A Method to Detect SMTP Flood Attacks using FlowIDS Framework

Mohd Zafran Abdul Aziz[†,††] and Koji Okamura[†††]

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

Summary
This publication presents a framework to detect SMTP Flood attacks on SDN-based platforms such as ONOS. We have revisited the SMTP security issues as well as the proposed solutions to overcome or mitigate the SMTP Flood attacks. ONOS offers network abstraction management as well as a centralized security solution for the SMTP attack detection and prevention. Due to robustness and flexibility of the ONOS, we have proposed FlowIDS as a subsystem 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. It can be integrated with the existing network security systems such as firewall, IDS, SDN controller and ONOS applications. The experiment results have shown that the proposed FlowIDS has provided a significance contribution in detecting and preventing SMTP flow attacks on SDN domains. It also provides a quick detection and mitigation on SMTP server by reducing the bandwidth consumption because of the attack traffic flows can be dropped at the early stage of attacks.

Key words:
SDN, SMTP, Spam, OpenFlow, Security, ONOS, Anomaly Detection , SMTP Flood Attack

1. Introduction

SDN is an architecture for multi devices communication in integrated networks. It provides manageable network infrastructures that consist millions of computing devices and software. In this work, we present a framework to detect SMTP attacks on ONOS distributed systems. We revisit the existing works on SMTP security such as ONOS, SDN, OpenFlow, D/DoS, botnet, spam etc. Later, we discuss the FlowIDS with an experiment on SMTP Flood attack using ONOS platform and Suricata NIDS. The primary objective of this work is to develop FlowIDS framework between ONOS distributed controller and network IDS. This also cover on a method to enhance anomaly detections of SMTP flood attacks.

We divided this work into six sections. The Introduction section provides an introduction as well the objective of this work. It follows by Related Works section that discusses ONOS, SDN, and SMTP attacks. We also show detection and prevention methods by a comparison table. We show the proposed FlowIDS framework, algorithm and design with integration on ONOS and NIDS in Section 3. After that, we show the experiment setup for the FlowIDS in section 4. Then, we discuss the experiment results in Results and Discussion section. Finally, we conclude this work and propose a suggestion in the Conclusion section.

2. Related Works

2.1 Software-defined Networking (SDN)

SDN is an architecture for multi devices communication in integrated networks. Open Networking Foundation (ONF) develops OpenFlow for SDN [1]. The ONF provides SDN resources (e.g. switch specification) for product manufacturer and software developer to implement SDN using the OpenFlow standard and protocol [2]. Figure 1 show a general 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. [3], [4].

Manuscript received June 5, 2017
Manuscript revised June 20, 2017

2.2 Open Network Operating System (ONOS)

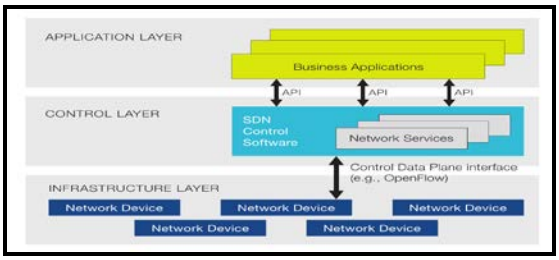

Figure 1. SDN's stacks [7]

To implement SDN architecture and its API (e.g. OpenFlow), ONOS [5] 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 [6].

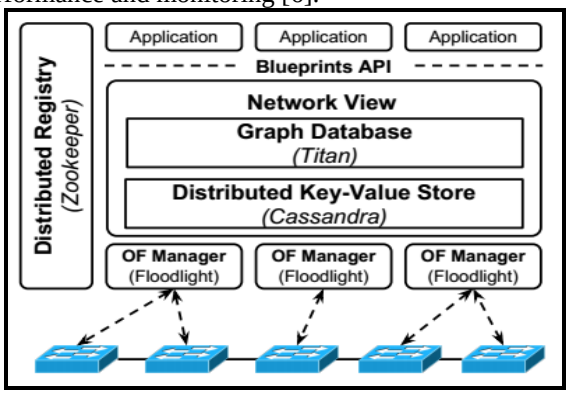

Figure 2. ONOS architecture [6]

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 explore ONOS as a potential platform for FlowIDS implementation.

