Vol.2018-IOT-40 No.34
2018/3/6

A proposal of a real-time OpenFlow DDoS detection tool

Wassapon Watanakeesuntorn[1,a)] Kohei Ichikawa[1,b)] Hajimu Iida[1,c)]

Abstract: The controller of the SDN network is a single point of failure. When the controller is down, the SDN network may stop working. The controller can be attacked by an attacker with DDoS attack techniques. In this paper, we propose a DDoS detection in the OpenFlow network by analyzing the OpenFlow messages in control plane by using machine learning. For the purpose, we use a DRAPA's DDoS traffic dataset for training and evaluating the proposed model. The DRAPA dataset only contains the traffic data of data plane, however we generate a dataset in control plane by simulating the network traffic from the dataset, and use the generated dataset for training and evaluation. In addition, we plan to integrate the proposed mechanism with our "Opimon", OpenFlow Interactive Monitoring Tool, to monitor and detect DDoS attack in real-time.

1. Introduction

Nowadays, Internet is very large and more complex. It is very difficult to manage and control a computer network. Software-defined Network is an emerging technology that tries to simplify the management of the network. SDN tries to provide a new approch to the network technology. It tries to separate data plane and control plane. In the traditional network, the data plane and control plane is provide by a network switch. In the SDN, the control plane is provide by a centralize controller, which include routing decision and switch configuration, and the data plane is included in the network switch which is a dump switch for forwarding data. The advantages of SDN are trying to simplification and flexibility in the network.

OpenFlow protocol is a standard protocol which used for communicating between OpenFlow controller and OpenFlow switches. In addition, the OpenFlow controller can be programmable [1]. It has many OpenFlow library or framework to develop the OpenFlow controller such as NOX, POX, Ryu [2], and others which the programmable controller can make the OpenFlow network more dynamically and easy to control and manage the network.

The behaviour of the OpenFlow network is determined by a cooperative between an OpenFlow controller and OpenFlow switches [3]. When a flow comes to an OpenFlow switch, the switch will try to lookup the flow in the flow table, which contain a match of the flow and an action of the switch. If the flow matches with attribute in the flow table, the switch will follow the action on the flow table's attribute. If the flow not match to any attribute in the flow table, the switch will send OpenFlow message to the OpenFlow controller for asking the action for this flow. The controller will find the route for this flow and response with flow modification message. When the switch receives flow modification message from the controller, the switch adds the flow into the switch memory or flow table and do the action which contain in the message from the controller. If the same flow comes to the switch, the switch will follow the action in the flow table without sending the OpenFlow message to the controller. The OpenFlow network is shown in Figure 1.

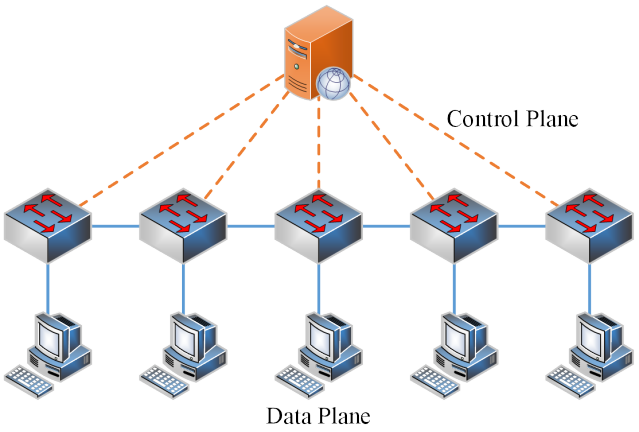

Fig. 1 OpenFlow Network

On the downside of the OpenFlow network is the centralize controller. The controller of the OpenFlow network is providing routing algorithm and controlling the OpenFlow switch. The OpenFlow controller can be a single-point of failure in the OpenFlow network. When the controller is unavailable, the switch will not add a new flow in to switch's flow table which can cause the switch drop every new flow and the network may down. In addition, a switch memory can be attacked. When the switch memory is full, the switch will not add a new flow or action from the controller into the flow table which can cause the switch cannot forward the new flow or the packet and the network may be down. These problems are the weak points of the OpenFlow network

[1] Nara Institute of Science and Technology, 8916-5, Takayama-cho, Ikoma, Nara, 630-0192, Japan
[a)] wassapon.watanakeesuntorn.wq0@is.naist.jp
[b)] ichikawa@is.naist.jp
[c)] iida@itc.naist.jp

that can happen when the network is attacked by attackers.

