Proactive Approach for the Prevention of DDoS Attacks in Cloud Computing Environments

Badr Alshehry and William Allen

Abstract Contemporary security systems attempt to provide protection against distributed denial-of-service (DDoS) attacks; however, they mostly use a variety of computing and hardware resources for load distribution and request delays. As a result, ordinary users and website visitors experience timeouts, captchas, and low-speed connections. In this paper, we propose a highly inventive multilayer system for protection against DDoS in the cloud that utilizes Threat Intelligence techniques and a proactive approach to detect traffic behavior anomalies. The first layer of the model analyzes the source IP address in the header of incoming traffic packets and the second layer analyzes the speed of requests and calculates the threshold of the attack speed. If an attack remains undetected, the incoming traffic packets are analyzed against the behavior patterns in the third layer. The fourth layer reduces the traffic load by dispatching the traffic to the proxy, if required, and the fifth layer establishes the need for port hopping between the proxy and the target website if the attack targets a specific web-application. A series of experiments were performed and the results demonstrate that this multilayer approach can detect and mitigate DDoS attacks from a variety of known and unknown sources.

Keywords Distributed denial-of-service attacks · Cloud computing · Proxy firewall · Threat intelligence · Computer security

1 Introduction

Distributed denial-of-service (DDoS) attacks have become highly complicated and have enormous destructive potential. During Q2 of 2015 the most powerful attack occurred at a speed of 250 Gb/s, followed by an attack at 149 Gb/s during Q3 of the

B. Alshehry (✉) · W. Allen
School of Computing, Florida Institute of Technology, Melbourne, FL 32901, USA
e-mail: balshehry2005@myfit.edu

W. Allen
e-mail: wallen@fit.edu

© Springer International Publishing AG 2017

R. Lee (ed.), *Applied Computing and Information Technology*,
Studies in Computational Intelligence 695,
DOI 10.1007/978-3-319-51472-7_9

same year. The total number of DDoS attacks increased by 180% compared to the previous year [1]. Most complex attacks imitate ordinary HTTP traffic generated by botnets [2]. Attackers load scripts into infected botnet agents, which perform actions similar to those of ordinary users when they browse websites, but at high speed. The larger the botnet, the heavier the load it can produce on a target server. The destructive impact of a DDoS attack is that it significantly delays business processes [3, 4]. E-shops, news agencies, stockbrokers, banks, and many other types of businesses are very sensitive to stable continuous operation. Any, even short, interruption in the availability of their systems may lead to significant losses or even wide-scale disruption of the business.

In response to the threats described above, we realized the necessity of new technology for DDoS prevention. Our multilayer system implements both proactive preventive methods based on behavioral analysis, and threat intelligence, which in combination, provide proven attack prevention.

The main hypothesis of our research is that a highly effective system for DDoS protection in the cloud can be developed by taking into account the growing nature of the risk landscape.

Threat intelligence is a rapidly growing, though relatively young field of cyber security. Security vendors and independent researchers define this term as a complex process described by some common properties. We analyzed several definitions [5–9] and then combined them to form our own definition to highlight the most important properties and features of threat intelligence:

> *Threat Intelligence (TI) is a process to gather knowledge, aggregated from reliable sources, cross-correlated for accuracy. It must be timely, complete, assessed for relevancy, evaluated, and interpreted to create actionable data about known or unknown security threats that can be used to effectively respond to those threats*

The key benefits of using threat intelligence to prevent DDoS attacks are:

1. Protection of target websites from botnets (by implementing a botnet IP database and checking the incoming IPs against it), and DDoS attacks (by utilizing our five-layer system).
2. Decreasing the system load by blocking threats outside the target website perimeter (layer 2 determines the speed, layer 4 decreases the speed by the dispatcher and proxies).
3. Reducing system outages and cost of threats elimination and recovery (this is the general effect provided by our five-layer system: effective prevention of DDoS will eliminate system outages).
4. Automation of protection process from continuously growing threats (as mentioned above, we automate the prevention process by scripts used in our five-layer system).
5. Reduction of time needed to respond to new threats (because we use a proactive approach in our multilayer model, we are not required to wait until vendors identify new attack samples and update their signature bases).

