Real-time DDoS Attack Detection Method for Programmable Device*

Haifei MIAO and Guang CHENG

School of Cyber Science and Engineering, Southeast University, Nanjing, 211189, China
Key Laboratory of Computer Network and Information Integration of Ministry of Education of China
Jiangsu Provincial Key Laboratory of Computer Network Technology, Nanjing, 211189, China

hfmiao@seu.edu.cn; gcheng@njnet.edu.cn

***Abstract* - Aiming at the problem that DDoS (Distributed Denial of Service) attacks are difficult to identify in real time with high accuracy and low energy consumption, a real-time DDoS attack detection method based on programmable device OpenBox is proposed. The method adopts a combination of software and hardware. On the hardware side, OpenBox updates the counter when forwarding packets, and supports user-defined hardware actions. It can provide the feature values required for detection at the hardware level. On the software side, it runs a hardware awareness module based on sliding window and an online detection module based on machine learning. The hardware awareness module senses the network status in real time according to the threshold. When the network is abnormal, the online detection module is launched to detect DDoS attack. A DDoS attack detection prototype system based on this method is implemented, and deployed on OpenBox. Experiments show that the method can detect DDoS attack in real time with low resource occupancy and high accuracy.**

Index Terms - DDoS attacks, OpenBox, detection, real time, machine learning

I. INTRODUCTION

Cyber threats have always been an issue that cannot be ignored. Among them, the damage caused by DDoS attack is not to be underestimated. The main purpose of the DDoS attack is to consume the computing resources of the attack target, so that the network service cannot be accessed for a period of time. DDoS attack has become one of the main threats to network security due to large traffic, wide range, and large losses. How to detect DDoS attacks accurately in real time is an urgent problem to be solved.

DDoS attacks are difficult to identify in real time with high accuracy and low energy consumption. The existing DDoS attack detection methods generally have problems such as lack of real-time performance, high resource occupancy rate, and low detection accuracy. In order to detect DDoS attacks accurately in real time and with low energy consumption, this paper proposes a real-time DDoS attack detection method based on programmable device OpenBox.

The paper is organized as follows: the second chapter introduces related works, including related researches and programmable device OpenBox.; the third chapter introduces the real-time DDoS attacks detection method based on OpenBox, which is explained from two aspects: hardware awareness module and online detection module; the fourth chapter introduces the experimental process and analysis of experimental results; the fifth chapter summarizes the full text; and the last chapter is to thank the research funds for their support.

II. RELATED WORKS

A. Related Researches

There are several researches on DDoS attack detection. Shuang Wei et al. [1] proposed a two-stage DDoS detection and defense system called TDSC. In the first stage, the input flows are divided into 4 parts and then cluster size distribution analysis is used to detect DDoS attack. Because cluster analysis is used as the basic detection algorithm, TDSC can easily separate DDoS attacks from the legitimate flash crowd. According to the DDoS attack result output in the first stage, the TDSC can filter the attack traffic in the second stage. Sabah Alzahrani et al. [2] proposed a DDoS detection system using an artificial neural network that detects the occurrence of DDoS attacks by integrating signatures. This method achieves higher accuracy and detection rate. Luyong Zhang et al. [3] proposed a DDoS detection algorithm based on sliding window. The researchers discussed the effect of sliding windows of different sizes on the accuracy of the results, and selected an optimal value as model parameter. Jie Hao et al. [4] proposed a method for detecting DDoS attacks using ANN (Artificial Neural Network). The author used the Turing test method to identify which user accesses the system and records user behavior data, including the resources accessed by the user and the time and time of user access. Attacks are detected by designing intelligent user control systems and training RBF neural networks. Kashyap et al. [5] select the entropy value of the minimum feature set according to different transmission protocols to detect DDoS attacks, so that the traffic processing time is greatly reduced. However, this method is deployed on the victim side and has a large complexity. Nguyen et al. [6] proposed a framework called Anti-DDoS framework. Anti-DDoS can detect DDoS attacks actively by classifying the network status. They use the k-nearest neighbor method to classify the network state into each phase of the DDoS attack. However, the computational cost of real-time detection is high.