Denial of service (DoS) is one of a common cyber-attack in a computer network which aims to make the resource or service unavailable for a legit user. If the attack traffic comes from multiple sources, it is called "Distributed Denial of Service (DDoS)". Behaviour of DDoS is flood a huge of network packets to a victim until the service or resource is unavailable. There are many types of packets which use in a DDoS attack and some type of DDoS attack is very hard to detect. The DDoS can cause many problem on the OpenFlow network. There have many research about DDoS detection in the OpenFlow network. They try to provide a technique to detect DDoS in the OpenFlow network. Deep learning is one of popular techniques which use to detect a DDoS.

Deep learning is one of the popular machine learning methods. It based on artificial neural network technique. Deep learning can classify data based on the trained model that it is used to classify in many classification problem. It can train deep learning to supervised learning, which is trained by labeled data that we know the output of the trained dataset, or unsupervised learning, which is train model by unlabeled data that the system can classify by themself.

In this research, we purpose to use deep learning to detect the DDoS in the OpenFlow network. In addition, we try to implement this method in our OpenFlow monitoring tool, Opimon (OpenFlow Interactive Monitoring), to monitoring the OpenFlow network and detect DDoS in real-time.

2. Related Works

There are many research on SDN and OpenFlow network. However, there are not much research about security field in the OpenFlow network. In the literature, they try to provide their method to detect DDoS in the OpenFlow network. There has a concise survey about security field to against DDoS attack in SDN or OpenFlow network [4]. In this article, they collect many research for DDoS detection and mitigation in the SDN and OpenFlow network. It has many kind of techniques to detect DDoS in the SDN such as managing the flow table to mitigate the limitation of switch memory when DDoS attack is occurs which they suggested that table replacement policies should use multiple parameters such as number of packets, generation data and other properties of the flow, rather than one parameters such as flow timeout [5]. One of the interesting techniques in the survey is the use machine learning to detect the DDoS in SDN. On this survey article, it suggests a research which they use a support vector machine (SVM) classifier to classify the traffic and detect DDoS in SDN [6]. In addition, they tried to compare SVM with other machine learning techniques and they conclude that SVM has the best performance in thier research. They use DRAPA dataset [7] which is a DDoS traffic in data plane. However, we think using data plane traffic only is may not enough to detect DDoS in some case for OpenFlow network such as a traffic that aims to flood the OpenFlow messages to the controller. We purpose to use the data plane dataset and control plane dataset to fully detect DDoS in the OpenFlow network.

For the literature, SVM is typically used for supervised binary classification but we purpose to train an unsupervised learning to detect a DDoS in the OpenFlow network that deep learning can be trained into supervised or unsupervised learning tasks. There are many research about using deep learning to detect DDoS in deep learning. In [8], this paper analyzed many of machine learning techniques for handling the issues of intrusion and DDoS attacks on SDN. they compared 5 machine learning techniques which they show pros and cons of these 5 techniques. For the neural networks, they use a neural network to classify normal and attack traffic patterns. They concluded that the neural network is capable to generalize from limited, noisy, and incomplete data. In addition, neural network does not need expert knowledge to use it to classification. However, they mention that the nueral network is slow training process and may not suitable for real-time detection.

In [9], they purpose to use deep learning to detect DDoS in software defined network environment. They implemented the DDoS detection on the top of SDN controller. They mention that their tool can classify the normal and attack traffic with an accuracy of 99.82% with very low false-positive. However, their DDoS detection system implemented as a network application on SDN network and it uses Northbound API which it may not be compatible with every controller because Northbound API is not standard for every SDN controller.

From our previous work [10], we develop an OpenFlow Interactive Monitoring (Opimon) which is a general monitoring tool for OpenFlow network. It monitors the OpenFlow network as a proxy monitoring tool and it compatible with every OpenFlow controller and OpenFlow network. We approach to add a new feature to our tool which is a DDoS detection in real-time. From many literature, we purpose to use a deep learning to detect a DDoS in OpenFlow network.

3. Approach

Our approach is tring to develop a general monitoring tool for an OpenFlow network which includes security measurement to protect the OpenFlow network from attackers. From previous work, we developed a real-time monitoring tool called "Opimon". In this work, we purpose to add security feature in Opimon which is detecting a DDoS attack in the OpenFlow network. We purpose to implement deep learning technique to classify the normal traffic and attack in OpenFlow network. We use traffic from the data plane and control plane to detect a DDoS attack in the OpenFlow network. For training dataset, we use DRAPA dataset which include a DDoS traffic in data plane. For the control plane dataset, we simulate the network traffic from DRAPA dataset and monitor the OpenFlow messages between an OpenFlow controller and OpenFlow switches. On training model, we use the traffic from 2 planes to train the model. We purpose to train an unsupervised learning with deep learning to detect a new type of DDoS attack in the OpenFlow network.

