoS detection such as binding, location tracker, packets filtering, port and flow, statistic queries, and port status. The main parameters such as packet in, switch statistics, and port status were collected using OpenFlow's API which is to provide data for mitigation against the DoS attacks. All these parameters were processed by Dossy for ensuring the quality of service (QoS) of SDN networks are preserved against the DoS or DDoS attacks. Dossy relied on flow-based analysis and self-organizing maps (SOM) algorithm as the core processing engine for detection and mitigation the DoS attacks. For those interested on the review of SMTP and SDN security, one may refer to our previous work [1] which has revisited various other works on the related issues.

3. FlowIDS

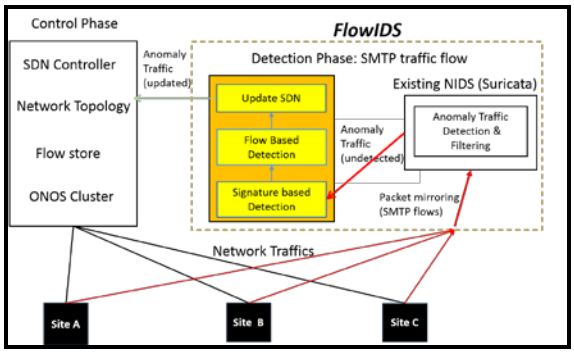

Fig. 4. FlowIDS framework [17]

FlowIDS is a framework for anomaly detection on SMTP traffic flows. The novelty of the FlowIDS is the detection method, whereby it uses decision tree clasification to detect attack flows. It can be integrated with the existing network security systems such as firewall, IDS, SDN controller and ONOS application. In this work we have chosen Suricata IDS because it has open APIs (open source) that can be used for interoperability between ONOS and other SDN platforms for an abstraction network control and monitoring. Figure 1 shows the FlowIDS framework. In the figure, FlowIDS collects all undetected anomaly traffic flows by the NIDS (e.g. Suricata). The first stage is to check the SMTP traffic flows against the existing flow based signature for known SMTP traffic flow attack. If the known attack is mounted, it will update SDN (e.g. ONOS) to drop the SMTP traffic flows. For the second stage, a flow-based detection is used to detect unknown anomaly for SMTP traffic flows. To improve for a real-time detection, FlowIDS will distribute the second stage computations into multiple distributed computing systems. This will improve the real-time detection of new attack flows. It also provides load balancing for processing huge SMTP traffic flows. If the second stage has detected flow attacks, it will update SDN controller to drop the SMTP traffic flows (bad flows) and also updates the flow based signature (first stage) for a future signature attack detection. If the SMTP traffic flows passed the second stage, it will update SDN controller for legitimate SMTP traffic flows.

FlowIDS relies on either decision tree (DT) classification or deep learning (DL) algorithm for legitimate smtp flow detections. Figure 5 shows the example of classification DT for SMTP legitimate flow value. DL is used for learning the SMTP legitimate flow value by using a nonlinear processing or condition. The decision tree (DT) classification and deep learning (DL) algorithm to identify legitimate SMTP traffic flow which can be used to detect the SMTP flood attacks on the same malicious dataset. The proposed method has reduced the network utilization bandwidth during the attacks. Figure 6 shows the training and validation processes of the DL by FlowIDS. In this work, we have tested FlowIDS with DT and DL in SDN environment.