Threat intelligence is a reliable modern technology to effectively protect against DDoS attacks and other threats taking into account the exponential growth of their complexity and intensity.

2 Background and Related Work

Cho et al. [10] proposed a DDoS prevention system based on the combination of a packet-filtering method with a double firewall. The first firewall analyzes the router path, whereas the second classifies data packets as being either normal or abnormal.

Botnets remain a highly destructive threat to cyber security. Graham et al. [11] attempted to detect botnet traffic within an abstracted virtualized infrastructure, such as that available from cloud service providers. They created an environment based on a Xen hypervisor, using Open vSwitch to export NetFlow Version 9. They obtained experimental evidence of how flow export is able to capture network traffic parameters for identifying the presence of a command-and-control botnet within a virtualized infrastructure. The conceptual framework they describe presents a non-intrusive detection approach for a botnet protection system for cloud service providers.

Karim et al. [12] reviewed methods of botnet detection and presented a method to classify botnet detection techniques. Their work highlights aspects pertaining to the analysis of these techniques with qualitative research design. The authors define possible future ways of improving the techniques of botnet detection and identify persistent research problems that remain open.

The evolution of DDoS attacks and their place in modern hybrid attacks and threats have been described in detail [13]. The nature of a DDoS attack, its effect on cloud computing, and problems that need to be considered while selecting defense mechanisms for DDoS were described in detail [14]. The authors' recommendation is to choose a functional, transpicuous, lightweight, and precise solution to prevent DDoS attacks, without any specific details.

The detection of DDoS attacks with the aid of correlation analysis formed the basis of research by Xiao et al. [15]. Their approach is based on a nearest-neighbors traffic classification with correlation analysis. It improves the classification accuracy by exploiting the correlation information of training data and reduces the overhead resulting from the density of training data.

Approaches to combatting both known and unknown DDoS attacks considering the real-time environment were described [16]. A method based on an artificial neural network (ANN) was used to detect attacks based on their specific patterns and characteristic features, thereby enabling these attacks to be distinguished from ordinary traffic.

3 Research Objectives and Methodology

3.1 Research Objectives

Let us define the aims and objectives of our research.

3.1.1 Defining Threat Intelligence and Its Scope

Our definition differs from other existing definitions, because in it we highlight all major properties of threat intelligence as a process to obtain knowledge. Most existing definitions describe threat intelligence as either a process or knowledge. However, it is neither knowledge nor simply a process; instead, it is a process to obtain actionable knowledge about both known and unknown threats. In our definition, we combine such mandatory properties as reliable sources, accuracy, completeness, relevancy, evaluation, interpretation, and being actionable.

3.1.2 Proposing an Innovative Method to Prevent DDoS Attacks in the Cloud Environment

Our method is different from existing methods, because we use a complex multilayer system, which combines several techniques developed by us into an integrated system. In particular, we use our own enhanced method of IP traceback, own method of threat intelligence, own method of traffic dispatching, and own method of port hopping. The joint operation and interaction of these methods make our system unique and highly effective.

3.1.3 Introducing New Proactive Approach to Defend Threats Related to DDoS Attack

According to 2015 reports of major vendors [1], threats related to DDoS attacks have been increasing significantly. Moreover, almost no new types of attacks are invented. Instead, hackers improve old existing methods and add more power to them, for example, by using an amplification method. Their main aim is to exhaust system resources and overload the communication channels. That is why we can state that threats of DDoS attacks are critical today and can be expected to be of great importance in subsequent years as well.

3.1.4 Designing a Multilayer System for DDoS Prevention in the Cloud Using Threat Intelligence Techniques

Threat intelligence techniques are used by many existing solutions. In addition, IP traceback, port hopping, and many other techniques are used to prevent DDoS attacks. Yet, there is no effective solution in the world capable of really protecting against a DDoS attack. However, in the 21st century, many servers on the Internet can be shut down with a single command using SYN flooding. Other servers can be taken offline by DNS amplification requests or other very simple types of DDoS attacks. Our system is designed to provide a complex and integrated solution that uses the power of the best techniques, which were reinvented by us to solve existing problems and eliminate existing bottlenecks.