4. Design and Implementation

We design Opimon as a general monitoring tool for OpenFlow network which it monitors the communication between switches and OpenFlow controller in the control plane and visualize the OpenFlow network on the web interface. In addition, we pur-

pose to include security feature into the Opimon which is DDoS detection. We use a machine learning technique to analyze the flow and OpenFlow message, which the monitoring tool collects into the database, to classify the traffic between normal traffic and attack traffic. Originally, the Opimon has 2 parts, which are Monitoring part and Visualizing part. The architecture of Opimon is illustrated in Figure 2. Then we add a new part into Opimon, Analyzing part, as it shown in Figure 3.

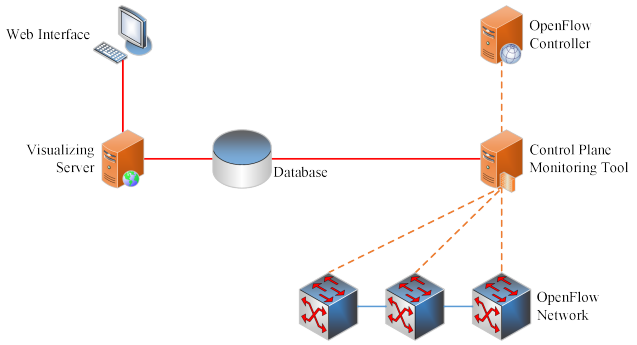

Fig. 2 Original Opimon Architecture

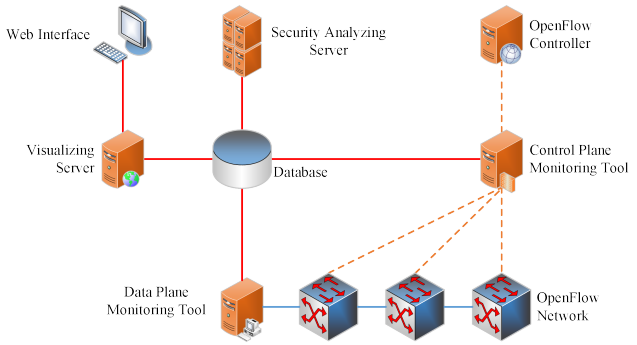

Fig. 3 Opimon with Security Analyzing Server

Monitoring part is a part for monitoring communication between the OpenFlow controller and OpenFlow switches with a monitoring tool. The monitoring tool collects the OpenFlow message between the controller and the switch into the database. Opimon's monitoring tool is a proxy monitoring the communication between the OpenFlow controller and OpenFlow switches. It uses Ryu SDN Library for reading the OpenFlow messages. Ryu SDN Library is an OpenFlow library which is wrote from Python programming language and Ryu is updated library and easy to maintain the code for our monitoring tool. The monitoring tool reads the OpenFlow message from the controller or the switches and collects the message information into a database. We use MongoDB database, which is NoSQL database, because it more flexibility to collect the data which it may have different attribute in the message.

Visualizing part is a part for visualizing overview information about the OpenFlow network to the user via web interface. The main component in visualizing part is visualizing server. It query data from database and process a raw OpenFlow message into a visualize-ready format for reducing data process on the web browser on the client side. The visualizing server runs web back-end service to receive and response a request from the web browser. We use Node.js for the web back-end service. On the front-end side, we use D3.js and jQuery to visualize data to users via the web interface. On the web interface, it shows an overview of monitored network such as OpenFlow network topology, statistics of OpenFlow switches and active flow in the flow table for each switch. In addtion, it shows the alert of the DDoS attack on the network. The web interface displays the information in real-time. It refreshes the information of the network every one minute. The web interface is shown in Figure 4.