* This work is supported by the National Key R&D Program of China under Grant No. 2017YFB0801703, the National Natural Science Foundation of China under Grant No. 61602114, CERNET Innovation Project No. NGII20170406, and the Natural Science Foundation of Jiangsu Province (NO. BK20151416).

Koay et al. [7] proposed a new set of entropy-based features to help detect DDoS attacks accurately, and introduced a new multi-classifier system based on machine learning to improve the versatility and accuracy of detection.

However, these methods are too computationally intensive to detect DDoS attacks in real time, or they can detect attacks in real time but with low accuracy. In order to solve the above problems, this paper proposes a real-time DDoS attack detection method based on programmable device called OpenBox. In this study, the DDoS attack is pre-judged by using the hardware counter on OpenBox. When a suspected DDoS attack occurs, the packets are extracted to the OpenBox software level, and the machine learning module runs to classify immediately. Therefore, DDoS attacks can be detected in real time with low resource occupation and high recognition accuracy.

B. OpenBox: A Programmable Device

OpenBox is a programmable switching device that uses a combination of multi-core CPU and FPGA architecture, and flexible software and hardware functional partitioning mechanism to support the research of next-generation network architecture, new network protocols and packet processing mechanisms. It is an independent network device based on FPGA. Since the FPGA can be flexibly programmed through the Verilog language, the packet parsing rules and forwarding rules in the OpenBox can be arbitrarily modified to become a Layer 2 switch, a Layer 3 router, or a network device for performing any specific task. OpenBox is mainly composed of user space, kernel space and FPGA. Users can write code in user space, control OpenBox's processing of packets, and implement programmable functions.

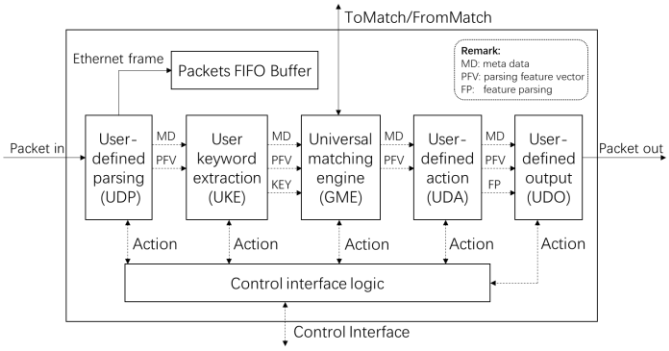

Fig. 1 Hardware Pipeline of OpenBox

The bottom layer of OpenBox consists of five stages of pipelines: user-defined parsing (UDP), user keyword extraction (UKE), universal matching engine (GME), user-defined action (UDA) and user-defined output (UDO). The hardware pipeline is shown in Figure 1. After writing the code in the OpenBox user space, it will compile and run on the platform, and the user's custom action will be inserted into the OpenBox hardware stream.

To support this study, we used the Verilog language to program the FPGA so that OpenBox can provide some of the features required for DDoS attack detection from the hardware level. In summary, OpenBox provides a hardware foundation for this study, enabling real-time DDoS attack detection.

III. DDoS Attack Detection Method Based on OpenBox

In order to reduce resource utilization and improve accuracy, this paper proposes a real-time DDoS attack detection method based on OpenBox. The method consists of two modules: hardware awareness module based on sliding window and online detection module based on machine learning. Both modules run in the OpenBox user space. The hardware awareness module can sense the network status in real time by interacting with the hardware counter. After the network abnormality is sensed, the online detection module extracts the features for re-identification. Based on machine learning technology, it can detect DDoS attacks with high accuracy.

A. Hardware Awareness Module Based on Sliding Window

In order to reduce the impact of attack detection on the normal use of the network, the module reads the hardware counter in advance to sense the network status. A DDoS attack generally displays the following traffic characteristics: a large number of different sources simultaneously access the same destination in a short period of time; the destination cannot handle such a large number of requests; thus, packet loss occurs. Therefore, the number of received packets on the router is much larger than the number of packets sent, and the router port may be inequitable or the traffic is increased. With such features, network state eigenvalues can be provided at the hardware level by programming the FPGA. A software program can be run in the OpenBox user space to read the hardware counters to detect network anomalies in advance with minimal resource cost.