3.1.5 Introducing an Improved Method of IP Traceback

We named our method iDPM (improved Deterministic Packet Marking). It improves standard methods of the DPM type by using two octets of the options field, which allows us to store information about the route and IP address of the packet in full, without splitting it into two or more parts, as other methods do. Our method allows us to restore the full route on the victim's side and to protect it from packet loss by using the options field to repeat each IP address in two or more packets.

3.1.6 Introducing Our Own Simple and Effective Port-Hopping Method

Our port-hopping method uses unique formulas to calculate the port number. Moreover, we use a traffic dispatcher and proxy server(s) to add additional security to this method, because only the IP address of the dispatcher is visible from an external network. A malicious user would have to break both the dispatcher and proxy and would have to know the formulas to be able to spoof the port number and connect to the target website directly.

3.1.7 Experimental Confirmation of the Effectiveness of Our Method

We test and confirm the effectiveness of our method compared to other popular techniques, including IP traceback, port hopping, and entropy-based anomaly detection.

3.2 Methodology

The concept of our work is based on combining several protection methods and adding a proactive approach with Threat Intelligence. We establish five protection layers for all incoming traffic. The logic of these steps is detailed below. The logic diagram is shown in Fig. 1.

3.2.1 Layer 1

At the first layer, we analyze the IP sources. If we find that a large amount of anomalous traffic started from some range of IPs, we check these IP addresses to determine whether they belong to Botnet IPs.

Packet-forwarding techniques such as NAT and encapsulation may be used on the way of Internet traffic. Such techniques obfuscate the real origin of packets. We analyzed the originating IP address using our traceback method, which we developed by analyzing existing IP traceback methods, selected the most appropriate approach based on deterministic packet marking, and improved it.

Our method improves the approach followed previously [17–30] and represents improved deterministic packet marking (iDPM) as having the best relation between effectiveness and ease of implementation.

Usually, the field identification (16 bits), fragment offset (16 bits including flags), DSCP (6 bits), ECN (2 bits), and even TTL (8 bits) are used in different packet marking methods. As a result, the limitation of the size of these fields does not allow the full IP to be stored in one packet; thus, it is fragmented. In our method, we propose to use the Options field for our needs. It consists of 32 bits,

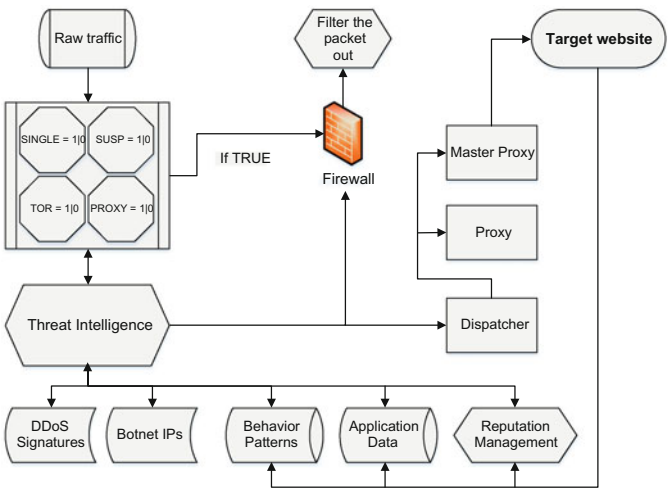

Fig. 1 Logic diagram of DDoS prevention method

which are necessary to store the full length of an IP address. Moreover, the IP header may have a variable number of options. If octet 20 is busy with real options, we may use the next octet. The maximum size of the header is 480 bits, of which only the first 160 bits (octets 0-16) are mandatory. Thus, we have 320 bits left for the Options field, which theoretically may be used to store up to 10 IPs. We use only one word (32 bits) from these 10 available to store one IP address without fragmentation.

Our aim is to trace the full route of the packets. For this purpose, we need to record the originating IP of the local computer (it may be in the format of a local network, e.g., 192.168.1.1) and the IPs of all routers through which the traffic passes. This approach allows us to trace the source IP even if NAT or proxies are used.

