Development of the Algorithm for Protection against DDoS-Attacks of Type Pulse Wave

Ilya V. Chugunkov, Leonid O. Fedorov, Bela Sh. Achmiz, Zarina R. Sayfullina
Department of Computer Systems and Technologies
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Moscow, Russia
Ilya.V.Chugunkov@ieee.org

***Abstract*— Protection from DDoS-attacks is one of the most urgent problems in the world of network technologies. And while protect systems has algorithms for detection and preventing DDoS attacks, there are still some unresolved problems. This article is devoted to the DDoS-attack called Pulse Wave. Providing a brief introduction to the world of network technologies and DDoS-attacks, in particular, aims at the algorithm for protecting against DDoS-attack Pulse Wave. The main goal of this article is the implementation of traffic classifier that adds rules for infected computers to put them into a separate queue with limited bandwidth. This approach reduces their load on the service and, thus, firewall neutralises the attack.**

Keywords— network; DDoS; DPI

I. INTRODUCTION

Preventing network attacks is one of the most difficult tasks in the field of information systems protection. Most modern systems have a distributed structure, their architecture is based on the use of network technologies. And ensuring the operability of such systems depends on the ability to resist malicious acts that are aimed at disrupting the work of both the network itself and the information system functioning within its framework. One of the most dangerous types of criminal activities on the Internet are the so-called DDoS-attacks. The methods used by criminals are constantly evolving and improving, from single attempts they go to corporate development. At the same time, modern systems for detecting intrusions and attacks are far from perfect and insufficiently effective from the point of view of security decisions. Therefore, the methods of work in this direction are necessary and relevant.

II. DESCRIPTION OF THE PROBLEM

Over the past few months, we have witnessed the emergence of a new attack model, rapidly gaining popularity among hackers. She was given the title “Pulsovaya Wave”. Such a hacker attack, conducted on the computer system, is characterized primarily by short, repetitive, at regular intervals, pulses whose peak power can reach 350 Gbit/s. This attack is illustrated in Fig. 1 [1].

Fig. 1. Example of DDoS attack Pulse Wave

Such an attack can last a very long time. Pulse Wave attack has a number of advantages. First, so attackers can attack multiple targets at once. When the impulse stops, and a short lull sets in, the botnet does not stand idle, but attacks another target. Secondly, such attacks are extremely inconvenient for standard DDoS protection systems , which are based on hybrid protection techniques, that is, the first defense line is hardware on-premise products, and the second layer of protection is cloud solutions. The fact is that each impulse disconnects the equipment of the target company. To restore performance after one peak, this attack takes several minutes, but the first pulse is followed by the second, third and so on. This allows the attacker to stretch the DDoS attack for a long time and interferes with the correct operation of the security solutions. Thus, the hardware solution have neither the time nor the bandwidth to request aid cloud and server “crashes”.

III. MODELING ATTACK

For the study of DDoS-attacks “Pulse Wave” protection and development has created a model of the network consisting of a server, the user, the attacker and the gateway router. The server is located in the Net 2 has a static IP = 10.0.0.1 and is responsible for processing requests and come for logging information about them. The server is written in language python using the framework flask. Users and Attackers are on the second network Net 1. Their IP are generated randomly from the address pool 11.0.0.0/24 . Also there is A router connecting these two networks and performing the function of a firewall. The scheme of the network model is shown in Fig. 2.

978-1-5386-4340-2/18/$31.00

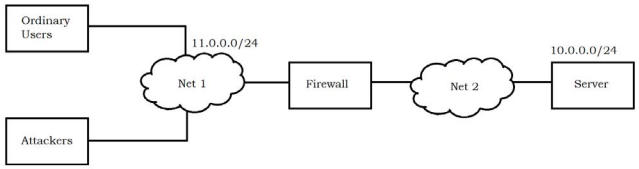

Fig. 2. Diagram of the model networks

The normal operation of the server was simulated, by sending to it the get-requests for the HTTP protocol. For this, python was used library scapy, allowing rd it's quite simple and easy to configure the sending of the appropriate type of request. In this case, the load on the processor was measured. The result of the program is shown in Fig. 3.

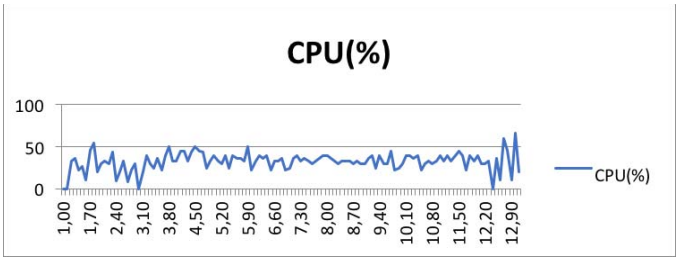

Fig. 3. CPU load during normal server operation

As a result of testing, the server load was 30-40% on average. On the graph, there are minor deviations in the readings from the mean, due to the randomness of the values of the delay times between the sent requests. The maximum indicator does not exceed 67%.

IV. Implementation of the Attack

Then the attack on the server was simulated. To simulate a large number of attacking bots these python modules have been used in the network, as a subprocess and multiprocessing, with the help of which a copy of the process simulating the attack of the botnet nodes on the server was created. The results of the attack presented are graphs of CPU load and the number of unique IP addresses of senders (Fig. 4-5).

