Poster: Flow Inspection Scheduling for Cyber Security on Software-Defined Networks

Sunghwan Kim, Seunghyun Yoon, and Hyuk Lim
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology (GIST)
Gwangju 61005, Republic of Korea
{sunghwankim, seunghyunyoon, hlim}@gist.ac.kr

***Abstract*—Software-defined networking (SDN) is an emerging network technology that decouples the control plane from the data plane. Using SDN, a fine-grained flow-level inspection for cyber security can be performed by capturing suspicious packets on the network and steering the captured traffic to a traffic analyzer. This work proposes a flow inspection scheduling for various types of traffic flows on SDNs in order to enhance the traffic inspection performance when the inspection capability of traffic analyzer is limited. It performs continuous traffic sampling for suspicious traffic and newly generated traffic while normal flows are probabilistically sampled. The performance of the flow inspection scheduling scheme was evaluated through SDN testbed experiments.**

I. INTRODUCTION

The number of Internet users and devices are explosively increasing with tremendous amount of network traffic, and securing the network from large numbers of cyber-attacks, including denial of service, advanced persistent threat, and malware, has become more important issue. The cyber-attacks can inflict a significant loss to the company or government establishment. For example, it was reported that the estimated damages of the CryptoWall 3.0 ransomware in the US were up to $325 million in 2015. As a cyber-threat defense mechanism, the role of a network traffic analyzer such as an intrusion detection system (IDS) has become important to identify and prevent the spread of these kinds of cyber-attacks via the Internet.

The total amount of network traffic is too enormous to monitor, and inspect the entire traffic; we therefore need to selectively capture traffic by using a sampling method, and inspect the behavior of the selected traffic. The traffic capturing can be easily implemented by software-defined networking (SDN) technology [1], which decouples the control plane from the data plane. Unlike conventional networking, SDN manages network resources centrally through the direct communication between an SDN controller and the switches on the network using the OpenFlow (OF) protocol. SDN technology enables us to duplicate and redirect the traffic of interest by simply updating a forwarding table in an OF-enabled switch; in essence, it is possible to sample and steer traffic to any destination flexibly and dynamically.

In [2], Ha *et al.* proposed a sampling rate decision algorithm for an IDS on SDNs, which minimizes the capture-failure rate of the malicious flow. Capture-failure rate indicates a probability that the IDS fails to recognize a malicious flow. In [3], Yoon *et al.* proposed a sampling point and rate decision scheme using betweenness centrality measure for scalable network traffic monitoring on SDNs. The experiment results indicated that when all flows on the network are fairly sampled, fast detection of malicious flows is achieved. In selectively sampling traffic, however, there exists a possibility that some packets that include important signature for a security threat are missing and the threat detection fails. In such cases, the IDS requires to perform continuous packet capturing of suspicious flows for some time duration to accurately investigate the flows. In addition, newly generated flows should be intensively investigated in order to avoid the propagation of attacks that exploits victims on the network.

II. PROPOSED ALGORITHM

We propose a flow inspection scheduling scheme that determines traffic sampling method, time, and duration for different types of traffic flows in order to enhance the inspection performance, while total aggregated sampled traffic volume remains lower than the capacity of the IDS.

Two types of traffic sampling methods, i.e., continuous sampling (CS) and probabilistic sampling (PS) are considered. For newly detected flows and suspicious flows that are identified by the IDS, the continuous sampling for a certain time duration is more desirable because the behavior of data flows can be exactly discovered without any missing packets belonging the flows. For the continuous sampling, an SDN controller duplicates these flows and steers them to the IDS for a predetermined duration. The duration should be carefully determined because each security attack needs a different observation time to be accurately classified with a high detection rate and a low false rate. It can be tabulated for different types of attacks and chosen according to the IDS alarm information if a flow is reported to be suspicious. If there is no new flows nor suspicious flows, the probabilistic sampling is preferred. Each flow can be fairly sampling with the same ratio p_s, which is calculated by p_s = min(1, 'available IDS capacity' / 'traffic rate summation of all flows'). This rate-proportional sampling can achieve balanced traffic monitoring for normal flows excluding new or suspicious flows.

Figure 1 illustrates the operation of the flow inspection scheduling scheme. Initially, there are three flows, i.e., f_1, f_2, and f_3. Since they are normal, PS performs until f_1 is reported to be suspicious by the IDS. At t = 1, the SDN controller received

This work was supported by IITP grant funded by the Korea government (MSIP) (No. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats).

the IDS alarm for f_1 and duplicates every packet belonging to f_1 to the IDS. This CS lasts until f_1 is dropped at $t = 5$. From $t = 1$ to 3, PS is performed for f_2 and f_3, but the sampling ratio p_s becomes smaller because f_1 is sampled at the full rate of f_1 and the available IDS capacity is reduced by the rate of f_1. At $t = 3$, new flow f_4 is detected by the SDN controller, and all packets of f_1 are also duplicated during n_4 seconds. In this figure, n_4 is set to 4. At t = 7, time duration n_4 for f_4 is over, and PS is performed for every flow again.