This paper uses byte rate as a primary feature to detect DDoS attacks. In general, the byte rate in a normal network stream will be relatively stable. Although there will be some subtle fluctuations, in terms of statistical regularity, the flow is still relatively stable. When a host experiences a DDoS attack, the byte rate grows in a short period of time, quickly reaching a large order of magnitude.

The hardware awareness module is improved on the DDoS attack detection method based on sliding window which was proposed by Luyong Zhang et al. [2]. The main idea of sliding window is to calculate the eigenvalues in a particular time sequence and to slide rearward by period. The main purpose of using the sliding window method is to monitor whether the entire network environment is relatively stable, and whether there is sudden traffic.

The definition of the sliding window is as follows: Suppose there is a time sequence $X_1X_2...X_{t-1}X_t$, (the current time is t, and the window length is n), then at time t, the window is $[X_{t-n+1}X_{t-n+2}...X_{t-1}X_t]$, and at time t+1, the window becomes $[X_{t-n+2}X_{t-n+3}...X_tX_{t+1}]$. Then at time t, the average byte rate is calculated as shown in equation 1.

$$f_t = \frac{1}{n}\sum_{t-n+1}^{t} X_i \tag{1}$$

The definition of the sliding window is as follows: Suppose there is a time sequence $X_1X_2...X_{t-1}X_t$, (the current time is t, and the window length is n), then at time t, the window is $[X_{t-n+1}X_{t-n+2}...X_{t-1}X_t]$, and at time t+1, the window becomes $[X_{t-n+2}X_{t-n+3}...X_tX_{t+1}]$. Then at time t, the average byte rate is calculated as shown in equation 1.

Since OpenBox can provide the total number of bytes received at current time t from the hardware level in real time, represented by Y_t, formula 1 can be directly converted to equation 2. The improved formula requires less computation and eliminates the need to collect packets, so it has little impact on the network environment.

$$f_t = \frac{1}{n}(Y_t - Y_{t-n+1}) \quad (2)$$

According to the idea of the sliding window threshold, if $f_{t+1} > (1+b) f_t$, where the threshold b is taken as 0.5, then the hardware awareness module will conclude that the network has experienced abnormal fluctuations.

In summary, the hardware awareness module based on the sliding window is able to sense the network status in real time based on the threshold. Once a network anomaly is detected, the method will modify the hardware rules to change the flow of the message and forward it to the online detection module. In this module, DDoS attacks will be detected using machine learning techniques.

B. Online Detection Module Based on Machine Learning

1) DDoS attack detection mechanism:

The DDoS attack generally has a significant difference from normal traffic in terms of the number of packets, the number of source and destination ports, the number of source and destination IPs, and the ratio of errors generated by the destination. This module uses machine learning to detect DDoS attacks. Machine learning can be used to distinguish between normal traffic and abnormal traffic by analyzing a large amount of historical data, extracting feature values, and modelling.

In order to detect DDoS attacks in real time, this paper chooses to use 2 seconds as a cycle. After the packets flows in, the module analyzes the traffic situation of each cycle, extracts the feature values in each cycle, forms a feature vector, and performs real-time DDoS attack detection by the trained DDoS attack recognition program.

2) Features and Classifier Selection:

DDoS attack detection mainly has two categories: protocol analysis and traffic model analysis, but each has its own limitations. When selecting features in this paper, we select features based on these two categories, which can combine the advantages of the two types of DDoS attacks detection methods and have better generalization ability. This study also considers the characteristics of real-time and resource occupancy, using the recommendations of Karimazad and Faraahi [8], and selected 11 features as shown in Table 1 for DDoS attack detection and classification. Among them, eight features are statistical features, which can be obtained directly from the hardware counters on OpenBox without calculation. The other three features are computational features, which require a little calculation in the user space of programmable device OpenBox.