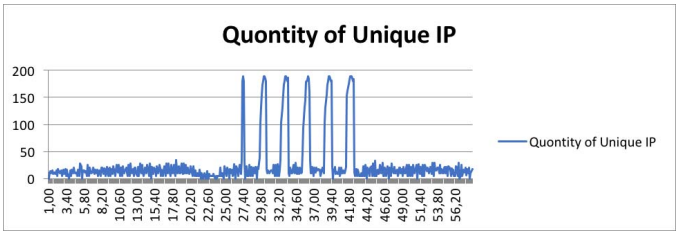

Fig. 4. The number of unique IP from time

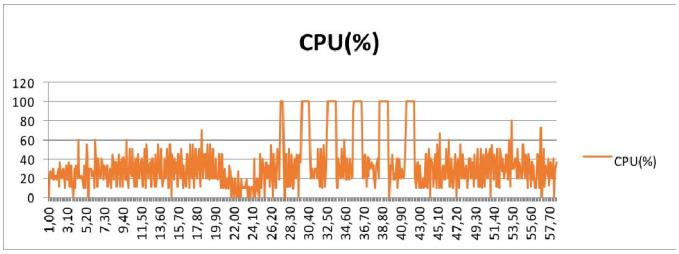

Fig. 5. Load on the processor from time to time

At the moment of the beginning of the attack, the network sharply increases activity, the number of unique IP per unit time, it is these IP it was decided to mark suspicious. A sharp increase in the number of unique IP serves as a signal to the system that an attack has begun. Based on this information the binary IP address classifier was implemented, which allocates 2 classes based on network activity analysis: user requests and botnet requests. Blocking the botnet requests marked thus allowed to reduce the load on the server, avoiding for the most part the damage a from the attack.

This classifier identifies the botnet IP with high accuracy, but with them there are also IP users who tried to access the server at the time of the attack. Thus, 100% of the botnet nodes do not gain access to the server, but together with them 10-15% of users. In order to neutralize the result of false triggering, it was decided to send “suspicious” traffic to a separate message queue with a limited bandwidth. Thus, users marked as a botnet node could access the pulses and be labeled with the correct class already at the next impulse.

With the use of programs developed by the classifier was written ma, which dynamically forms chart iptables used to mark traffic a in accordance with the source class at the PREROUTING stage. To create a separate queue of packets, the HTB script was used, which creates message queues using the *tc* utility based on the described classes. The solution scheme is shown in Fig. 6.

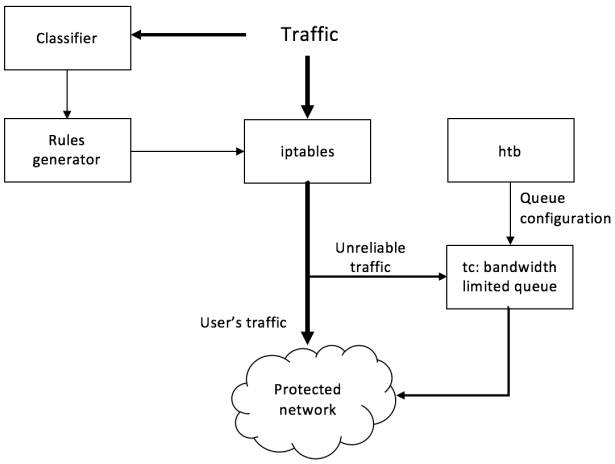

Fig. 6. Decision flowgraph

V. Results

The result of the solution's work was a decrease in the number of time-consuming waits for user model queries from 98% to 5-10% for the duration of the attack. The result of the simulation of the protection mechanism is shown in Fig. 7.

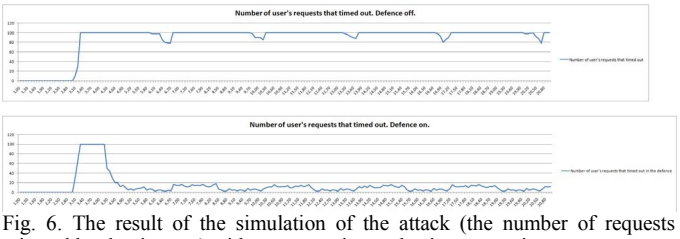

Fig. 6. The result of the simulation of the attack (the number of requests rejected by the timeout) without protection and using protection

ACKNOWLEDGMENT

The authors are sincerely grateful to the head of the Department of Computer Systems and Technologies of the National Research Nuclear University "MEPhI" Professor M.A. Ivanov for help and support during the research. Research of statistical safety of stochastic transformation blocks was held within the framework of the Program of improving the competitiveness of the National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), as well as in the framework of the priority areas grant program of the Russian Science Foundation "Fundamental and exploratory studies by individual research groups".

REFERENCES

[1] A.A. Maksutov, N.O. Fedorova, I.A. Cherepanov, M.M. Makedonskaya "General-purpose tool for modelling of custom network devices and protocols" in *Proc. 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, ElConRus 2017*, St. Petersburg, Russian Federation, 2017, pp. 176-178. DOI: 10.1109/EIConRus.2017.7910522.

[2] D. Knuth. The Art of computer programming. Volume 2: Seminumerical Algorithms. Third Edition. Addison-Wesley, 1997, 762 pp.

[3] Dowd, P.W.; McHenry, J.T., Network security: it's time to take it seriously, Computer, vol.31, no.9, pp.24-28, Sep 1998.

[4] Kartalopoulos S.V., Differentiating Data Security and Network Security, Communications, 2008. ICC '08. IEEE International Conference on, pp.1469-1473, 19-23 May 2008.

[5] Molva R., Institut Eurecom, Internet Security Architecture, Computer Networks & ISDN Systems Journal, vol. 31, pp. 787–804, April 1998.

[6] Curtin M., Introduction to Network Security, http://www.interhack.net/pubs/network-security.

[7] Marin G.A., Network security basics, Security & Privacy, IEEE, vol.3, no.6, pp. 68–72, Nov–Dec. 2005.