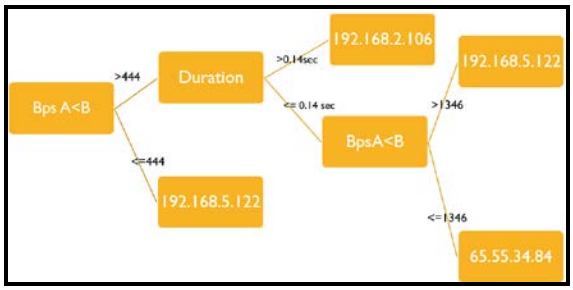

Fig. 5. An example of classification DT for SMTP legitimate flow

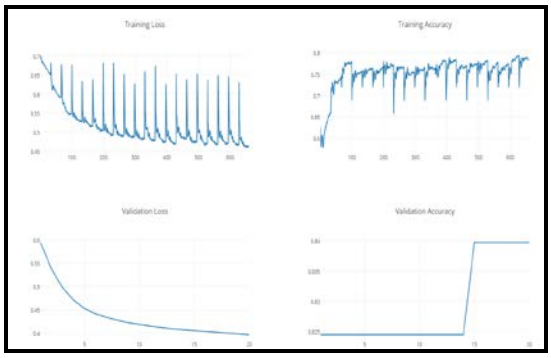

Fig. 6. DL training and validation processes

4. Simulation Setup

These simulations have used dataset internet traffic from Internet traffic dataset University Brunswick Canada [18] and botnet dataset from Malware Capture Facility Project [19]. Another work done by G. Carter [11], the author has used the same dataset for his research on mitigation SMTP flood attack. However, his work had focused on server time out as a method to detect the SMTP flood attack.

The entire experiments were executed on cloud computers by 8 Core Xeon CPU, 16 GB RAM, 80 GB storage, and gigabit network adapters. The SDN configuration is used for simulation the SMTP flood attack in standalone SDN controller. Figure 7 shows the simulation for SMTP flow attacks that originated from nodes h2 and h3 which are targeting the smtpserver as a single site attack. The network performance in this simulation is measured between node h12 (legitimate user) and smtpserver. Figure 5 shows the process flow of detection the SMTP flow attack. The simulation is divided into four subcases as follows:

1. No SMTP flow attack.
2. SMTP flow attacks at time 10 to 30 seconds. There is no IDS to detect the SMTP flow attacks.
3. SMTP flow attacks at time 10 to 30 seconds. NIDS (Suricata) is used to detect the SMTP flow attacks.
4. SMTP flow attacks at time 10 to 30 seconds. NIDS (FlowIDS + Suricata) (DT) is used to detect the SMTP flow attacks.
5. SMTP flow attacks at time 10 to 30 seconds. NIDS (FlowIDS + Suricata) (DL) is used to detect the SMTP flow attacks.

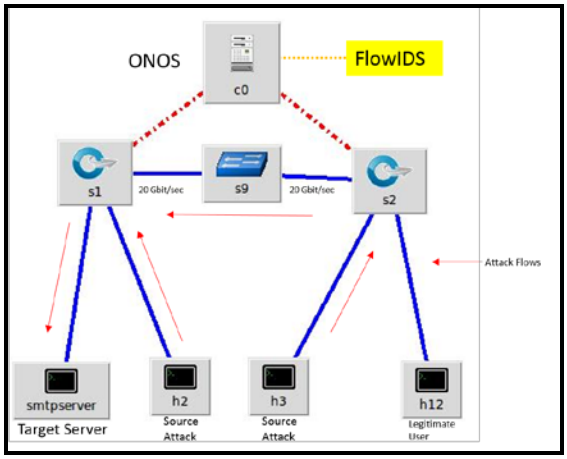

Fig. 7. Experiment setup for SMTP single site attack

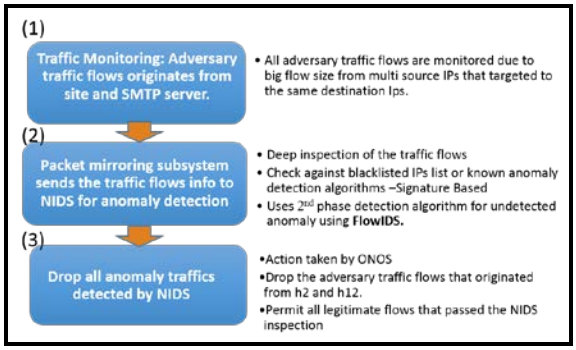

Fig. 8. A summary of FlowIDS experiment on SDN

5. Results

This section presents the results of SMTP attack as shown in Figure 7. For the subcase 1 (no attack), the network bandwidth between node h12 and smtpserver was steady at 7.0 GBits/sec whereby there is no SMTP flow attack. In the subcase 2 (attack without IDS), the network bandwidth has almost grounded close to 0 GBits/sec at second 12 when the SMTP flow attacks are mounted. This is expected to happen because the subcase 2 does not have IDS in the simulation setup. Referring to the subcase 3