The classification algorithm selected in this paper is SVM (Support Vector Machine) [9]. The reason for choosing this algorithm is that the classifier built using support vector machines can simultaneously minimize empirical errors and maximize geometric edge regions. In terms of precision and recall, support vector machines provide better performance than traditional methods.

TABLE I
FEATURES OF THE DDOS DETECTION METHOD

Feature name	Type	Description
protocol_type	Computational	Protocol with the largest number of packets
src_bytes	Statistical	Average bytes from source to destination
dst_bytes	Statistical	Average bytes from destination to source
flag_count	Computational	Packet numbers where SYN, ACK, PSH, URG or RST is 1
src_ip_count	Statistical	Different source IP numbers
packet_length	Computational	Average packet length
packet_count	Statistical	Number of packets per unit time
tcp_packet_count	Statistical	Number of TCP packets
tcp_src_port_count	Statistical	Number of source ports in TCP packets
tcp_dst_port_count	Statistical	Number of destination ports in TCP packets
tcp_fin_flag_count	Statistical	Number of TCP packets where FIN is 1

The SVM classifier seeks to separate hyperplanes that produce the largest separation boundary. This approach is known to be associated with structural risk minimization. In a more general case, the data points in the input space are not linearly separable; before applying the linear maximum edge classifier, the data vector x is mapped to a high dimensional space which is called a feature space using a nonlinear transformation. To avoid potential defects in over-fitting in this high-dimensional space, the SVM uses a kernel function in which the nonlinear mapping is implicitly embedded. As long as the condition is met, the function can be used as a kernel function. By using the kernel function, the decision function in the SVM classifier has the following form, as shown in Equation 3.

$$f(x) = \sum_{i=1}^{N_s} a_i y_i \Phi(s_i)\Phi(x) + b = \sum_{i=1}^{N_s} a_i y_i K(x_i, x') + b. \quad (3)$$

Where, $K(x_i, x')$ is a kernel function, x_i represents a so-called support vector determined from the training data, N_s is the number of support vectors, and y_i is a category indicator (e.g. indicator '+1' represents the first category and indicator '-1' represents the second category) associated with each x_i, a_i is a constant, also determined by training, and b is the threshold in the SVM.

In the software developed in this experiment, we selected the open source support vector machine library 'LibSVM',

which is provided by scholars Chih-Chung Chang and Chih-Jen Lin [10].

C. DDoS Attack Detection Prototype System

In order to verify that the DDoS attack detection algorithm proposed in this paper can detect DDoS attacks with low energy consumption and high accuracy, a prototype DDoS attack detection system is developed. We wrote software and hardware code on the programmable device OpenBox, and implemented the DDoS attack detection prototype system according to the algorithms in Parts A and B of Chapter III, which can run on OpenBox.

The prototype system consists of three modules: the FPGA module, the hardware awareness module, and the online detection module. The FPGA is first programmed as required, allowing OpenBox to simultaneously count the packets as they are forwarded, providing the count-type eigenvalues required in Parts A and B of Chapter III. Then we write the software code. The hardware awareness module reads the hardware counter of OpenBox on a periodic basis, and judges whether the network has fluctuated according to the threshold. The online detection module starts after the hardware awareness module recognizes the abnormal network fluctuation, extracts the message characteristics, and uses the trained classifier model to identify whether it is a DDoS attack. The flow chart of the prototype system is shown in Figure 2.

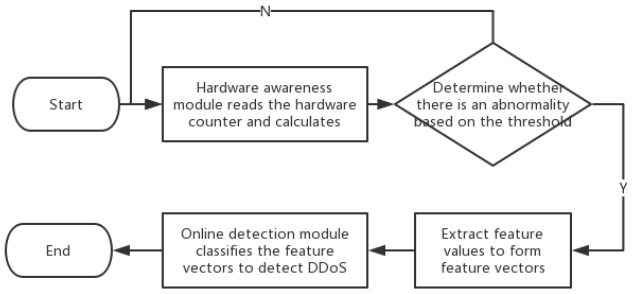

Fig. 2 Flow Chart of the Prototype System

We have also developed a training system for the SVM model, which can be used to manipulate OpenBox for data acquisition and model training. Due to limited space, it will not be repeated here.

