Compressing Deep Learning Model for Agile Moving Target Defense

Xiaoyu Xu, Hao Hu, Xianwei Zhu

1 Zhengzhou Information Science and Technology Institute, Henan, Zhengzhou, China
2 China Electronic Engineering Design Institute, Beijing, China
*email: xxyin1992@163.com

Abstract. The moving target defense achieves the effect of defending against network attacks by constantly changing the attack surface. IP hopping defense is a typical representative of network layer moving target defense technology. It has been verified to show considerable defense effect against DDoS attacks and scanning attacks. Aiming at the problems that the system resource overhead of the existing IP hopping defense technology is too large, an agile IP hopping defense technology based on compressed neural network is proposed in this paper. A convolutional neural network (CNN) is deployed to sense the attack. In the training, the CNN uses techniques of clipping and quantizing to make the trained model show low storage occupation and high processing efficiency. The lightweight CNN determines the current attack situation according to the flow table data uploaded regularly by each switch in the data plane. Then configure and trigger two different levels of IP hopping according to the judgment results. Experimental results show that compared with the current typical IP hopping defense methods, the proposed method can significantly reduce the system overhead, including storage occupation, channel occupation and so on. In terms of security, the performance of the proposed method is equivalent to the existing state-of-art method against DDoS attacks and scanning attacks.

1. Introduction

By constantly changing the target system environment or resource allocation relationship, the moving target defense increases the difficulty for attackers to use vulnerabilities to create and maintain the attack chain, which can significantly improve the threshold of network attack and the cost of attackers. At present, moving target defense technologies on network layer can be divided into IP hopping defense technology, end information hopping defense technology, routing hopping defense technology and so on. This paper mainly focuses on IP hopping defense technology [1]. IP hopping defense technology makes it difficult for attackers to lock attack targets through IP addresses by constantly changing the IP addresses of protected hosts / servers. Even if the attacker locates the IP address of the attack target, because the IP address will hop in a short time, the attacks launched through the IP address, such as DDoS will also fail in a short time.

Some researchers have proposed adaptive IP hopping defense technologies [2-3]. The basic principle is to deploy a perception engine to sense the behaviour of attackers. It guides IP hopping defense according to the perceived results, including IP hopping configuration and IP hopping trigger. These methods improve the pertinence of IP hopping defense to attack behavior.

The network attack awareness method based on deep learning plays an important role in improving the defense effect of IP address hopping. However, after transplanting the deep learning model to the CPU
platform, its performance decreases significantly. The computing efficiency can be reduced by 20-50 times compared with that on the GPU platform. Moreover, because of its large capacity, the deep learning model occupies much more storage than the general model. Therefore, although the prospect of deploying the deep learning model in adaptive IP hopping defense is promising, its deployment cost and system overhead are also of concern.

This paper proposes to use the model compression technology [4] to reduce the storage occupied by the deep learning model, improve its computing efficiency on the CPU platform, and make it agile for the adaptive IP hopping defense. The proposed technology is applied to the existing two-level IP hopping defense mode to resist scanning attacks and DDoS attacks.

2. Threat model
Our proposed IP hopping defense is mostly focus on scanning attack and DDoS attack. The two kinds of attacks are modeled as follows:

2.1. Scanning attack
Scanning attack in this paper refers to one kind of cooperative scanning attack. This cooperative scanning indicates that a number of hosts act as scanners and sample (probe) IP addresses in the protected network. The whole IP space will be divided and assigned to the scanners, and will be scanned uniformly. The purpose of attacker is testing which IP addresses is currently used online, so that he can prepare for following attacks such as DDoS.

We assume that the scanning attack is start from the outer network but is a propagating scanning. Each newly probed host may get infected and start acting as a scanner. Once a host is probed, it takes time to infect the host. Scanners can share the IP address space that has not been scanned.