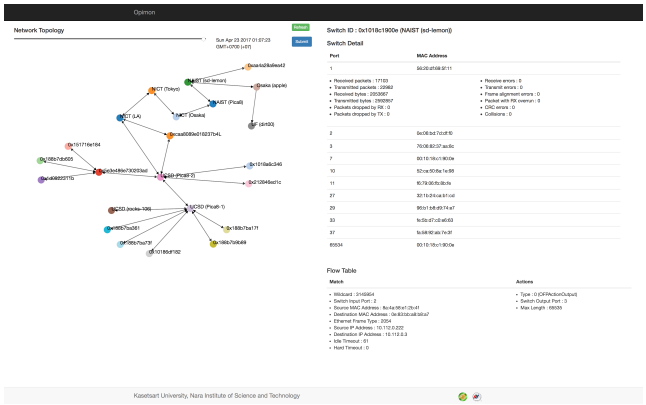

Fig. 4 Web interface

Analyzing part is a part for analyze the network traffic and classify between a normal network traffic and attack traffic which is using for detecting a DDoS on the network. The main component of this part is a security analyzing server which used to analyze the traffic from the data plane and control plane traffic. We use Python programming language to implement deep learning and run it on security and analyze server. For a training dataset, we use DRAPA dataset as a dataset in the data plane. For the control plane training dataset, we simulate a traffic from DRAPA dataset in simulated OpenFlow network and capture the communication or OpenFlow messages between the OpenFlow controller and the OpenFlow switches. After collecting the control plane training dataset, we train a deep learning model from data plane dataset and control plane dataset. We verify the model by using other DDoS traffic dataset and run them with our model to find the accuracy of our model. In addition, We add new monitoring tool for monitoring a data plane traffic from a core switch on the network to monitor the traffic in the data plane for detecting the DDoS attack in the OpenFlow network.

5. Evaluation and Progress

We use DRAPA dataset, which include DDoS traffic in data plane, and simulate the dataset in the OpenFlow network to collect the OpenFlow message during DDoS in the control plane. We build a simulated OpenFlow network to simulate the traffic from dataset. We use Mininet, a network emulator, to simulate the OpenFlow network and we use our monitoring tool, Opimon, and Wireshark, the worlds foremost and widely-used network protocol analyzer, to monitor and capture the communication in the control plane. We collect all of the OpenFlow messages into the database and use data plane dataset and control plane dataset for

training a deep learning model to classify the normal traffic and DDoS traffic. For current progress, we are trying to simulate the dataset into simulated OpenFlow network to get the OpenFlow messages. In addition, we are trying to optimize the performance of Opimon and deep learning accuracy.

6. Conclusion

Opimon is a general monitoring tool for OpenFlow network which it shows overview information about monitored OpenFlow network includes with network topology, flow table and statistics of the OpenFlow switches. In addition, Opimon can detect a DDoS in the OpenFlow network by using deep learning to classify the traffic in real-time. It uses a data from the data plane and control plane traffic to analyze and detect DDoS. We try to optimize the performance and accuracy of the result of deep learning. For future work, we plan to add a DDoS mitigation for the OpenFlow when DDoS is detected. In addition, we plan to improve a user interface and show the path of the flow into the web interface.

7. Acknowledgement

This work was partly supported by JSPS KAKENHI Grant Number 15K00170. Funding for this research is partly provided by International Priority Graduate Programs (IPGP) and Creative and International Competitiveness Project (CICP) 2017.

References

[1] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, ”OpenFlow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev. 38, 2 (March 2008), 2008, pp. 69-74.

[2] Ryu SDN Library, https://osrg.github.io/ryu-book/en/html/

[3] OpenFlow 1.0.0 Specification, 2009, http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf

[4] K. Kalkan, G. Gur and F. Alagoz, ”Defense Mechanisms against DDoS Attacks in SDN Environment,” in IEEE Communications Magazine, vol. 55, no. 9, pp. 175-179, 2017.

[5] N. Tri, T. Hiep, and K. Kim, ”Assessing the Impact of Resource Attack in Software Defined Network,” Proc. 2015 Intl. Conf. Info. Networking, 2015, pp. 42025.

[6] Kokila RT, S. Thamarai Selvi and K. Govindarajan, ”DDoS detection and analysis in SDN-based environment using support vector machine classifier,” 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, 2014, pp. 205-210.

[7] DARPA Intrusion Detection Data Sets, https://www.ll.mit.edu/ideval/data/

[8] J. Ashraf and S. Latif, ”Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques,” 2014 National Software Engineering Conference, Rawalpindi, 2014, pp. 55-60.

[9] Niyaz, Quamar & Sun, Weiqing & Javaid, Ahmad. (2016). A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN). ICST Transactions on Security and Safety. 4. . 10.4108/eai.28-12-2017.153515.

[10] W. Watanakeesuntorn, P. Uthayopas, C. Chantrapornchai and K. Ichikawa, ”Real-time monitoring and visualization software for OpenFlow network,” 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand, 2017, pp. 1-5.