DDoS Detection and Protection Based on Cloud Computing Platform

Tianwen Jili and Nanfeng Xiao*
School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
Email: 422289391@qq.com; xiaonf@scut.edu.cn

Abstract. Currently, the distributed denial service attacks (DDoS) are increased rampantly on the internet, the threshold of such attacks is relatively low for the malicious attackers. Therefore, the DDoS attacks are serious threats to the security availability of the cloud computing. Aiming at the threats, this paper studies the malicious traffic identification and the detection scheme of the denial service attack under the software-defined Network (SDN), which uses the SDN forwarder to distinguish the DDoS attack traffic and adopt the corresponding filtering means to achieve protection for the distributed denial service attack. This paper firstly implements cloud platform resource calls, then an attack detection technology based on information entropy is proposed and implemented to carry out the DDoS attack detection, because the size of the entropy value can show the discrete or aggregated characteristics of the current data set, which can be used to detect abnormal data traffic. At last, the experiments are also carried out to verify and analyze the effectiveness of the DDoS attack detection and the protection methods.

Keywords. Cloud computing; software-defined network (SDN); distributed denial service attack; attack detection.

1. Introduction
The concept of Software Defined Network (SDN) is a new type of network architecture that was first proposed by Clean Slate Research Group in Stanford University in 2008, it uses the leveling idea to divide the forwarder and the control functions of the traditional network into three separate planes, which simplifies the network management and solves the problems of high network configuration complexity [1]. But at the same time, the SDN also faces security issues, the most direct and the major threat is the distributed denial service attack. Because there is a controller for overall network traffic management in the SDN environment, the controller of the SDN is highly vulnerable under the DDoS attacks, which makes the entire network is out of control and embarrassing. The features of the denial service attack as a popular and low-threshold attack means that the attackers using such attacks often have a wide range destruction. Therefore, the DDoS detection and protection in the cloud computing environments are urgently needed to be solved.

The SDN controller can make the computer network management more convenient. At the same time, the universal interface provided to the cloud computing platform makes the network resource scheduling more efficient [1], but the risk brought by this model makes it is easier to become the target of the
distributed denial service attack. Therefore, considering the vulnerability of the single-point failure in the SDN, there are many DDoS attack detection methods based on the information entropy, and the DDoS attack detection based on the various classification algorithms, etc. [2], the detection and the protection still have not a perfect solution so far. Therefore, this paper proposes a feasible and effective method by combining various known protection approaches.

2. Related Technology

2.1. Cloud Computing Overview [3, 4]
As shown in figure 1 the cloud computing services are divided into the three levels: infrastructure as a service (IaaS), buted platform as a service (PaaS), software as a service (SaaS) [5], and the deployment methods of the cloud computing are [6]: private cloud, community cloud, public cloud, hybrid cloud. The weakness of the cloud computing makes the DDoS attacks often may get good results, therefore the defense way has always been the security focus. How to deal with the DDoS attacks in the current cloud computing environment needs to take into account the technical details related to the network architecture, and the software-defined networks are the key, therefore most of the process strategies are based on this.

![Figure 1. Cloud computing infrastructure.](image)

2.2. Software Defined Network [7]
As shown in figure 2, the software-defined network separates the control plane of the routing node from the data plane, and implements it in the form of the software coding [8]. This kind of the architecture enables the network manager to change the configuration equipment without the cumber-some, and the SDN controller centrally and uniformly processes the network requirements and the tasks related to the traffic distribution [9].

OpenFlow is a network communication protocol at the data link level, which plays an important role in the path selection and the flow control of the network data packets. The OpenFlow is also the first widely used data control level interface protocol, as shown in figure 3. Under the OpenFlow protocol, the controller can open up a dedicated channel to communicate with the underlying switch. This channel uses the SSL (Secure Sockets Layer) protocol for the encryption to ensure the security of the information transmission.
2.3. Distributed Denial Service Attack
The DDoS attacks have been deeply studied by both the attackers and the defenders. As shown in figure 4, the DDoS attackers firstly use the existing system vulnerabilities to control a group of the zombie machines on the network to form a botnet, and then use the botnet to send a large amount of the request data package to the target. The package establishes an incomplete connection according to the corresponding service protocol to occupy the band width of the server, so that the server can not respond to the request of the normal users [10-12].