It is not a trivial task to detect traffic coming from botnets, because a good attack copies the user-agent of a genuine browser and imitates other signs of normal behavior. However, we can point out some initial indicators that would greatly help to reduce the power of an attack at the first two layers. These indicators include several variables (SUSP, SINGLE, PROXY, TOR) that we define and use in our method.

Traffic is considered suspicious (SUSP = TRUE) when a non-standard user-agent is detected. We allow search and stats bots, crawlers and validators as well as all standard browsers including mobile ones, but all others trigger this variable to TRUE, indicating the potential need for blocking. SINGLE = TRUE in case of a large number of requests from a single source. PROXY = TRUE indicates that the usage of a proxy server is detected. TOR = TRUE signifies the usage of Tor is detected by exit nodes or by checking the TorDNSEL value.

3.2.2 Layer 2

The second layer analyzes the speed of requests. If it is found that the rate at which inbound traffic is higher than a speed value, which is calculated below as a value of S (3), we can form blocking rules and pass them to the firewall. Otherwise, we simply pass the traffic to layer 3.

We capture the traffic by using any server tool that records the incoming traffic packets for 1 ms and counts the number of bits in the captured data. Then we multiply it by 1000 to obtain the number of bits per second.

The statistics of website visits may be taken from web analytics software such as Alexa, Google Analytics, and AWStats. We need unique visitors and the peak number of monthly visitors. Then we represent the numbers in the form of (1):

$$P = \begin{pmatrix} Range[a_1 - b_1] & peak_1 \\ Range[a_2 - b_2] & peak_2 \\ \cdots & \cdots \\ Range[a_n - b_n] & peak_n \end{pmatrix}, \tag{1}$$

where a range of monthly visitors (e.g., a_1-b_1) corresponds to the peak value of monthly visitors for the last three months for this range (e.g., $peak_1$). This information is useful to determine the possible attack speed threshold.

Assume the number of visitors for a day is U_i, where i is the day, d is the number of days (30), A is the number of visitors required to trigger an attack, then we multiply the corresponding number obtained from (1) by M to allow an excess number of visitors before we trigger an attack (the value of M is defined by experiments):

$$A = P\left[\left[\frac{\sum_{i=1}^{d} U_i}{d}\right]\right] * M. \tag{2}$$

The exact numbers in (1) may vary for different studies, but it does not affect the general formulas for A and S. This means the formulas we developed in this study will be universal for any other types of websites.

Once we know the number of visitors triggering the attack, we can calculate the speed of attack S:

$$S = \frac{A}{86400} * sizeof(packet) \tag{3}$$

where S is the rate at which we can consider traffic to be malicious in regard to DDoS attack activity.

3.2.3 Layer 3

Traditionally, Threat Intelligence is associated with the number of feeds received from many different sources. Special dedicated staff analyze these feeds for relevancy and all other properties. Although we also use feeds, our TI system is more complex in that it represents a combination of five modules. The TI architecture is shown in Fig. 2.

1. Behavioral Patterns. In our system, the criteria are specific to DDoS attacks and are provided by the target website we protect.
2. Application Data protects against ADDoS (Application DDoS) attacks.
3. Botnet IPs by third-party services.
4. DDoS signatures by third-party services.
5. Reputation Management (RM). We have identified our method of RM by calculating the reputation for each packet using the values of variables SUSP, SINGLE, PROXY, TOR, S, and the speed received from the previous layer.