2.3 Security Issues on Simple Mail Transfer Protocol (SMTP)

Distributed systems such as cloud computing and Internet of Things (IoT) are not the main factors for organizations to migrate theirs network infrastructure into SDN. Another reason for the migration is a centralized network security can be directly done by the SDN architecture [8], [9]. The SDN architecture allows an abstraction of network security monitoring and control in providing a central authority for clustered networks, which previously hard to be done by traditional distributed networking systems and infrastructures [1], [2]. This allows various security parameters such as firewall, IDS, antivirus and malware tools to be integrated by SDN control planes. To realize the SDN-based security, ONOS is the right choice for the integrated network security prototype development. Due to this concern, we choose ONOS for FlowIDS implementation. The following paragraphs will discuss some related works on SMTP security threats and countermeasures.

N. Hoque et al. (2014) [10] 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) [11] 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. It was simulated in POX controller using Mininet emulator. C. Schafer [12] (2014) uses geolocation and country to detect an anomaly that can be used to identify spam email. A novel contribution, Theoretical Geographical Travelling Speed (TGTS) method is proposed in his work. T. Sochor (2014) [13] revisited the existing methods to detect and prevent spam messages.

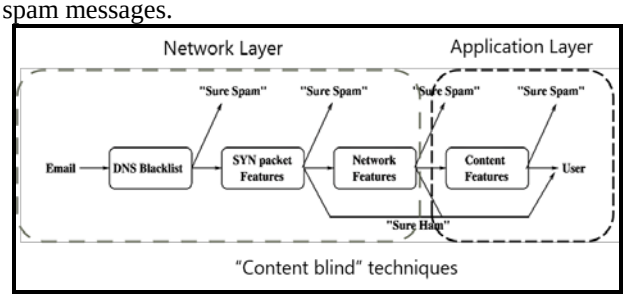

Figure 3. A decision tree of content blind technique [16]

Multi-layer protection technique such as blacklisting and greylisting was discussed. E. B. Beigi et al. (2014) [14] reexamined flow-based for botnet detection, which also studies its effectiveness in detection using a predefined dataset. T. Ouyang [15] et al. (2014) study spam filtering pipeline for finding its accuracy and tradeoff in four layers. The authors used three decision trees: packet features, flow features and the combination of both features. Figure 3 show example of decision tree for spam email detections.

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. The entropy can help to improve malicious mail analysis and detection for protocols: SMTP, IMAP4, POP3 and HTTPS.

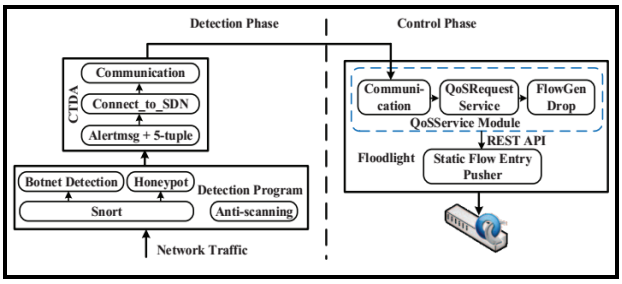

Figure 4. CTDA architecture for anomaly detection

Table 1. SMTP attack detection methods by IDS or firewall [23]

	Anomaly-based	**Stateful protocol analysis**	**Signature-based**
Pros	Effective to detect new and unforeseen vulnerabilities. Less dependent on OS. Facilitate detections of privilege abuse.	Know and trace the protocol states. Distinguish unexpected sequences of commands.	Simplest and effective method to detect known attacks. Detail contextual analysis.
Cons	Weak profiles accuracy due to observed events being constantly changed. Unavailable during rebuilding of behavior profiles. Difficult to trigger alerts in right time.	Resource consuming to protocol state tracking and examination. Unable to inspect attacks looking like benign protocol behaviors. Might incompatible to dedicated OSs or APs.	Ineffective to detect unknown attacks, evasion attacks, and variants of known attacks. A Little understanding of states and protocols. Hard to keep signatures/patterns up to date. Time-consuming to maintain the knowledge

R. Sahay et al. (2015) [17] propose an implementation of a distributed collaboration framework for sharing information that can be used to mitigate DDoS in SDN. J. Jeong et al. (2015) [18] propose security services in SDN for a centralized firewall and DDoS mitigation systems. Y. Yan et al. (2015) [19] review DDoS attacks on cloud computing and then how to prevent the DDoS attacks by implementing SDN in the cloud computing. P. Holl (2015) [20] discusses multiple methods to detect and prevent DDoS attacks in SDN such proactive and reactive defenses, and post-attack analysis. Huang et al. (2015) [21] propose IDS for cloud computing systems using SDN architecture. The authors have proposed system can detect and block many botnet or malware using Cooperative Threat Defending Algorithm (CTDA) as shown in Figure 4.