(attack with NIDS), the network bandwidth has dropped to 5.8 GBits/sec at second 10 and it was flatted at seconds 16 until 24. NIDS (Suricata) begins to detect the SMTP flow attacks at second 25 and the SMTP flow attacks are rapidly dropped from the network at seconds 25 until 29. For the subcase 4 (attack with FlowIDS(DT)), the network bandwidth has dropped to 5.8 GBits/sec at seconds 10. The bandwidth rapidly drifts down until second 15 whereby the network bandwidth is steady at 2 GBits/sec. This trend remains steady until second 24. The network begins to recover at second 25 because FlowIDS has successfully identified the attack flows and then updates the SDN controller with the latest attack signatures. For the subcase 5 (attack with FlowIDS(DL)), the network bandwidth has dropped to 5.8 GBits/sec at seconds 10. The bandwidth rapidly drifts down until second 15 whereby the network bandwidth is steady at 4 GBits/sec. This trend remains steady until second 24. The network begins to recover at second 21 because FlowIDS has successfully identified the attack flows and then updates the SDN controller with the latest attack signatures. Based on simulation result on using deep learning algorithm, the bandwidth utilization is near to 29% whereby the decision three algorithm has used up to 68 % for single site mitigation during the attacks.

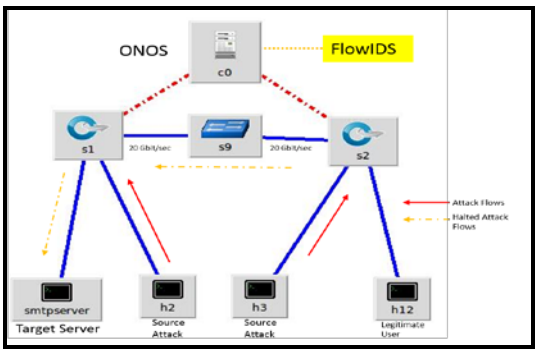

Fig. 9. SMTP flow attacks are dropped at Switch 1 and 2 using FlowIDS in SDN

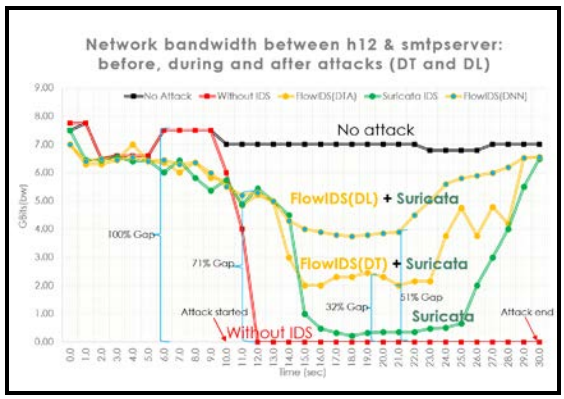

Fig. 10. Comparison of FlowIDS with DT and DL algorithms

6. Discussion

1.1 Comparison DT and DL algorithms

Figure 10 shows the comparison of FlowIDS performance using DT and DL algorithms in a single SDN domain simulation. Referring to Figure 6, the simulation was implemented using FlowIDS (DT) as the core engine for SMTP flow attack detections which are referred to the author's previous work [17]. The figure 10 has shown that DL algorithm provides a better network bandwidth handling compare to DT algorithm. FlowIDS (DL) has saved the network bandwidth at 4 GBits/sec at second 15 and it was steady until second 21. Comparing to FlowIDS (DT), it has settled the network bandwidth at 2 GBits/sec by the same duration. Through the DL algorithm, FlowIDS has preserved 2 GBits network bandwidth during the attacks. It also has shown a significant network recovery at second 22 whereby DT has recovered 3 seconds late compared to DL.

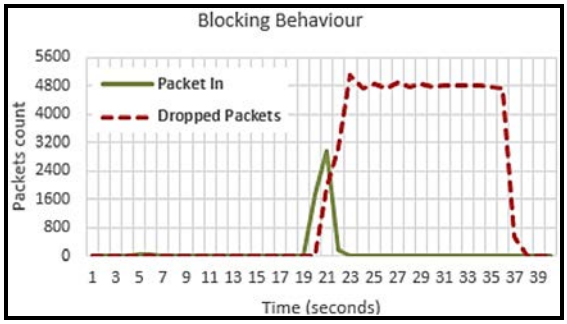

Fig. 11. Dossy packet blocking (drop) behavior [2]

1.2 Comparison with the nearest work

This subsection will discuss the performance of FlowIDS (DL) with Y.E. Oktian et al. [2]. Figure 11 shows the performance of Dossy in blocking DoS packets. The network flooded by 3000 packets at second 20.5 as the peak of attacks. Dossy began to detect the DoS packets at second 21 and almost all DoS packets were dropped at second 23 until the end of attacks. Figure 12 the performance of FlowIDS in blocking DoS packets. The network flooded by 2500 packets at second 10.5 as the peak of attacks. FlowIDS began to detect SMTP flow attacks at second 13 and almost all attack packets were dropped at second 19 until the end of attacks.