We next define the formulas for attack detection A at a given time t–$A(t)$. Assume:

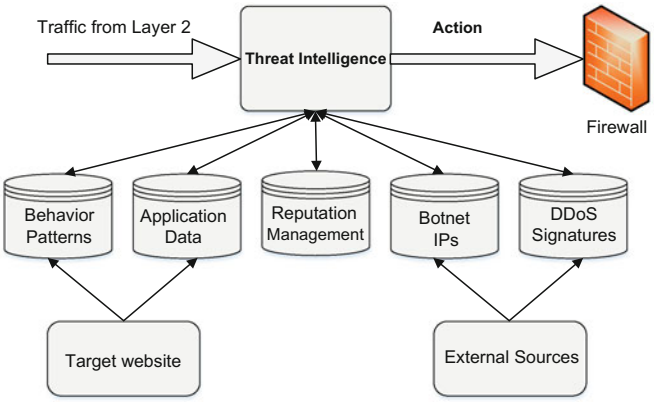

Fig. 2 Threat intelligence architecture

P pages per visit,
T time on site,
V new visitors,
B bounce rate,
R_N response time of a target website page,
N number of target website pages
RT reputation based on traffic variables
RP value of Reputation Management
A_1 attack detection for method 1 (behavior patterns),
A_2 attack detection for method 2 (application data)
A_3 attack detection for method 5 (reputation management)

Then—

$$A_1(t) = \begin{cases} 1, |P_t \to 0 \\ 0, |P_t \to \infty \\ 1, |T_t \to 0 \\ 0, |T_t \to \infty \\ 1, V_t \to \infty \\ 0, |V_t \to 0 \\ 1, |B_t \to \infty \\ 0, |B_t \to 0 \end{cases} \tag{4}$$

$$A_2(t) = \begin{cases} 0, \left| \left(\frac{\sum_{i=1}^{N} R_{Ni}}{N} \right)_t \to 0 \right. \\ 1, \left| \left(\frac{\sum_{i=1}^{N} R_{Ni}}{N} \right)_t \to \infty \right. \end{cases} \tag{5}$$

$$RT(t) = \begin{cases} 0, \ |speed < S \\ 1, \ |(PROXY = 1 \ OR \ TOR = 1) \ AND \ SUSP = 1 \\ 1, \ |speed > S \ AND \ (PROXY = 1 \ OR \ TOR = 1) \\ 1, \ |speed > S \ AND \ SUSP = 1 \end{cases} \tag{6}$$

$$RP(t) = \frac{RT(t)}{1 / \left(\left(\frac{\sum_{i=1}^{N} R_{Ni}}{N} \right)_t * S \right)} * 100\% \tag{7}$$

$$A_3(t) = \begin{cases} 0, \ |RP(t) < 100\% \\ 1, \ |RP(t) \geq 100\% \end{cases} \tag{8}$$

$$A(t) = A_1(t) \ AND \ A_2(t) \ AND \ A_3(t) \tag{9}$$

In the result, if *A(t)* = 1, we have an active attack at the given time, otherwise there is no attack.

3.2.4 Layer 4

The fourth layer dispatches the traffic to the proxy server to reduce traffic load, if necessary.

3.2.5 Layer 5

This is the last protection layer of our methodology and it strengthens our method by adding the port hopping technique. Our designed pseudo-random algorithm for changing port numbers resides in the fifth layer. This algorithm is known only to the proxy and target website.

$$Port(t) = (PRND0 \oplus t) \ mod \ 65535, \tag{10}$$

where PRND0 is the pseudo-random number generator, synchronized between the proxy and the target website, t is the current time and 65535 is the greatest possible port number.

4 Experiments

In the course of our study we conducted practical experiments to verify our assumptions and methods of attack detection and prevention. The aim of the experiment is to prove that our developed method to prevent DDoS attacks in the cloud is effective, accurate, and has strong advantages compared to other methods.

We tested source IP detection for different scenarios (with real and spoofed source IP) for layer 1, and then calculated the speed for layer 2 for low speed, normal speed, and high speed attacks. Figure 3 displays the charts for different speed, and Table 1 lists the values for the corresponding speed.

Then we calculated the values of A1 for behavioral patterns, A2 for application data, A3 for reputation management, and the resulting A indicating the attack at layer 3.

After that we divided the speed by Dispatcher at layer 4, as shown in Fig. 4. Then we checked the generation of port numbers for the port hopping method at layer 5. We ran the script generating the port numbers using our method, and we confirmed the numbers were random and different each time.

Lastly, we ran experiments using the overall method that resulted in the generation of firewall blocking rules. Table 2 contains the results of launching attacks from 1, 2, and 5 of our 5 VM clients.