Q. Yan et al. (2016) [22] present a survey on SDN, DDoS in cloud computing. A survey on collaborative attack mitigation and response [6] show 50 % of respondents disclose having a cooperation with ISP to assist them in mitigating and responding to a security event/incident. To deploy multi-domain controller, distributed ONOS was introduced to monitor network between multi-domain [7], and SnortFlow [8] is used to communicate with SDN controller. The following Table 1 shows the summary of SMTP attack detection methods by IDS or firewall.

3. FlowIDS

FlowIDS is a framework for anomaly detection on SMTP traffic flows. The novelty of the FlowIDS is the detection method, whereby this work has introduced flow based attack detection on the SMTP traffic flows. It can be integrated with the existing network security systems such as firewall, IDS, SDN controller and ONOS application. The following subsections will present the framework, anomaly detection method, anomaly detection algorithm, and anomaly detection performance evaluations.

3.1 Framework

Figure 5 shows the FlowIDS framework that is integrated with NIDS, namely Suricata.

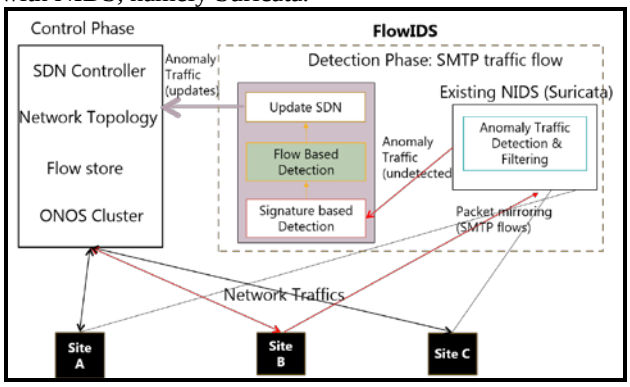

Figure 5. An overview of FlowIDS framework

In this work we have chosen the Suricata because it has open API (open source) that can be used for interoperability between ONOS and other SDN platforms for an abstraction network control and monitoring.

Figure 6 shows the process flows to detect SMTP attack using FlowIDS. 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 attacks. If known attacks are mounted, it will update SDN (e.g.

ONOS) to drop the SMTP traffic flows. For the stage two, 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 stage two work into multiple distributed computing systems. This will reduce computing processing and loading if the FlowIDS is run on the same machine (or virtual machine) with the NIDS. It also provides load balancing for processing huge SMTP traffic flows. If the stage 2 has detected an attack, it will update SDN to drop the SMTP traffic flows and also update the flow based signature (stage 1) for a future signature attack detection. If the SMTP traffic flows passed the stage 2, it will update SDN for legitimate SMTP traffic flows.

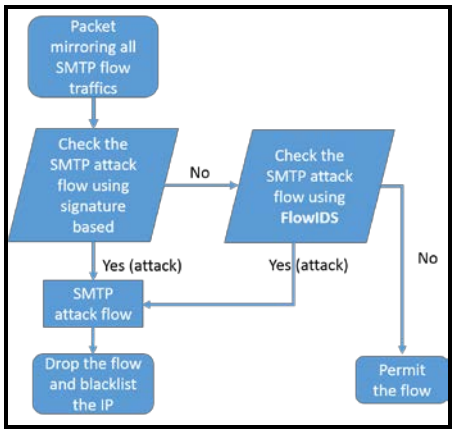

Figure 6. A process to detect SMTP attack using FlowIDS

3.2 Anomaly Detection Method

FlowIDS uses decision tree algorithm as the anomaly detection method. Decision tree algorithm defines the minimum and maximum value of SMTP traffic flow rates using the real Internet traffic dataset for legitimate or attack flow. The following labels define the usages:

Flow_legit: a legitimate of SMTP traffic flows.

Flow_rate_min: a minimal flow of the legitimate SMTP traffic flows.