Comparing Dossy, FlowIDS has detected the attacks earlier by 10 seconds than Dossy, and the highest of attack packet count is lesser than Dossy by 500 packets. By this comparison, we have shown that FlowIDS has significantly

improved the QoS of SDN. Based on previous work result using dossy packet blocking system [2], all packet were dropped until 0 % during the attack. Comparing to our work, the packet has dropped only up to 33 % which prevent the network bandwidth cripple to 0% during the attack

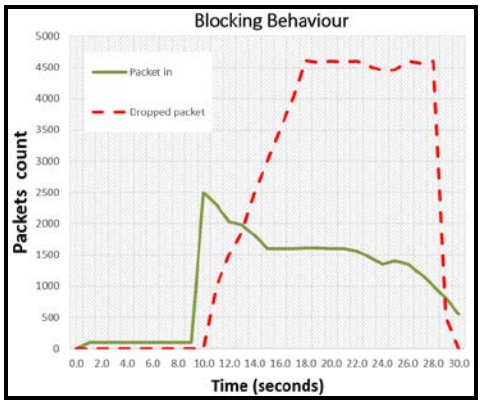

Fig. 12. FlowIDS (DL) packet blocking (drop) behavior

7. Conclusion

We have presented simulations of FlowIDS framework using DT and DL algorithms for anomaly detection on SMTP traffic flows. DL has shown a significant efficiency in detection of SMTP attack flows compared to DT classification value. The simulations of the FlowIDS framework were conducted for DT and DL in a single site. In previous work for multi-sites [20], the proposed method allows the FlowIDS to update SDN controllers with the latest SMTP spam signatures. It will prevent any known of SMTP spam email from entering others SDN domains or sites, this method allows to collaborate and mitigate the SMTP flood attack on SMTP server close to the source of attacks in other site network topology (early mitigation). By the mixture of FlowIDS and Suricata NIDS, both systems have offered better SMTP flow attack detection and prevention compared to the standalone Suricata NIDS as the main security parameter.

Lastly, we have shown the performance comparison of FlowIDS and Dossy. FlowsIDS has shown improvement on attack mitigations in term of earlier detection, bandwidth consumption, network recovery time and legitimate SMTP flow traffic still in process during the attacks. The critical requirement in this research work is to reduce the network bandwidth consumption in single the SMTP flow attacks being mounted. It can be done if the SMTP flow attacks are early detected at the source of attack sites. Then the mitigation can be deployed faster before the SMTP flow attacks are spreading to other sides.For future work, we will simulate FlowIDS with DL algorithm in multi-sites. We also plan to implement FlowIDS by experimental testbed after all simulation works are done.

Acknowledgment

The authors with to thank UNSW for support the work and Kementerian Pengajian Tinggi, Malaysia and University Teknologi MARA for the PhD scholarship.

References

[1] M. Z. A. Aziz and K. Okamura, “A Security Trending Review on Software Define Network (SDN),” Journal of Advanced Research in Computing and Applications, vol. 6, no. 1, pp. 1–16, 2016.

[2] Y. E. Oktian, S. Lee, and H. Lee, “Mitigating Denial of Service (DoS) Attacks in OpenFlow Networks,” 2014 International Conference on Information and Communication Technology Convergence (ICTC), pp. 325–330, 2014.

[3] Open Networking Foundation, “Software-Defined Networking: The New Norm for Networks,” 2012.

[4] O. N. Foundation, “OpenFlow,” 2016. [Online]. Available: https://www.opennetworking.org/sdn-resources/openflow/57-sdn-resources/onf-specifications/openflow?layout=blog. [Accessed: 29-Jan-2016].

[5] S. H. Park, B. Lee, J. You, J. Shin, T. Kim, and S. Yang, “RAON: Recursive abstraction of OpenFlow networks,” Proceedings - 2014 3rd European Workshop on Software-Defined Networks, EWSDN 2014, pp. 115–116, 2014.