IV. Experiment and Analysis

A. Experimental environment

In this experiment, OpenBox is used as the switch, and the PC is used as the terminal to test and detect the DDoS attacks. OpenBox is based on Intel Atom CPU and Altera arria V FPGA. The hardware configuration is Intel J1900 CPU, 4G RAM, 32G ROM. All three PC configurations are Intel i7 3770 CPU, 8G RAM, running Ubuntu 16.04 operating system. In addition to being connected to terminals H1 and H2, the programmable switch OpenBox is also connected to the controller. The controller can configure a flow table, issue hardware rules, train classifiers offline, issue classifier models, and control DDoS attack generators. The experimental topology diagram is shown in Figure 3.

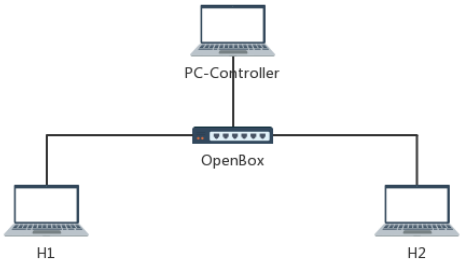

Fig. 3 Experimental Topology Diagram

On H1, the DDoS attack traffic generation software traffic-generator (TG) is used to simulate DDoS attacks to H2. Both H1 and H2 run common software such as iPerf to simulate normal traffic. The DDoS attack detection prototype system for programmable device implemented in Part C of Chapter III runs on OpenBox.

B. Experimental content

Firstly, a large amount of normal traffic and DDoS attack traffic such as TCP flood, UDP flood, and ICMP flood were generated by the TG program. The data collection was run for 4 days. Each type (Normal, TCP flood, UDP flood, ICMP flood) was collected for 24 hours, and data was collected every 2 seconds. A total of 172,800 data were collected. Then the SVM classifier was trained on the controller, and the kernel function used radial basis kernel function that had been verified to be the best for classification [11]. After the training was completed, the controller updated the classification model to OpenBox.

The experiment was then carried out. First, when normal communication was performed between two machines, such as establishing a TCP Socket to transfer data, and using iPerf to generate a background stream, the classification result was always displayed as 0, which means no attack was detected. After the TCP SYN Flood attack was generated by the DDoS attack generator, the classification result was displayed as 1 which means the TCP SYN Flood attack was detected. After the DDoS attack type was adjusted, for example, when the UDP Flood attack was performed, the classification program in OpenBox displayed the result as the corresponding code.

This experiment monitored the real-time performance of DDoS attack detection and identification after the attack was initiated. The experiment found that the detection system can determine the DDoS attack type in 1-2 cycles, that is, 4 seconds, and print the attack type through the console. We tested each attack type for 10 minutes with a period of 2 seconds to calculate the identification accuracy. The accuracy of the identification of different attack types by the real-time DDoS attacks detection program based on OpenBox is shown in Table II.

TABLE II
Results of the DDoS Attacks Detection Experiment

Attack Type	Total cycles	Cycles with Correct Identification	Accuracy
SYN Flood	300	289	96.3%
UDP Flood	300	282	92%
ICMP Flood	300	286	95.3%

It can be seen from Table II that the method can accurately identify three different types of DDoS attacks: TCP SYN Flood, UDP Flood, and ICMP Flood. Among them, the recognition rate of the TCP SYN Flood attack is the highest, reaching 96.3%. It provides a prerequisite for the subsequent cleaning of DDoS attack traffic.

At the same time, we noticed that the online detection consumes a large amount of system resources for the switching device. Therefore, this paper used a combination of hardware awareness and machine learning online detection to detect DDoS attacks and reduced resource utilization. We also monitored the resource usage of the programmable device OpenBox in the experiment. The resource usage of the switching device is monitored. The CPU and RAM usage curve of OpenBox during detection is shown in Figure 4.