Flow_rate_max: a maximum flow of the legitimate SMTP traffic flows.

Duration_min: a lifetime (runtime) of the *Flow_rate_min.*

Duration_max: a lifetime (runtime) of the *Flow_rate_max.*

Flow_rate: traffic flows between SMTP server and client during online.

The following labels define the traffic flow detection evaluations:

High Rate Attack (HRA) flow: An ingress flow with higher traffic rate within the upper limit of the *flow_legit* but its duration less than the *flow_legit* duration (*duration_min).*

Short Rate Attack (SRA) flow: An ingress flow with higher traffic rate within the upper limit of the *flow_legit* but its duration greater than the *flow_legit* duration (*duration_max*).

Long High Rate (LHR) attack flow: An ingress flow having a traffic rate and traffic duration that exceeding the upper limit of the *flow_legit* duration (*duration_max* and *flow_rate_max*).

Idle User (ISA) flow: An ingress flow with traffic rate within the legitimate range of the *flow_legit* but its duration greater than the upper limit of the *flow_legit* duration.

Long Low Rate (LLR) attack flow: An ingress flow with lower traffic rate that is lower than the lower limit of the *flow_legit* but its duration is greater than the upper limit of the *flow_legit* duration (*duration_max* and *flow_rate_min*).

```
FlowIDS
for each new incoming flow at switch
    Do
        newUserList{} = source IP/MAC of the new flow
        Install OpenFlow rules for forwarding a new IP/MAC in switch flow table
        Monitor behavior, rate and duration of traffic flows using a NIDS (Suricata)
        If identify alert msg suricata then {NIDS detect}
                Mitigate: drop flow (source_address)
        else if flow_rate >= flow_rate_max and flow duration <= duration_min of flow_legit then {HRA}
                Mitigate: drop flow (source_address)
        else if flow_rate >= flow_rate_max and flow duration <= duration_max of flow_legit then {SRA}
                Mitigate: drop flow (source_address)
        else if flow_rate >= flow_rate_max and flow duration >= duration_max of flow_legit then {LHR}
                Mitigate: drop flow (source_address)
        else if flow_rate_min < flow_rate < flow_rate_max and flow duration >= duration_max of flow_legit then {ISA}
                Mitigate: drop flow (source_address)
        else if flow_rate <= flow_rate_min and flow duration >= duration_max of flow_legit then {LLR}
                Mitigate: drop flow (source_address)
        else
                Legitimate flow: let the flow pass
    end if
end for
```

Figure 7. FlowIDS algorithm

3.3 Anomaly Detection Algorithm

FlowIDS uses decision tree algorithm to detect SMTP flow attacks on network traffics. Figure 7 shows the implemented FlowIDS in network switch for detection and prevention of the SMTP flow or spam email attacks. The FlowIDS relies on HRA, SRA, LHR, ISA, LLR, Flow_legit, duration_min, duration_max etc as general rules (or if-else conditions) for the decision tree algorithm to perform anomaly detection and network flow dropping. Any flow that passed the general rules is assumed as legitimate SMTP traffic flows.

3.4 Anomaly Detection Performance Evaluations

The performance metric is used to evaluate the performance of FlowIDS in detecting an anomaly in SMTP traffic flows. The performance metric is commonly employed for data analysis in pattern recognition and

information retrieval fields. This work has selected the regular approach of performance metric evaluations such as a true positive (TP), true negative (TN), false positive (FP) and false negative (FN). These metrics are used as binary classification metrics for measuring system performance. These metrics suitable for a botnet attack identification because of both fall on a binary classification problem. These classifications are defined as follows:

TP: a number of legal packets that are identified correctly whereby it allows the legal packets to reach the destination IP.

TN: a number of attack packets that are dropped in the network whereby it prevents the attack packets from reaching the destination IP.

FN: a number of legal packets that are falsely discarded whereby it prevents the legal packets from reaching the destination IP.

FP: a number of attack packets that are falsely forwarded whereby it allows the attack packets to reach the destination IP.