[6] V. K. Gurbani, M. Scharf, T. V. Lakshman, V. Hilt, and E. Marocco, “Abstracting network state in Software Defined Networks (SDN) for rendezvous services,” IEEE International Conference on Communications, pp. 6627–6632, 2012.

[7] SDxCentral, “Inside SDN Architecture,” 2016. [Online]. Available: https://www.sdxcentral.com/resources/sdn/inside-sdn-architecture/. [Accessed: 31-Jan-2016].

[8] ONOS, “ONOS,” 2017. [Online]. Available: http://onosproject.org/. [Accessed: 05-Feb-2017].

[9] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, and B. Lantz, “ONOS: Towards an Open, Distributed SDN OS,” in Proceedings of the third workshop on Hot topics in software defined networking - HotSDN ’14, 2014, pp. 1–6.

[10] S. Scott-Hayward, G. O’Callaghan, and S. Sezer, “SDN Security: A Survey,” 2013 IEEE SDN for Future Networks and Services (SDN4FNS), pp. 1–7, 2013.

[11] R. Kl and P. Smith, “OpenFlow: A Security Analysis,” in 21st IEEE International Conference on Network Protocols (ICNP), 2013.

[12] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, “Network attacks :

Taxonomy , tools and systems," Journal of Network and Computer Applications, vol. 40, pp. 307–324, 2014.
[13] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks," 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 63–68, 2014.
[14] T. Sochor, "Overview of e-mail SPAM Elimination and its Efficiency," Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on, pp. 1–11, 2014.
[15] T. Bakhshi, "State of the art and recent research advances in software defined networking," Wireless Communications and Mobile Computing, vol. 2017, 2017.
[16] S. Fernandes, "Internet Traffic Profiling," in Performance Evaluation for Network Services, Systems and Protocols, Cham: Springer International Publishing, 2017, pp. 113–152.
[17] M. Z. A. Aziz and K. Okamura, "A Method to Detect SMTP Flood Attacks using FlowIDS Framework," International Journal of Computer Science and Network Security, vol. 17, no. 6, pp. 1–8, 2017.
[18] "Dataset internet traffic from University New Brunswick (UNB) Canada." [Online]. Available: http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html.
[19] S. García, "Malware Capture Facility Project. CVUT University. Dataset CTU-Malware-Capture-Botnet-1," 2013. [Online]. Available: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-1/%0A. [Accessed: 03-Feb-2013].
[20] M. Zafran, A. Aziz, G. Jourjon, S. Jha, and K. Okamura, "A Collaborative Mitigation SMTP flood Attack using SDN platform on Multi Site," International Journal of Computer Science and Network Security, Vol 15 no.7, pp. 1–9, 2017.
[21] M. Still and E. McCreath, "DDoS protections for SMTP servers," Journal of Computer Science and Security ,no. 4, pp. 537–550, 2011

Mohd Zafran Abdul Aziz has received his first Bachelor Degree (B. Eng of Electrical and Computer Science) from Kumamoto University, Japan on March 01 and obtained his Master Degree (MSc of Engineering) from Tokyo University Of Technology,Japan on March 2008. He also has 6 years in industrial as project engineer in several multinational company focus on industrial automation and instrument engineer. Currently on study leave as lecturer from Computer Department of Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia. He is currently a PhD candidate and belong to Department of Advanced Information Technology, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan.

Koji Okamura is a Professor at Research Institute for Information Technology, Kyushu University and Director of Cybersecurity Centre Kyushu University, Japan. He received B.S and M.S. Degree in Computer Science and Communication Engineering and Ph.D. in Graduate School of Information Science and Electrical Engineering from Kyushu University, Japan in 1988, 1990 and 1998, respectively. He has been a researcher of MITSUBISHI Electronics Corporation Japan for several years and has been a Research Associate at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan and Computer Centre, Kobe University, Japan. He's area of interest is Future Internet and Next Generation Internet, Multimedia Communication and Processing, Multicast/IPV6/QoS , Human Communication over Internet and Active Network. He is a member of WIDE, ITRC, GENKAI, HIJK project and key person of Core University Program on Next Generation Internet between Korea and Japan sponsored by JSPS/KOSEF.