![SDN system](image1)

**Figure 2.** SDN system.

![Flow table entry structure](image2)

**Figure 3.** Flow table entry structure.

![DDoS attack model](image3)

**Figure 4.** DDoS attack model.

3. Implementation of Detection and Protection Scheme for DDoS Attacks
This section will build a small virtual cloud computing environment. It is based on the infrastructure provided by the OpenStack, an open source cloud platform, which can simulate the infrastructure in real-world cloud, including control nodes, computing nodes, storage nodes, virtual leased users, system software, such as machines and data-bases. Then, the experimental data set of the network intrusion
detection is injected into the established environment, and the flow controllers maintained by the relevant controllers and the virtual switches in the under-lying network architecture of the cloud computing platform are used to collect the characteristics of the network traffic, and the script is written through the previous one. The algorithm proposed by this paper distinguishes the traffic, draws the dynamic graph of the network traffic entropy by using the related drawing tools, and visually verifies the feasibility of the scheme by comparing the labels of the experimental data sets with the time points of the special changes such as the steep increase.

3.1. Construction of Cloud Computing Platform

This section uses OpenStack, an open source software developed by NASA and Rackspace [13-19], which is primarily composed of several components that work together to meet the needs of virtually all kinds of the cloud environments, providing the infrastructure as a service (IaaS), designed to provide the software support for the construction and the management of the public and the private clouds, enabling both can create and provide the cloud computing services on their own. The OpenStack has a rich community ecosystem and the numerous technology developers.

In order to simulate the real network, using Fuel as a tool to assist in the construction of the cloud computing platform, the goal of the Fuel is to deploy the environment of the OpenStack, and provide the log real-time services and the key business health check related functions. The Fuel has the following advantages [20]: (1) support new nodes in the autonomous discovery environment; (2) can easily set node information; (3) support Linux distribution series; (4) provide programming interface for programmers; (5) can realize dynamic addition and deletion of the nodes; (6) address division and resource allocation scheduling are intuitive and convenient; its main structure is shown in the following figure 5.

![Fuel installation mode](image)

Figure 5. Fuel installation mode.

The Fuel provides the developers with a web user interface and the command line interface for the rapid configuration and the management of the OpenStack environments. The architecture nodes are divided into the master nodes and the slave nodes. The master node is the server used to install the Fuel application and
executes. From the initial configuration of the nodes, the address assignment and the other tasks, the slave node is subdivided into the computing nodes, the storage nodes, and so on. The virtual machine of each node and the main node of the Fuel are created through VirtualBox. The core node supports the core functions of the entire cloud environment. Table 1 lists the environment on the cloud computing platform. The topology of the entire cloud computing platform in this environment is shown in figure 6. The private cloud architecture built by the OpenStack contains eight virtual machines, which are used as the computing and the storage node resources of the cloud platform system. The four virtual machines are set up as the botnets on the periphery, and the zombie host runs the attack script on one of the nodes. To attack, the botnet topology is shown in figure 7.

Table 1. Node configuration.

| Parameters                  | Number |
|-----------------------------|--------|
| Control node                | 1      |
| Control node CPU            | 1      |
| Control node memory/G       | 2      |
| Computing node              | 2      |
| Computing node CPU          | 2      |
| Computing node memory/G     | 2      |

Figure 6. Cloud system topology.
In addition, among the computing (storage) nodes, the DDoS attack detection agent is deployed, and the proxy script collects the current traffic information according to the flow table maintained by the underlying switch of the network within a fixed sampling period, according to the previous section. The calculation method calculates the conditional entropy and the threshold. After the calculation, the window slides forward and starts the next sampling period, so that it can be used to monitor the traffic entropy change of the current network environment in real time. According to the DDoS attack detection algorithm proposed before, if an attack occurs, a warning is output to the control node. The command line interface of the Fuel master node is shown in figure 8. In addition, the web console of the Fuel main node is shown in figure 9.

Figure 7. Botnetwork topology.

Figure 8. Command line interface.
During the operation of the whole system, if a DDoS attack is performed on one of the hosts in the cloud environment (see the attack mode shown in figure 4), the conditional entropy image drawn by the collected data in this time range is changed as follows, which can be clearly observed, as shown in figure 10. The conditional entropy increases sharply between 90s and 150s and is maintained within a larger interval, therefore it can be concluded that there is suspicious DDoS attack traffic in the current network.

