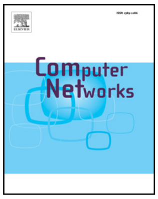
Computer Networks 000 (2016) 1–12

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Suspicious traffic sampling for intrusion detection in software-defined networks[☆]

Taejin Ha, Sunghwan Kim, Namwon An, Jargalsaikhan Narantuya, Chiwook Jeong, JongWon Kim, Hyuk Lim[*]

School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Republic of Korea

ARTICLE INFO

Article history:
Received 30 September 2015
Revised 4 March 2016
Accepted 29 May 2016
Available online xxx

Keywords:
Intrusion detection
Cloud technology
Software-defined network
Traffic sampling

ABSTRACT

In order to defend a cloud computing system from security attackers, an intrusion detection system (IDS) is widely used to inspect suspicious traffic on the network. However, the processing capacity of an IDS is much smaller than the amount of traffic to be inspected in a large-scaled network system. In this paper, we propose a traffic sampling strategy for software-defined networking (SDN) that fully utilizes the inspection capability of malicious traffic, while maintaining the total aggregate volume of the sampled traffic below the inspection processing capacity of the IDS. We formulate an optimization problem to find an appropriate sampling rate for each switch, and sample the traffic flows in the network according to the optimal sampling rates using the SDN functionalities. The simulation and experimental results indicate that the proposed approach significantly enhances the inspection performance of malicious traffic in large-sized networks.

1. Introduction

Cloud technologies have become some of the most promising next-generation technologies for the efficient and cost-effective management and operation of diverse industrial applications. A cloud system is a distributed platform that manages and coordinates a number of distributed devices and services, and provides a wide range of services and functions. Cloud services are expected to be deployed to enhance the operation and management efficiency for a variety of Internet application and service systems. To strengthen the security in a cloud system, an IDS is widely deployed. An IDS is one of the most important means for protecting a network from threats. It is used to monitor network behaviors and inspect data packets for detecting malicious activities. In a conventional network, the IDS is configured to operate in two modes, passive and in-line modes [2]. In the passive mode, the IDS is connected to the network as a subordinate of a network node, and it passively receives and inspects data packets captured from the node. In the in-line mode, the IDS is set up as a stand-alone node in the network. In both modes, the IDS is located at a certain location connected to a network link and conducts a traffic analysis on local flows going through the link.

However, because of an explosive expansion in the network scale and the increase in network traffic, it has become much harder to determine the inspection points where the IDS is placed and where the data packets are captured within the network. This is also because the IDS has a limitation of hardware resources in terms of CPU power, memory access speed, and storage capacity [3]. In particular, for a large-scale network security system, a number of IDSs are needed to be deployed to inspect all data packets owing to their inspection capability, which incurs high costs.

With the aim of achieving a scalable inspection of a large-scaled network with a limited capacity IDS, we use software defined networking (SDN) technology. SDN is an emerging network architecture that decouples the network control plane from the packet forwarding plane (data plane) [4]. It has a centralized controller, called an SDN controller, which is responsible for all network control decisions of the network-wide distributed forwarding elements [5]. The controller uses an OpenFlow (OF) protocol, which is a communication networking protocol, to access OF-enabled switches using OF APIs [5]. Since the scale of cloud systems is expanding, and communication network management between distributed server systems requires diverse functionalities, SDN is a promising solution for the networking architecture connecting between the server host systems. Under an SDN-based network system, it is possible to easily sample data traffic from multiple switches and forward them to one of the IDSs in the cloud system [4].

[☆] A part of this work has been demonstrated at the demo session in IEEE International Conference on Cloud Networking (CloudNet), October 2014 [1].

[*] Corresponding author.
E-mail address: hlim@gist.ac.kr (H. Lim).

http://dx.doi.org/10.1016/j.comnet.2016.05.019
1389-1286/© 2016 Elsevier B.V. All rights reserved.

In this paper, we consider an IDS to prevent malicious data propagation in an SDN-based network. If the network traffic to be inspected is much larger than the IDS capacity, the IDS cannot inspect all packets in the network. Therefore, it is desirable to sample a certain amount of data traffic from the network switches and forward it to the IDS using the SDN functionalities. We propose a sampling rate adjustment method that determines the appropriate sampling rates at the network switches for fully utilizing the inspection capability of malicious traffic while the total aggregate volume of the sampled traffic is kept below the maximum processing capacity of the IDS.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of previous related work. In Section 3, we present the system model and derive a relationship between the sampling rate and the capture-failure rate of malicious traffic when the network traffic is sampled. Then, in Section 4, we propose a measurement-based sampling method along with the details regarding the algorithm used for obtaining the appropriate sampling rates. In Sections 5 and 6, a performance evaluation is presented, followed by concluding remarks in Section 7.