The aforementioned metrics are used to calculate the performance measurements as follows:

Precision (PN): computes the percentage of forwarded legal packets.

$$PN = \frac{\text{TP}}{TP + FP}$$

Recal (RL): computes the percentage of forwarded legal packets to the destination IP.

$$RL = \frac{\text{TP}}{TP + FN}$$

True Negative Rate (TNR): computes the percentage of dropped attacks packets.

$$TNR = \frac{\text{TN}}{TN + FP}$$

Negative Predictive Value (NPV): computes the percentage of dropped attacks packets that actually attack packets.

$$TNR = \frac{\text{TN}}{TN + FN}$$

F-measure (FM): evaluates the system effectiveness (success) to forwarded legal packets during runtime. This metric combines the percentage of PN and RL.

$$FM = \frac{2 * \text{PN} * \text{RL}}{PN * RL}$$

4. Experiment Setup

Figure 8 shows the experiment setup for SMTP flow attacks that originated from nodes h3 and h12 to the target smtpserver. The experiment setup is divided into four subcases as follows:

1. No SMTP flow attack.
2. SMTP flow attacks at time 10 to 30 seconds. There are 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) is used to detect the SMTP flow attacks.

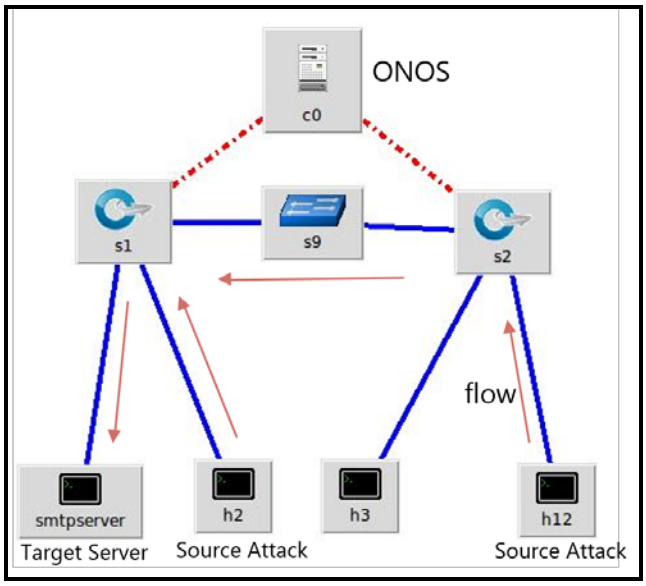

Figure 8. Experiment setup for SMTP attacks

Figure 9 shows the summary of the SMTP flow attacks detection and prevention using FlowIDS. This experiment setup has used dataset internet traffic [24] from Internet traffic dataset University Brunswick Canada (refer to Figure 10) and botnet dataset from Malware Capture Facility Project [25]. Another work done by G. Carter [11], the author have used the same dataset for his research on mitigation SMTP flood attack. However this author had focused on server time out as a method to detect the SMTP flood attack. The entire experiment was executed on cloud computers by 8 Core Xeon CPU, 16 GB RAM, 80 GB storage, and gigabit network adapters.

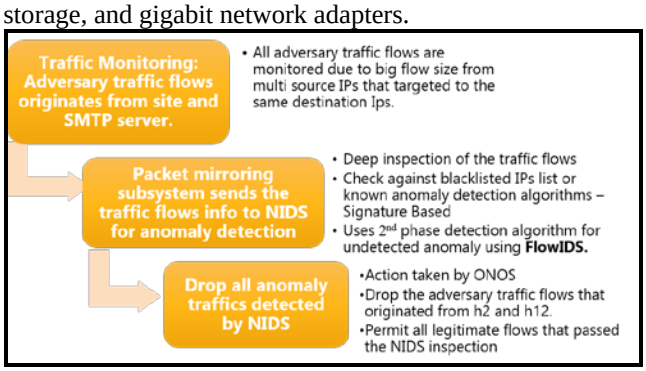

Figure 9. A summary of FlowIDS experiment (subcases 4)

Day	Date	Description	Size (GB)
Saturday	12/6/2010	Normal Activity. No malicious activity	4.22
Sunday	13/6/2010	Infiltrating the network from inside + Normal Activity	3.95
Monday	14/6/2010	HTTP Denial of Service + Normal Activity	6.85
Tuesday	15/6/2010	Distributed Denial of Service using an IRC Botnet	23.4
Wednesday	16/6/2010	Normal Activity. No malicious activity	17.6

Figure 10. A snapshot of parameter on dataset Internet traffic, ISCX University New Brunswick (UNB) Canada [24]

5. Results & Discussion

Figure 11 shows the decision tree used by FlowIDS to detect an anomaly in SMTP traffic flows using datasets [24], [25]. Any SMTP traffic flow does not comply with the legitimate flow as shown in the Figure 11 will be dropped. The performance metrics as shown in Figure 12 are used to evaluate the performance of FlowIDS and Suricata in detecting an anomaly in SMTP traffic flows.