3.2. Introduction to Network Traffic Data Set
In this section, KDD-CUP99 is used as the sample data set. The network traffic attack detection data set is a network data sample collected by simulation on the US Air Force LAN. The sample is divided into the training data with identification and the unmarked test data. The training data and the test data correspond to the different probability distributions, which makes the detection of the traffic attacks such as denial service attack more practical [21].
3.3. DDoS Attack Detection Script
This section describes the encoding process of the attack detection script deployed on the user's virtual machine. The script is completed in Python. The following is a detailed description of each key functional module. The most critical part of the whole script is the statistics of the information entropy. For each packet information, such as the source address, the destination port number is collected by the OpenVSwitch switch flow table under the SDN, the OpenVSwitch is an SDN system. With the OpenFlow as the virtual switch of the underlying protocol [22], the real-time maintenance flow table is shown below.

4. Experiments

4.1. Determination of the Second Threshold \( \lambda \)
The first step of the experiment is to count the traffic entropy of each machine in a normal network environment, and estimate the value of the second threshold \( \lambda \). When no attack occurs, the flow of the four virtual machines in one computing node can be calculated according to the following table. The entropy is maintained within a stable range. According to the statistical data, the second threshold \( \lambda \) can be initially determined to be 0.7, therefore it can be concluded that the discriminant of the DDoS attack in the current network is:

\[
H(sip \mid dip) \leq \mu + \lambda
\]  

where \( \mu \) is the average conditional entropy calculated under the normal flow. \( H(sip \mid dip) \) is the conditional entropy obtained in the previous window, as shown in figure 11.

4.2. DDoS Attack Simulation
The second step of the experiment is to use the KDD-CUP99 10% sample data to train the classifier. Firstly, the data set needs to be preprocessed. The pre-processed feature vector is input as a sample into the Bayesian classifier, and then the test data set is classified and detected. Firstly, it is need to preprocess the data set. The data for training and testing is shown in table 2.
Table 2. Training and test results.

| Training sample          | Kddcup data_10_percent       | Sample number | 494021 |
|--------------------------|-----------------------------|---------------|--------|
| Test sample              | Kddcup newtestdata_10_percent_unlabled | Sample number | 311079 |
| Training completeness    | 1                           |               |        |
| Test accuracy            | 83.71%                      |               |        |

The third step of the experiment is to detect the actual attack by configuring the attack script to the virtual machine in the botnet. The script used for the DDoS attack is TFN2K. The operation interface is shown in figure 12. After the TFN2K is compiled successfully, in addition to the tfn main program providing the interface selection for the attacker, the td daemon process is also generated for distribution to each zombie host to accept the attacker's dispatching command, and the host command to the target reaches the botnet. Each td agent performs various types of the DDoS attacks, and its attack demonstration is shown in figure 13 and table 3. The virtual machine configuration in the botnet is as follows:

![Figure 12. Tool TFN2K operation interface.](image)

Table 3. Zombie host attack configuration.

| Number of zombie hosts | 4 |
|------------------------|---|
| Target virtual machine | 1 |
| Type of attack         | Icmp echo flood |
| Attack rate            | 30000 parcets/s |
| Zombine internet address | 192.168.45-192.168.49 |

A protection script for the virtual machine is attacked in the computing node. It can be seen that when the DDoS attack is initiated, the virtual machine outputs a warning to the terminal as shown in figure 14.
In addition, the log file that records the data in the script can obtain the value of the current feature entropy vector and the threshold value at each time point, as shown in figure 15.

![Figure 14. Warning interface of the attacked node.](image1)

![Figure 15. Related data record.](image2)

4.3. Data Analysis

From the experimental data in the previous section, it can be seen that the multi-dimensional conditional entropy is effective for detecting the network traffic, especially in the large-scale network environment. This method is especially prominent for detecting the DDoS attacks. The entropy value of each time period is in line with the expected change. Regularly, but the conditional entropy calculation is more complicated, and there are problems of the performance optimization and the resource consumption. In addition, the
Naomi Bayes’ classification idea can classify the flow with relatively fuzzy conditional entropy, but the effect is not very prominent, and there is still room for the future improvement.