2.2. DDoS attack
The defense method proposed in this paper can target DDoS attacks directly using victim's IP address, such as ICMP flood and Smurf attack under ICMP Protocol. In addition, the proposed method can also target DDoS attacks under the upper layer protocol, as long as the attack depends on the IP address of victim, such as SYN Flood under TCP protocol. Table 1 lists three types of DDoS attacks that can be effectively targeted by the proposed defense method. While, obviously, the DDoS attacks that the proposed method can target are not limited to the following three types.

| Protocol | DDoS attack | Description |
|----------|-------------|-------------|
| ICMP     | ICMP Flood  | The attacker bombards the victim server with fabricated ICMP echo request packets from a wide range of IP addresses. |
| ICMP     | Smurf Attack| The attacker forged the ICMP echo-request packets with the victim sever’s IP address and broadcast them to a large range of network with an IP broadcast address. |
| TCP      | SYN Flood   | The attacker bombards the victim sever with fabricated SYN requests containing fake source IP addresses. This attack should be launched with victim sever’s IP address in the socket. |

3. Proposed defense method
The overall idea of the IP hopping moving target defense method proposed in this paper is to use the designed CNN [5] to sense network threats, and use the detection results of CNN to guide IP hopping defense. The overall framework process is shown in Figure 1. Firstly, the data is sampled and the
feature map to be detected is constructed. The data sampled in this method is the flow table data of each switch in the SDN data plane. After sampling, a two-dimensional matrix is constructed to input the subsequent convolutional neural network. Then, a CNN for detecting scanning attack and DDoS attack is proposed, and neural network compression technology is used in the training stage to compress the storage occupied by the CNN and improve the detection efficiency. Finally, the host range (hopping configuration) of IP hopping is determined according to the detection results of the CNN, and IP hopping is triggered also by the detection results of the CNN.

Fig. 1 Overall architecture of IP hopping defense with compressed CNN

3.1. Data sampling and feature construction
In traditional machine learning methods, the features to be detected need to be designed artificially. While in deep learning, deep neural network has the ability to abstract features layer by layer automatically. Therefore, the proposed method is based on the data plane of SDN. Each switch collects the flow table data and uploads it to the detector as the first layer input of deep neural network.

Considering that the detection purpose of CNN in this method is to determine whether the protected host directly connected to a specified switch is under attack, so each sample to be detected should represent the current attack situation of a switch. This method combines the data extracted from the specified switch with the data extracted from the switch directly connected to the specified switch to form the sample to be detected. In addition, in order to facilitate the subsequent processing of the above features using convolutional neural network, the above features are connected end to end to form a one-dimensional feature vector. Then the one-dimensional vector is transformed into a two-dimensional matrix. This process only changes the dimension of the feature, not the value of the feature.

3.2. CNN design and compression
The structure of the complete CNN is shown in Figure 2, which is composed of three convolutional modules and a judgment module. Each convolutional module is composed of a Convolutional layer, a Batch Normalization layer [6], a ReLU layer [7] and a Pooling layer. The judgment module is composed of a Fully-connected layer and a Softmax layer.

The network structure may keep evolving over training course. In another words, networks is able to learn more efficient structure during training. Quantization levels also adapt to the changing of structure. The pruning-quantization operation is performs on each layer and consists of three steps:

1) Clipping. A hyper-parameter $p$ is set to guide a boundary of value range $[c^-, c^+]$. It make sure that $p$ of the positive weights in this layer is less than or equal to $c^+$ while $p$ of the negative weights in this layer is greater than or equal to $c^-$. The weights in $[c^-, c^+]$ will be set to zero in the next forward pass. The corresponding connections will be temporarily removed in the next mini-batch. This operation will be performs again in each next iteration, so the removed connections may playback. The hyper-parameter $p$ is constant while $c^-$ and $c^+$ changes in each iteration.
Fig. 2 Structure of complete CNN

(2) Partitioning. A hyper-parameter \( b \) is set to guide the quantization interval. \( b \) is the storage budget of a weight. After Clipping, weights greater than \( c^+ \) and less than \( c^- \) are partitioned linearly into \( 2^b - 1 \) intervals. There will be totally \( 2^b \) intervals including the zero interval between \( c^+ \) and \( c^- \).

(3) Quantizing. This step determines the discrete values (quantization levels) that the weights are placed to. Each quantization level is equal to the average value of all weights in the corresponding quantization interval. All weights will be quantized to their quantization levels in the next forward pass. The quantizing will also be evaluated again in the next mini-batch.

3.3. IP hopping strategy
This method implements two-level IP hopping defense strategy, namely low-frequency hopping and high-frequency hopping. Each protected host can independently accept the above two levels of hopping. The trigger time of the above two-level hopping of any protected host is determined according to the detection results of CNN.