2. Related work

There has been a significant amount of work on intrusion detection technology and systems. Most research has focused on how to process a large amount of traffic samples efficiently. They proposed using various traffic characteristics such as the flow size, flow statistics, and flow entropy changes [3,6,7]. In [3], Androulidakis et al. proposed a flow-size based sampling technique. The authors considered that network attacks usually use small flows as a traffic source. Based on the observation, those flows smaller in size than a certain threshold are sampled with a constant probability. In [6], Kawahara et al. introduced a flow-statistic based intrusion detection strategy. The authors noted that the number of flows increases dramatically if a network is under attack. They spatially partitioned sampled traffic into several groups according to a source-autonomous system. Their experimental results indicate that an analysis of the flow statistics for individual groups can enhance the ability of intrusion detection. In [8], Mai et al. showed how traffic sampling degrades the detection ability of non-volume dependent anomalies. The authors used three intrusion detection algorithms to detect non-volume based attacks from both the original and sampled trace data. The results indicate that traffic sampling can increase the number of false positives and degrade the detection ability. In [9], Kacha et al. proposed a new pattern matching technique for improving the performance of Snort IDS. To reduce a rate of false alarms, it combined misuse and anomaly detection techniques. This method provides faster packet inspection with less consumption of the IDS resource compared to the conventional one.

Recently, research efforts have been made to exploit the emerging SDN technology for network security by a number of research groups. In [2], Shin et al. introduced a framework, called Cloudwatcher, that provides monitoring service for cloud networks. Using SDN technology, Cloudwatcher changes the paths of the network flows to allow them to pass through more secure links, or to have their data packets inspected by an IDS. In [10], Khurshid et al. used a packet forwarding operation of an SDN in real time for eliminating network errors such as routing loops, black holes, and access control violations. Since these errors mostly result in the unavailability of a service, they are weak points and make a network vulnerable to network attacks. In [11], Jafarian et al. proposed a technique for changing the Internet protocol (IP) address to protect an end node from attackers. This technique frequently mutates the actual IP address of the end-host into a fake virtual IP address. In [7], Giotis et al. proposed an entropy-based detection algorithm that measures the randomness of specific data sets in an SDN environment. The authors classified network attacks by observing the entropy changes. For instance, if there is a significant decrease in the number of destination IPs and destination ports, the network is considered to be under a DDoS attack.

Most of the existing work has focused on the sampling and inspection of traffic packets observed from only a single point in either a traditional or SDN-based network. In contrast, we propose a distributed sampling scheme that samples suspicious data packets from multiple switches on an SDN-based network. In consideration of the IDS inspection capacity, the proposed scheme determines the appropriate sampling rate for each switch to fully utilize the detection ability of an IDS. We provide an analytical model for evaluating the capture-failure rate of malicious traffic in an SDN. The network simulation results indicate considerable gains in the detection of malicious traffic when the IDS capacity is much smaller than the total traffic to be inspected.

3. System model

3.1. System description

We consider the SDN-based network shown in Fig. 1. The SDN-based network architecture is composed of an SDN controller and OF-enabled switches. OF is a communication networking protocol that enables the SDN controller to access the forwarding table of the switches. The SDN controller connects with the OF-enabled switches on the control plane, and by using OF, it can gather the flow status at each switch and control its flow forwarding table. All data packets are exchanged through OF-enabled switches operated by the SDN controller.

An IDS is usually used to inspect all or a certain number of packets that the IDS can capture at its connected links. A signature based IDS compares captured packets against a database of signatures from known threats. For example, Snort IDS [12] is one of the popular open-source IDSs, and it can detect malicious probes and attacks such as server message block probes, stealth port scans, and malicious code injection by analyzing network traffic against a rule set. It can also detect protocol anomalies such as TCP SYN flood attacks using a variety of preprocessors.

Fig. 1 shows the suspicious traffic inspection on an SDN-based network with IDS. For an SDN-based IDS, it is possible to sample the data packets at any OF-enabled switches by using a mirroring method at each switch, which is fully configurable by the SDN controller. The IDS then inspects all of the data packets that are mirrored from the switches, and generates a security alarm if it detects a network attack or suspicious packet. The security alarm is fed back to the SDN controller. Based on the inspection results of the IDS and the current status of the switches, the SDN controller can reconfigure the network to defend against an attack and make the network more secure.

While the SDN technology enables traffic flows at the switches to be sampled and forwarded to the IDS, the IDS cannot inspect all sampled packets if the sampled traffic volume is greater than its processing capacity. Therefore, an algorithm for determining the sampling rates of the switches needs to be developed to ensure that the total amount of sampled traffic is kept below the maximum inspection capacity of the IDS while minimizing the rate of missing malicious traffic packets.

3.2. Network model

We assume that there are f flows and n OF-enabled switches in the network. Let λ_i denote the malicious rate belonging to the ith flow. If a flow does not include any malicious packets, its malicious