5. Conclusions
The security risks caused by the distributed denial service attack are particularly eye-catching, and therefore attract a lot of discussion and solution. This paper elaborates on how to use existing theoretical foundations to provide the reliable technical solutions in the context of cloud computing. The main work of this paper is as follows: (1) discussing the DDoS attack threats encountered in the cloud computing environment, summarizing the basic theory of the cloud computing and expounding the principle of the DDoS attack in detail; (2) The theoretical principles related to detecting and identifying the malicious traffic technologies are reviewed, including the concept and the calculation method of the conditional entropy and the role played by the traffic monitoring, and the DDoS attack detection algorithm is designed and detailed. The technical details, such as the dynamic changes of the threat interval and the sliding window mechanism, are introduced in detail to lay the foundation for further implementation of the detection system; (3) Using the multi-dimensional condition entropy based on the cloud computing environment, the detection algorithm of the distributed denial service attack is designed and implemented in Python, and the relevant virtual environment is built for software to test the performance of the algorithm, and the test effectiveness was verified.

Acknowledgment
The research in this paper has been funded by the Fundamental and Applied Fundamental Research Fund of Guangdong Province (Project No. 2015A030308018), and the authors express our deep honest thanks.

References
[1] Yan Y 2018 Research and Implementation of Policy Generation System in Private Cloud (Beijing University of Posts and Telecommunications).
[2] Hetian Network Security Lab SDN-Based DDoS Attack Detection and Defense https://segmentfault.com/a/1190000018341766.
[3] Tian F 2012 Design and Implementation of Server Virtualization on China Mobile Public Service Cloud Platform (Tianjin University).
[4] Liu Z, Lin D and Peng Y 2017 Cloud Computing Technology and Application Basis (Beijing: People’s Posts and Telecommunications Press) pp 11-12.
[5] China Institute of Information and Communications 2017 Cloud Computing Key Industry Application Report http://www.chinacloud.cn/upload/17092108184894.pdf.
[6] An Y P Research of Cloud Computing and Private Hybrid Cloud https://3g.163.com/dy/article/E99GVEGS0511DV4H.html.
[7] Fundation O N 2014 Software-defined networking: The new norm for networks Software World 45 (6) 4.
[8] Li H 2015 Research on DDoS Attack Detection Method and Mitigation Mechanism Based on Software-Defined Network (East China Normal University).
[9] Tootoonchian A, Ghobadi M and Ganjali Y 2010 OpenTM: Traffic matrix estimator for OpenFlow networks Proceedings of the 11th International Conference on Passive and Active Measurement (Springer-Verlog) pp 201-210.
[10] Ambrosin M, Conti M, De Gaspari F, et al. 2015 Lineswitch: Efficiently managing switch flow in software-defined networking while effectively tackling dos attacks Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security ACM pp 639-644.
[11] Xia B 2015 Detection and Protection of DDoS Attacks in WLAN Based on Software-defined Network (Shanghai Jiaotong University).

[12] Sharma S, Sahu S K and Jena S K 2015 On selection of attributes for entropy based Detection of DDoS International Conference on IEE Advances in Computing, Communications and Informatics (ICACCI).

[13] Wangtrend Practical Experience of Docking A Large Bank SDN with OpenStack Cloud Platform http://www.talkwithtrend.com/Article/242819.

[14] Apiecionek L and Makowski W 2016 Firewall rule with token bucket as a DDoS protection tool IEEE International Scientific Conference on Informatics pp 32-35.

[15] Navaz A S S, Sangeetha V and Prabhadevi C 2013 Entropy based anomaly detection system to prevent DDoS attacks in cloud eprint Arxiv 62 (15) 42-47.

[16] Osanaiye O, Choo K K and Dlodlo M 2016 Distributed service denial (DDOS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework Journal of Network and Computer Applications 67 (C) 147-165.

[17] Entropy https://zh.wikipedia.org/wiki/%E7%86%B5.

[18] Naive Bayes Classification https://en.wikipedia.org/wiki/%E6%9C%B4%E7%B4%A0%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%88%86%E7%B1%BB%E5%99%A8.

[19] OpenStack https://zh.wikipedia.org/en/OpenStack.

[20] Blog Garden Fuel Quickly Install OpenStack https://www.cnblogs.com/dongdongwq/p/5627532.html.

[21] Jiang J, Liu Z and Xie X 2011 Intrusion detection model based on fuzzy neural network integration Journal of Shandong University (Natural Sciences) (09) 98-101.

[22] OpenVSwitch http://www.openvswitch.org/.