Intelligent Cyber-attack Defense System using Virtual Honeynets for Cloud Security

Jargalsaikhan Narantuya, Jiwon Yang, JongWon Kim, and Hyuk Lim
Gwangju Institute of Science and Technology (GIST)
123 Cheomdan-gwagiro, Buk-gu, Gwangju 61005, Republic of Korea
{jargalsaikhan.n, jwyang, jongwon, hlim}@gist.ac.kr

Abstract

We propose an artificial intelligence-based cyber-attack defense system for cloud security. The defense system is built based on a virtual honeynet, which is a fake network to invite and monitor suspicious traffic flows. The honeynet is implemented by using virtualization technologies, such as Docker-based containerization and virtual switches. Since containerization technology enables to launch services in a couple of seconds, it is possible to create highly-interactive honeypots on demand and to provide deep learning-based traffic analysis. Additionally, software-defined networking functionalities are efficiently used to implement flexible management and to hide the honeynet from attackers.

Introduction

As information technology advances, network attack techniques also become more intelligent, and it makes existing defense systems vulnerable to the latest attack strategies. This leads to the development of deep learning (DL)-aided intrusion detection system (IDS), which has a great potential of finding out the distinct characteristics of unusual network patterns and helps to block various cyber-attacks in advance. In this regard, artificial intelligence (AI)-based cyber-attack defense is becoming a promising solution for cloud security.

In this work, we propose a cyber-attack defense system that embeds a simple yet powerful DL-based strategy to segregate abnormal flows from the benign traffic. In order to learn patterns of malicious traffic without losing connection to the attacker, we implement an intelligent honeynet by leveraging software-defined networking (SDN) and virtualization technologies. A honeynet is a fake network designed to invite attackers and control their actions without knowing the system is fake. To track down the attackers and their targets, it is important to ensure that the honeynet provides the same functionalities of a real network. This kind of honeynet is mostly built by using high-interaction honeypots (HIHs) that enable to study both network and system activities of the attackers. However, existing HIHs are created among physical servers, and physical network devices are also required to connect them. Therefore, building an HIH-based honeynet on a large scale requires higher operational costs due to its complex management and inconvenient deployment.

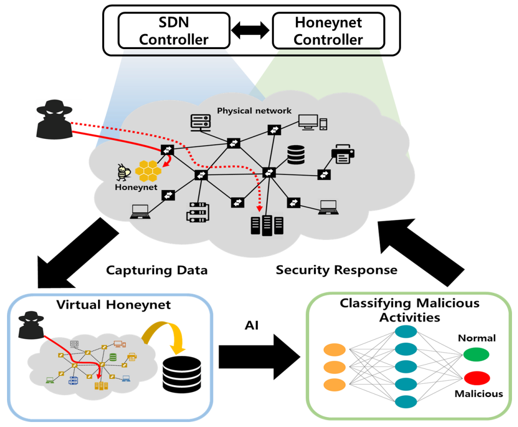

Figure 1: Proposed cyber-attack defense

There exist several works about SDN and AI-based cyber-attack defense with honeynet. In (Fan et al. 2019), the authors proposed a novel honeypot system named HoneyDOC, which is designed to support all-round honeypot. The proposed system consists of three main modules such as decoy, orchestrator, and captor. These modules are used to enable all-round design for capturing high-quality of attack data. In (Rashidi et al. 2018), the authors proposed a new honeynet system named HoneyV, which is built by leveraging the flexibility of SDN and network functions virtualization (NFV). HoneyV is designed to monitor multi-phase data in the network, and it redirects traffic flows to different types of servers based on their level of trust.

Proposed cyber-attack defense

Building virtual honeynet

We consider a cloud data center that consists of SDN-enabled switches and physical servers that host real services. As illustrated in Figure 1, a virtual honeynet provides a completely identical network and service topology as the data center. In this system, topology information of a real network is obtained with the help of SDN functionalities, and service information in physical servers is obtained by a developed application programming interface (API). Then, the honeynet is created based on the obtained information using the Open vSwitch (OVS) and pre-defined Docker images.

We develop a set of APIs to obtain service information from the physical network since SDN does not support this functionality. Specifically, the API server runs on the honeynet controller, and API agents run on both the physical and honeynet servers. The API server and agents communicate through HTTP messages. First, the honeynet controller obtains service information from the physical servers by sending a service request HTTP GET message to the API agents in physical servers. Next, it creates honeypots in the honeynet server based on the received service information by sending an HTTP POST message. After completing the build process, the honeynet controller sends network information (IP and MAC addresses) of honeypots to the SDN controller to send network traffic to the honeynet.