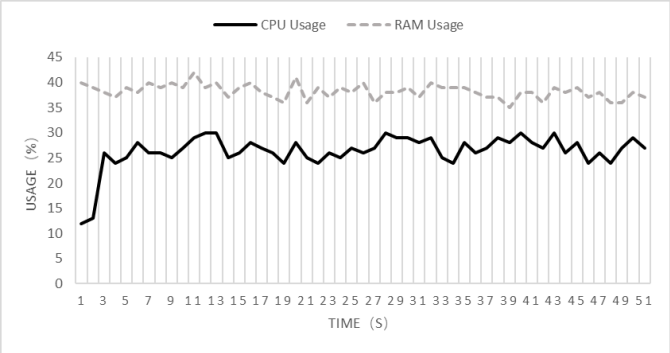

Fig. 4 CPU and RAM usage curve of OpenBox

As can be seen from Figure 4, the CPU is only used 10% when the system is running. When the online DDoS attack is started, the CPU usage is only about 25%, and the RAM is stable between 36% and 42%. It can be seen that the real-time DDoS attack detection method proposed in this paper has the characteristics of occupying small system resources.

V. Conclusion

This paper proposes a real-time DDoS attack detection method based on OpenBox. The method uses a combination of software and hardware for detection. Firstly, the hardware awareness module is used to read hardware counters to sense the network status, which consumes less resources. After detecting a possible network anomaly, the online detection module uses machine learning techniques to detect DDoS attacks. Since the programmable device OpenBox can provide most of the features from the hardware level, the amount of calculation is greatly reduced, real-time detection is realized, and the accuracy is not lowered. Experiments show that this method can effectively detect DDoS attacks in real time.

Although DDoS attacks can be effectively detected with low consumption and high accuracy, how to clean DDoS attacks is still a problem. Next, we will continue to research how to quickly and efficiently clean DDoS attacks based on the programmable device OpenBox.

Acknowledgment

This work was supported by the National Key R&D Program of China under Grant No. 2017YFB0801703, the National Natural Science Foundation of China under Grant No. 61602114, CERNET Innovation Project No. NGII20170406, and the Natural Science Foundation of Jiangsu Province (NO. BK20151416). We thank for their supports.

References

[1] Wei S, Ding Y, Han X. TDSC: Two-stage DDoS detection and defense system based on clustering[C]//Dependable Systems and Networks Workshop (DSN-W), 2017 47th Annual IEEE/IFIP International Conference on. IEEE, 2017: 101-102.

[2] Alzahrani S, Hong L. Detection of Distributed Denial of Service (DDoS) Attacks Using Artificial Intelligence on Cloud[J]. 2018:35-36.

[3] ZHANG L, Ming Q, CHI Y. DDoS Attack Detection Using Sliding Window Method[J]. DEStech Transactions on Computer Science and Engineering, 2017 (wcne).

[4] Siaterlis C, Maglaris V. Detecting incoming and outgoing DDoS attacks at the edge using a single set of network characteristics[C]//Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on. IEEE, 2005: 469-475.

[5] Kashyap H J, Bhattacharyya D K. A DDoS attack detection mechanism based on protocol specific traffic features[C]// Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology. 2012.

[6] Nguyen H V, Choi Y. Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDoS framework[J]. International Journal of Electrical, Computer, and Systems Engineering, 2010, 4(4): 247-252.

[7] Koay A, Chen A, Welch I, et al. A new multi classifier system using entropy-based features in DDoS attack detection[C]//Information Networking (ICOIN), 2018 International Conference on. IEEE, 2018: 162-167.

[8] Karimazad R, Faraahi A. An anomaly-based method for DDoS attacks detection using RBF neural networks[C]//Proceedings of the International Conference on Network and Electronics Engineering. 2011, 11: 44-48.

[9] Meyer D, Wien F H T. Support vector machines[J]. R News, 2001, 1(3): 23-26.

[10] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM transactions on intelligent systems and technology (TIST), 2011, 2(3): 27.

[11] Buhmann M. Radial basis function[J]. Scholarpedia, 2010, 5(5): 9837.