3.3.1. Two-level hopping. Low-frequency hopping updates the IP address space for the specified protected host. Our method randomly selects address spaces in the current network and allocates it to the host. If the selected IP address space has been occupied by another host, exchange the IP address space between the two hosts. High-frequency hopping changes the IP address for the specified protected host. That is, select an IP address in the current IP address space of the protected host to replace the IP address currently in use. The selection method adopts uniform random distribution and put back sampling.

3.3.2. Hopping triggering. The IP hopping configuration and IP hop triggering in this method are implemented according to the real-time detection results of CNN. The specific triggering method can be summarized as follows: if the flow table sample uploaded to the CNN is detected as malicious, perform high-frequency hopping on all protected hosts directly connected to the specified switch. If
the number of times that the flow table samples of a specified switch uploaded to the CNN being detected as malicious reaches the pre-set threshold, low-frequency hopping will be performed on all protected hosts directly connected to the specified switch.

4. Experiment results

4.1. Scanning attack defense
The evaluation of defense effect by scanning success rate of scanning attack is shown in Figure 4. It can be seen from the data in the figure that the performance of ACNND [3] method and the method proposed in this paper is better than that of classical RHM [1] and SEHT [2] methods in the whole process of scanning attack. The defense performance of the proposed method is slightly better than that of ACNND method in the middle and late stage of scanning attack.

![Fig. 4 Success rate of scanning attack](image)

4.2. DDoS attack defense
As shown in Table 2, our proposed method overcomes SEHT method obviously. However, our proposed method performs slight worse than ACNND method under Smurf Attack and SYN Flood.

| DDoS attack   | Service completion time (s) | SEHT     | ACNND    | Proposed method |
|---------------|-----------------------------|----------|----------|-----------------|
| ICMP Flood    |                             | 2.49s    | 1.43s    | 1.39s           |
| Smurf Attack  |                             | 6.01s    | 2.11s    | 2.25s           |
| SYN Flood     |                             | 4.12s    | 1.74s    | 1.81s           |

4.3. System overhead
The storage occupation of the model in this paper is slightly higher than that of SEHT, but it has obvious advantages over ACNND method. In terms of processing efficiency, the proposed method is better than both ACNND method and SEHT method. In terms of channel occupation, the proposed method is higher than SEHT method, but lower than ACNND method.

|                      | Storage occupation | Processing efficiency | Channel occupation |
|----------------------|--------------------|-----------------------|--------------------|
| SEHT                 | 419 KB             | 0.37852 s             | 91 kb/s            |
| ACNND                | 1.51 MB            | 0.99577 s             | 137 kb/s           |
| Proposed method      | 460 KB             | 0.29257 s             | 974 kb/S           |
5. Conclusion
In order to reduce the system overhead of IP hopping defense on the premise of ensuring security performance, a lightweight IP hopping defense technology based on compressed convolutional neural network is proposed in this paper. Using neural network compression technology, the processing efficiency is greatly improved, the occupation of storage is reduced, and the channel occupation is reduced, while the defense effect is mostly retained.

References
[1] Jafarian J H, Al-Shaer E, Duan Q 2012 Openflow random host mutation: transparent moving target defense using software defined networking Proceedings of the first workshop on Hot topics in software defined networks pp 127-132
[2] Lei C, Zhang H, Ma D, et al. 2017 Network Moving Target Defense Technique Based on Self-Adaptive End-Point Hopping Arabian Journal for Science & Engineering Springer Science & Business Media BV 42(8)
[3] Xu X, Hu H, Liu Y, et al. 2021 An Adaptive IP Hopping Approach for Moving Target Defense Using a Light-Weight CNN Detector[J]. Security and Communication Networks
[4] Tung F, Mori G 2018 Deep neural network compression by in-parallel pruning-quantization IEEE transactions on pattern analysis and machine intelligence 42(3) pp 568-579
[5] Kim P 2017 Convolutional neural network Matlab Deep Learning Berkeley pp 121-147
[6] Schilling F 2016 The Effect of Batch Normalization on Deep Convolutional Neural Networks
[7] Hara K, Saito D, Shouno H 2015 Analysis of function of rectified linear unit used in deep learning International Joint Conference on Neural Networks IEEE