The F-measure (FM) plays an important role to evaluate the system effectiveness (success) to forwarded legal packets during the experiment. For example, if SMTP traffic flows and duration from h12 (192.168.2.106) to smtpserver (192.168.5.122) are greater than 1346 flows and less or equal to 0.14 second, it passed the given conditions and the SMTP traffic flows are allowed to reach its destination (smtpserver).

Figure 13 shows the experiment results for the four subcases as aforementioned in Experiment Setup section.

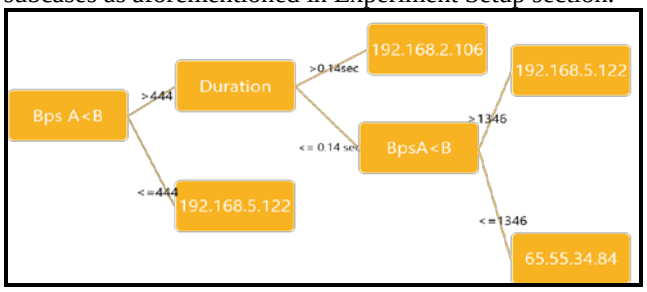

Figure 11. The decision tree of legitimate flow

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.999	0.045	0.999	0.999	0.999	0.954	0.978	0.999	192.168.5.122
0.846	0.007	0.611	0.846	0.710	0.715	0.919	0.481	192.168.2.106
0.000	0.001	0.000	0.000	0.000	-0.001	0.987	0.083	192.168.3.115
0.000	0.001	0.000	0.000	0.000	-0.001	0.499	0.001	65.55.34.84
0.000	0.001	0.000	0.000	0.000	-0.001	0.499	0.001	205.188.105.145
0.000	0.001	0.000	0.000	0.000	-0.001	0.499	0.001	64.12.78.142
0.000	0.000	0.000	0.000	0.000	0.000	0.493	0.001	209.85.212.54
0.000	0.001	0.000	0.000	0.000	-0.001	0.984	0.054	192.168.2.109
0.000	0.000	0.000	0.000	0.000	0.000	0.060	0.001	64.12.206.39
0.000	0.000	0.000	0.000	0.000	0.000	0.984	0.043	192.168.4.120
0.000	0.000	0.000	0.000	0.000	0.000	0.992	0.529	192.168.3.117
0.000	0.000	0.000	0.000	0.000	0.000	0.493	0.001	192.168.2.107
0.000	0.000	0.000	0.000	0.000	0.000	0.491	0.001	192.168.2.113
0.988	0.045	0.985	0.988	0.986	0.942	0.975	0.984	

Figure 12. The accuracy of flow-legit for normal traffic

The performances of the four subcases were evaluated based on network bandwidth consumptions during the SMTP flow attacks mounted at time 10 to 30 seconds. For the subcase 1, the network bandwidth between h12 and smtpserver is steady around 7.0 GBits/sec when there is no SMTP flow attack. For the subcase 2, the network bandwidth has almost fallen to ground that close to 0 GBits/sec when the SMTP flow attacks are mounted. This is expected to happen because the subcase 1 does not have IDS in the experiment setup. For the subcase 3, the network bandwidth has dropped between 0.2-0.7 GBits/sec at second 10. NIDS (Suricata) begins to detect the SMTP flow attacks whereby some of the SMTP flow attacks are dropped. For the subcase 4, the network bandwidth has dropped to 2 GBits/sec at second 15. By the combination of FlowIDS (Suricata & proposed algorithm), both systems have offered better SMTP flow attack detection and prevention whereby the network bandwidth is less plunged during the attacks.

Based on the graphs as shown in the Figure 13, we have used the network bandwidth consumptions as the performance benchmarking between the four subcases. The subcase 4 FlowIDS (Suricata & proposed algorithm) has shown around 30% less network bandwidth consumption compared to the subcase 3 (Suricata) during the attacks. The results have also shown that the FlowIDS has improved the network recovery rates that better than the standalone Suricata (subcase 3). Based on the results of the experiment, we can conclude that the proposed FlowIDS in this work has provided a significance contribution in detecting and preventing SMTP flow attacks on SDN.