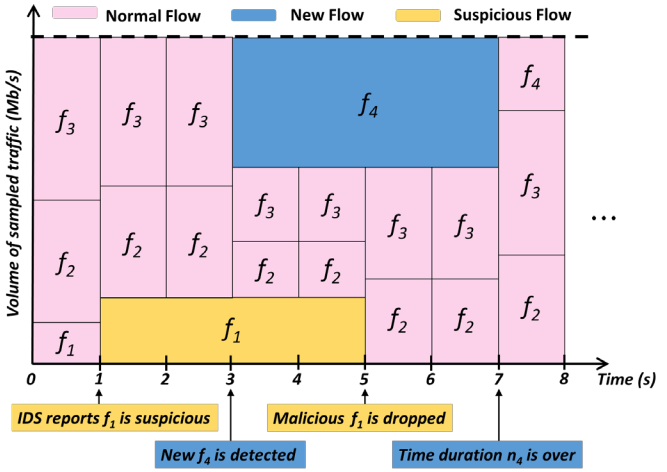

Fig. 1 An example of flow inspection schedule for four flows.

III. PERFORMANCE EVALUIATION

To verify and empirically evaluate our scheduling scheme, we constructed an OF-enabled network testbed. Figure 2 depicts the fat-tree topology of our OF-enabled testbed network, consisting of ten HP 2920 OF-enabled switches. The SDN testbed is managed by an OpenDaylight SDN controller running on a workstation, which is one of the most popular SDN controllers. For malicious packet inspection, we used Snort, which is an open-source, lightweight IDS, on another workstation. In the experiment, the number of flows is 4, and their data rate of flows varies from 200 to 800 Mb/s. The bandwidth of each link is 1 Gb/s, and the total processing capacity of Snort is 1Gb/s. The ODL updates the flow inspection schedule either when it receives an 'OFPT_PACKET_IN' message for newly added flows or when Snort reports the alarm of suspicious flow to ODL.

Figure 3 shows the aggregated throughput of sampled traffic forwarded to the Snort IDS. Initially, there are 3 flows, and the amount sampled traffic for each flow is rate-proportional. After 6 seconds, the Snort IDS reports that flow 1 is suspicious, and every packet belonging to flow 1 is duplicated and forwarded to the Snort IDS. At 17 seconds, the ODL controller detects a new flow and conducts continuous inspection for it. The proposed inspection scheduling scheme on the ODL inspects suspicious flow and newly generated flow continuously for a while. The remaining IDS capacity is rate-proportionally used for PS of the normal flows. It is also observed that the aggregated throughput of sampled traffic does not exceed the total capacity of the Snort IDS under the proposed scheduling scheme.

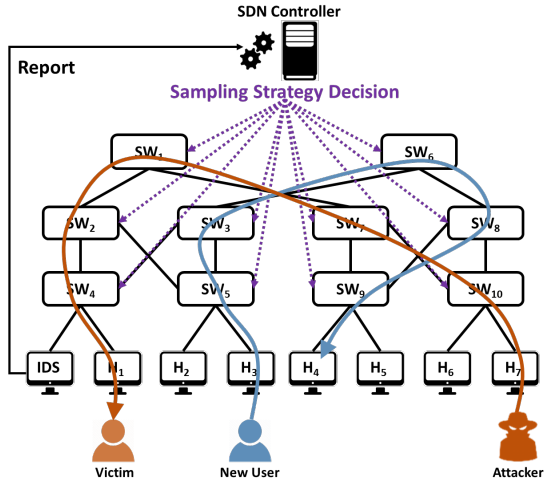

Fig. 2 Fat-tree topology of OF-enabled network testbed.

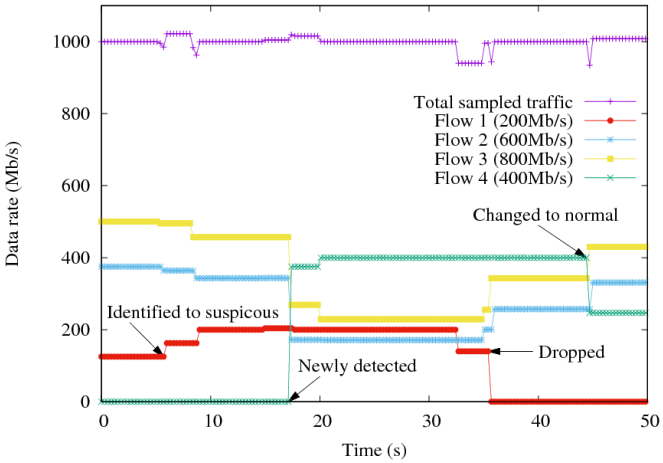

Fig. 3 Flow inspection scheduling measurement results.

IV. CONCLUSION

We proposed a flow inspection scheduling scheme that determines how to sample, when to sample, and how long to sample traffic flows based on the types and behaviors of traffic flows. If the flows are newly generated or suspicious, their packets are continuously forwarded to the IDS without any missing packets. Normal flows are probabilistically sampled at a same rate and it achieves a rate-proportional fair sampling, while the aggregated throughput of sampled traffic remains within the capacity of the IDS.

REFERENCES

[1] T. Ha *et al.*, "RTSS: Random traffic sampling and steering for OpenFlow-enabled networks," ACM CoNEXT Student Workshop, Irvine, California, December 12-15, 2016.

[2] T. Ha *et al.*, "Suspicious traffic sampling for intrusion detection in software-defined networks," Elsevier Computer Networks, vol. 109, pp 172-182. November 2016.

[3] S. Yoon *et al.*, "Scalable traffic sampling using centrality measure on software-defined networks," IEEE Commun. Mag., July 2017.