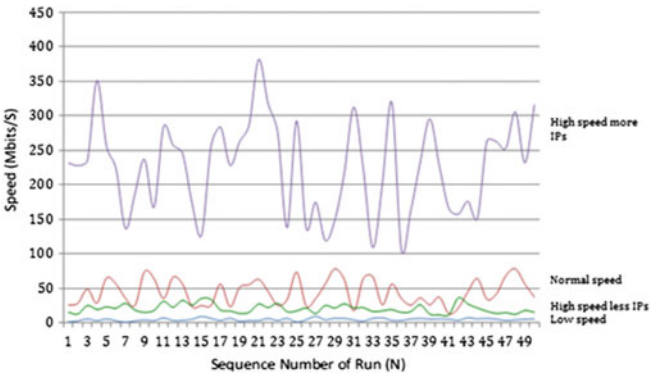

Fig. 3 Source IP detection at different traffic speeds

Table 1 Values of speed

Seq no. of run	Low speed	Normal speed	High speed (fewer IPs)	High speed (more IPs)
1	1,5	25,5	15,3	231,6
2	2,8	28,3	12,8	227,8
3	5,7	49,1	25,1	235,2
4	3,2	28,6	19,5	351,2
5	5,7	65,1	23,2	256,3
6	3,2	56,4	21,3	223,6
7	0,9	35,6	28,3	136,6
8	2,8	25,3	18,4	187,4
9	4,3	74,2	15,2	236,9
10	3,2	64,2	17,9	167,4

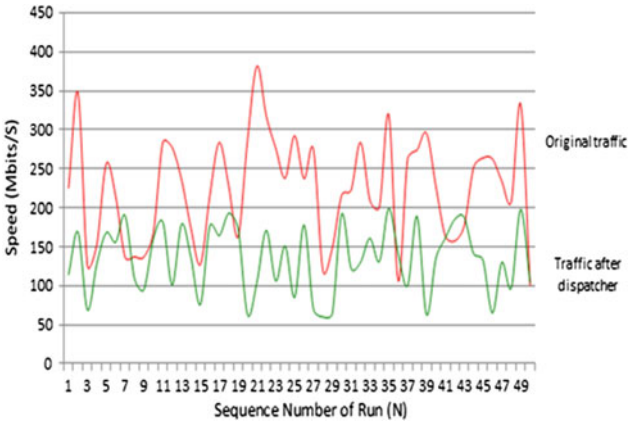

Fig. 4 Speed before and after processing by the dispatcher at layer 4

Table 2 Experimental results for overall method

IPs of attack	Number of IPs detected	Number of IPs blocked	False positive rate, %
VM6 192.168.0.141	1	1	0
VM6 192.168.0.141 VM7 192.168.0.138	2	2	0
VM6 192.168.0.141 VM7 192.168.0.138 VM8 192.168.0.139 VM9 192.168.0.143 VM10 192.168.0.144	5	5	0

Table 3 Comparison of methods

	Our method	Method 1[a] [31]	Method 2[b] [32, 33]	Method 3[c] [34]
Usage of multiple technologies	yes	no	no	no
Number of technologies used	6	1	1	1
Requiring of user's actions	no	yes	no	no
False positive rate	0%	50%	50%	30%
Average Response time (ms) of target on low load	0.346	0.564	0.287	0.334
Average Response time (ms) of target on high load	0.513	0.974	0.816	0.806
Dependence on signatures	no	no	no	no
Ability to detect unknown threats	yes	yes	yes	yes
Practical implementation	yes	yes/no	yes	yes
Overall System	excellent	good	good	good