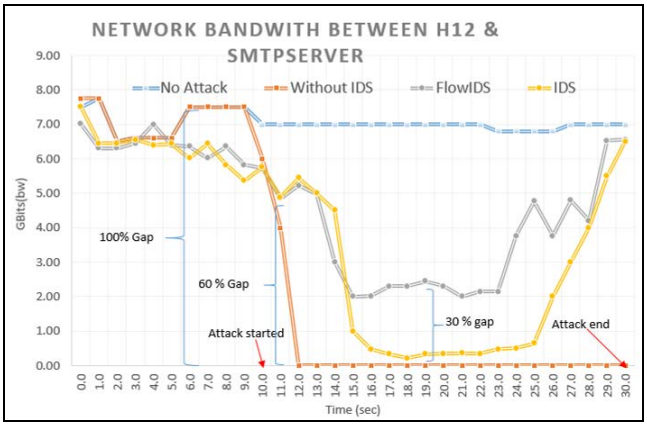

Figure 13. Experiment results of the four subcases

6. Conclusion

We have presented a framework for anomaly detection and prevention on SMTP traffic flows, namely FlowIDS. The proposed method allows the FlowIDS to update the ONOS controllers with the latest SMTP spam signatures.

It will prevent any SMTP spam email from entering others SDN domains. We also discussed the method for analyzing SMTP traffic flows using decision tree algorithm. We have shown that by the combination of FlowIDS, ONOS and NIDS, these integrated systems have offered better SMTP flow attack detection and prevention compared to standalone NIDS as the main security parameter. For the future work, we are planning to integrate the FlowIDS with multi-domains of SDN distributed platform to enhance detection and prevention SMTP flow attacks on the Internet.

Acknowledgement

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

References

[1] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," 2012.

[2] 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].

[3] 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.

[4] 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.

[5] ONOS, "ONOS," 2017. [Online]. Available: http://onosproject.org/. [Accessed: 05-Feb-2017].

[6] 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.

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

[8] 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.

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

[10] 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.

[11] 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.

[12] C. Schafer, "Detection of Compromised Email Accounts Used by a Spam Botnet with Country Counting and Theoretical Geographical Travelling Speed Extracted from Metadata," 2014 IEEE International Symposium on Software Reliability Engineering Workshops, pp. 329–334, 2014.

[13] 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.

[14] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," 2014 IEEE Conference on Communications and Network Security, pp. 247–255, 2014.

[15] T. Ouyang, S. Ray, M. Allman, and M. Rabinovich, "A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise," Computer Networks, vol. 59, pp. 101–121, 2014.

[16] T. Ouyang, S. Ray, M. Allman, and M. Rabinovich, "A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise," Computer Networks, vol. 59, pp. 101–121, 2014.

[17] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, "Towards Autonomic DDoS Mitigation using Software Defined Networking," Proceedings 2015 Workshop on Security of Emerging Networking Technologies, no. February, 2015.

[18] J. Jeong, J. Seo, G. Cho, H. Kim, and J.-S. Park, "A Framework for Security Services Based on Software-Defined Networking," 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, pp. 150–153, 2015.

[19] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," IEEE Communications Magazine, vol. 53, no. 4, pp. 52–59, 2015.

[20] P. Holl, "Exploring DDoS Defense Mechanisms," in Proceedings of the Seminars Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM), Winter Semester 2014/2015, 2015.

[21] N. F. Huang, C. Wang, I. J. Liao, C. W. Lin, and C. N. Kao, "An OpenFlow-based collaborative intrusion prevention system for cloud networking," in Proceedings of 2015 IEEE International Conference on Communication Software and Networks, ICCSN 2015, 2015, pp. 85–92.

[22] Q. Yan, F. R. Yu, S. Member, Q. Gong, and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments : A Survey , Some Research Issues , and Challenges," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 2–23, 2016.

[23] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013.

[24] "Dataset internet traffic from University New Brunswick (UNB) Canada." [Online]. Available:

http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html.
[25] 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/. [Accessed: 03-Feb-2013].

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 University Technology MARA, Shah Alam, and 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.