When new traffic arrives in the switch and does not match with any of the existing flow rules, the SDN sends a copy of that packet up to the controller. This procedure is performed by sending a "packet-in" message indicating "sending unknown traffic information to the controller". If this occurs, the controller decides an action (forward, drop, or modify) based on the pre-defined rule. The proposed system parses the destination IP address of the traffic to determine whether it is destinated to a real server or a honeypot. If the traffic is destinated to one of the servers in a real network, it is redirected to a honeypot that provides the same service as an original destination. In changing the traffic destined to a honeypot, the flow modification action of SDN is efficiently utilized with the network information of the honeynet.

Deep learning-based network traffic classification

To respond to the attack traffic sent to the honeypot, we monitor each action log of that traffic. In this way, the monitoring system automatically acquires features associated with each flow, and these features are utilized by a DL-based classifier to distinguish the malicious traffic from non-malicious ones. We mainly focus on TCP-related statistical features since TCP channels are highly exploitable to carry out attack campaigns. Below is the complete list of selected features:

- Average/Minimum/Maximum of TTL, TCP window, interarrival time, length of IP/TCP header, data payload
- Proportion of ACK, CWR, ECE, PSH, RST, SYN, URG
- Percentage of duplicate ACK, keep alive, lost segments, retransmission, window update, zero window packets

Considering that a vast amount of traffic traces must be handled within a short time, we developed a lightweight neural network model, which was relayed to five hidden layers. We employed an N-version deep neural network (NV-DNN) architecture by training eight independent classifiers to raise the model reliability while reducing erroneous prediction (Hui et al. 2019). If the traffic is categorized in the normal class, the honeynet controller lets the SDN controller remodify the flow and send the processed traffic to its original destination. Otherwise, it sends a warning sign to the SDN controller to block further packets conveyed over the corresponding flow channel and update IDS rule.

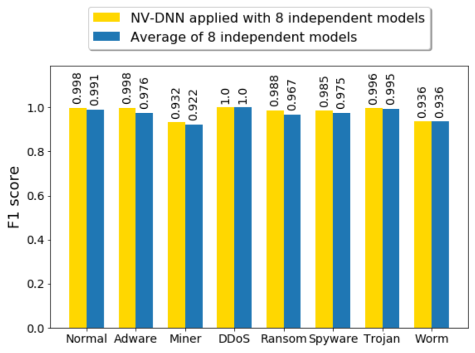

Figure 2: F_1-score of trained NV-DNN classifier

Result

For the evaluation of the DL-based traffic classifier to be embedded in the proposed honeypot system, we collected 36,000 traffic data [1] composed of normal traffic and 7 types of attack traffic data (evenly distributed). Figure 2 shows the F_1-score scores obtained by the independently trained models and NV-DNN implemented by these models. The NV-DNN classifier correctly identifies approximately 99.8% of the normal data as non-malicious, which means that it is capable of differentiating between normal flow and attack flow. Adware, DDoS, and trojan traffic samples are highly grouped in their actual classes, whereas the F_1-score scores of coin miner and worm are worse than others showing approximately 93%. In most cases, the NV-DNN shows better performance than the 8 separated classifiers, which demonstrates the performance improvement by model integration.

Acknowledgments

This work was supported by an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats) and (No. 2019-0-01842, Artificial Intelligence Graduate School Support)

References

Fan, W.; Du, Z.; Smith-Creasey, M.; and Fernández, D. 2019. HoneyDOC: An efficient honeypot architecture enabling all-round design. *IEEE Journal on Selected Areas in Communications* 37(3):683–697.

Hui, X.; Zhuangbin, C.; Weibin, W.; Zhi, J.; Sy-Yen, K.; and Michael, L. 2019. NV-DNN: Towards fault-tolerant dnn systems with n-version programming. *In Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*.

Rashidi, B.; Fung, C.; Hamlen, K. W.; and Kamisinski, A. 2018. HoneyV: A virtualized honeynet system based on network softwarization. In *IEEE/IFIP Network Operations and Management Symposium (NORMS)*, 1–5.

[1]Available in https://stratosphereips.org/category/dataset.html