[a]IP traceback
[b]Port hopping
[c]Entropy-based anomaly detection

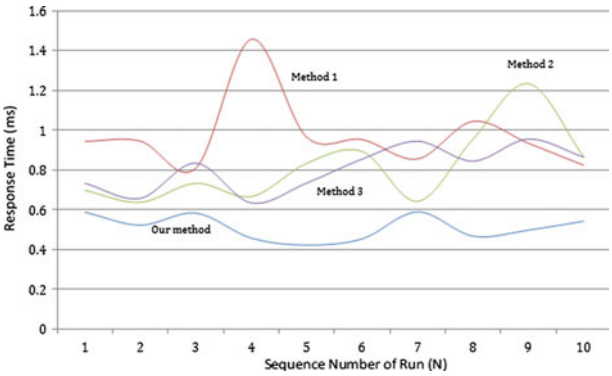

Fig. 5 Response time for different methods

Thus, the result is confirmed, and the attack is successfully detected and blocked. Furthermore, we compared our method to other popular methods used for DDoS protection. These results are presented in Table 3 and shown in Fig. 5.

Thus, all experiments were successful and we met the expected results. Our main hypothesis is confirmed, and the attacks we successfully detected and blocked.

In addition, we compared our method to other popular methods used for DDoS protection.

For example, Fig. 5 below shows the response time in milliseconds for different methods using a high traffic load:

5 Conclusions

The hypothesis of our research is that a highly effective DDoS protection system for the cloud can be developed taking into account the growing nature of the risk landscape. Our study and the set of experiments we carried out showed that existing methods have shortcomings and they could not positively confirm this hypothesis. At the same time, our proposed method introduced enhancements to existing techniques such as IP traceback, port hopping, and reputation management. Moreover, we introduced a completely new definition and methods for threat intelligence and the results of our experiments confirmed that the definition occupies a central part of our protection method and allows complex DDoS attacks to be prevented proactively without human intervention. Thus, we confirmed our main hypothesis and we showed that our method produces 0% false positives, a minimal response time on target with and without load, the ability to detect unknown threats, and a high level of practical implementation.

We would like to suggest possible follow-up studies in response to our research, related to different combinations of DDoS protection techniques in one complex multilayer system such as ours.

There are many possible ways to incorporate existing or newly developed methods into one system, a variety of possible protocols to achieve their interaction, and their potential enhancements.

Modern threats dictate the overestimation of the potential consequences of a successful attack and require us to always be a step ahead of malicious attackers. This in turn requires new complex methods to be used instead of a single technology. No standalone technology today would stop a powerful DDoS reflection attack over 500 Gb/s. The only way to be victorious over cybercriminals is to combine the efforts of scientists and security vendors to produce all-in-one solutions that would proactively mitigate attacks of any given type.

References

1. Akamai, State of the Internet Report (2015).
2. Wang, A., Mohaisen, A., Chang, W., Chen, S.: Delving into internet DDoS attacks by botnets: characterization and analysis. In: 45th Annual IEEE/IFIP International Conference Dependable Systems and Networks (DSN), 379–390 (2015).
3. Arbor Networks. Worldwide Infrastructure Security Report, DDoS Threat Landscape. APNIC Conference (2016).
4. Riverhead Networks. DDoS Mitigation: Maintaining Business Continuity in the Face of Malicious Attacks, Cupertino: Riverhead, Cisco (2004).
5. Friedman, J., Bouchard, M.: Definitive Guide to Cyber Threat Intelligence, CyberEdge Press (2015).
6. Cyber threat intelligence - how to get ahead of cybercrime, Ernst & Young Global Limited (2014).
7. Chismon, D., Ruks, M.: Threat Intelligence: Collecting, Analysing, Evaluating. MWR InfoSecurity Ltd (2015).
8. Farnham, G., Leune, K.: Tools and standards for cyber threat intelligence projects, SANS Institute (2013).
9. McMillan, R.: Definition: Threat Intelligence. Gartner, 2013.
10. Cho, J.H., Shin, J.Y., Lee, H., Kim, J.M., Lee, G.: DDoS Prevention System Using Multi-Filtering Method (2015).
11. Graham, M., Winckles, A., Sanchez-Velazquez, E.: Botnet detection within cloud service provider networks using flow protocols. In: IEEE 13th International Conference on Industrial Informatics (INDIN), 1614–1619 (2015).
12. Karim, A., Salleh, R.B., Shiraz, M., Shah, S.A.A., Awan, I., Anuar, N.B.: Botnet detection techniques: review, future trends, and issues. Journal of Zhejiang University SCIENCE C **15**, 943–983 (2014).
13. Mansfield-Devine, S.: The evolution of DDoS. Computer Fraud & Security **2014**, 15–20 (2014).
14. Deshmukh, R.V., Devadkar, K.K.: Understanding DDoS Attack & its Effect in Cloud Environment. Procedia Computer Science **49**, 202–210 (2015).
15. Xiao, P., Qu, W., Qi, H., Li, Z.: Detecting DDoS attacks against data center with correlation analysis. Computer Communications **67**, 66–74 (2015).
16. Saied, A., Overill, R.E., Radzik, T.: Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing **172**, 385–393 (2016).
17. Ramesh, S., Pichumani, S., Chakravarthy, V.: Improving the Efficiency of IP Traceback at the DoS Victim. http://www.cs.utah.edu/~sramesh/attachments/ip_traceback.pdf.

18. Saurabh, S., Sairam, A.S.: Increasing Accuracy and Reliability of IP Traceback for DDoS Attack Using Completion Condition. Int. J. Network Security **18**, 224–234 (2016).
19. Li, J., Sung, M., Xu, J., Li, L.: Large-scale IP traceback in high-speed Internet: Practical techniques and theoretical foundation. In: Proceedings of the IEEE Symposium on Security and Privacy, 2004. 115–129 (2004).
20. Gong, C., Sarac, K.: IP traceback based on packet marking and logging. In: IEEE Conference on Communications (ICC). **2**, 1043–1047 (2005).
21. Foroushani, V.A., Zincir-Heywood, A.N.: Deterministic and authenticated flow marking for IP traceback. In: IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 397–404 (2013).
22. Yan, D., Wang, Y., Su, S., Yang, F.: A precise and practical IP traceback technique based on packet marking and logging. J. Inf. Sci. Eng. **28**, 453–470 (2012).
23. Aghaei-Foroushani, V., Zincir-Heywood, A.N.: On evaluating IP traceback schemes: a practical perspective. In IEEE Security and Privacy Workshops (SPW), 127–134 (2013).
24. Sung, M., Xu, J. IP traceback-based intelligent packet filtering: a novel technique for defending against Internet DDoS attacks. IEEE Trans. Parallel Distrib. Syst. **14**, 861–872 (2003).
25. Park, K., Lee, H.: On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In *INFOCOM 2001.* Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Proceedings **1**, 338–347 (2001).
26. Song, D.X., Perrig, A.: Advanced and authenticated marking schemes for IP traceback. In: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Proceedings, **2**, 878–886 (2001).
27. Parashar, A. Radhakrishnan, R.: Improved deterministic packet marking algorithm for IPv6 traceback,. In: International Conference on Electronics and Communication Systems (ICECS), 1–4 (2014).
28. Amin, S.O., Hong, C.S.: On IPv6 Traceback. In: The 8th International Conference on Advanced Communication Technology, ICACT 2006. **3**, 2139–2143 (2006).
29. Amin, S.O., Kang, M.S., Hong, C.S.: A lightweight IP traceback mechanism on IPv6. In: Emerging Directions in Embedded and Ubiquitous Computing, Amin, S.O., Kang, M.S., Hong, S.C. (Eds.) Springer, Berlin Heidelberg (2006).
30. Kim, R.H., Jang, J.H., Youm, H.Y.: An Efficient IP Traceback mechanism for the NGN based on IPv6 Protocol, IITA'09 (2009).
31. Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical network support for IP traceback. In: ACM SIGCOMM Computer Communication Review, **30**, 295–306 (2000).
32. Shi, F.: U.S. Patent No. 8,434,140. Washington, DC: U.S. Patent and Trademark Office (2013).
33. Morris, C.C, Burch, L.L., Robinson, D.T.: U.S. Patent No. 8,301,789. Washington, DC: U.S. Patent and Trademark Office (2012).
34. Source code of the entropy-based network traffic anomaly detector. Retrieved May 26, 2016, from https://github.com/anacristina/entropy-based-anomaly-detector.