Early DDoS Detection Based on Data Mining Techniques

Konstantinos Xylogiannopoulos[1], Panagiotis Karampelas[2], and Reda Alhajj[1]

[1] University of Calgary, Calgary AB T2N 1N4, Canada
[2] Hellenic Air Force Academy, Dekelia Air Force Base, Attica, Greece

Abstract. In the past few years, internet has experienced a rapid growth in users and services. This led to an increase of different type of cyber-crimes. One of the most important is the Distributed Denial of Service (DDoS) attack, which someone can unleash through many different isolated hosts and make a system to shut down due to resources exhaustion. The importance of the problem can be easily identified due to the huge number of references found in literature trying to detect and prevent such attacks. In the current paper, a novel method based on a data mining technique is introduced in order to early warn the network administrator of a potential DDoS attack. The method uses the advanced All Repeated Patterns Detection (ARPaD) Algorithm, which allows the detection of all repeated patterns in a sequence. The proposed method can give very fast results regarding all IP prefixes in a sequence of hits and, therefore, warn the network administrator if a potential DDoS attack is under development. Based on several experiments conducted, it has been proven experimentally the importance of the method for the detection of a DDoS attack since it can detect a potential DDoS attack at the beginning and before it affects the system.

1 Introduction

The proliferation of the internet enabled smart mobile devices all over the world along with the available networked corporate or home personal computers has created an enormous battlefield for cyberwar and cyber games. New devices have become the target of malevolent hackers who desire to take advantage of the security weaknesses of the newly available applications together with the illiteracy of the new users of this technology. By taking control of the innumerous devices, cyber criminals can materialize their plans easily e.g., to invade users privacy, to steal users identity, to start different types of attacks such as Distributed Denial of Service (DDoS) attacks, scan attacks or Trojan attacks [1]. According to recent reports [2], more than 4,800 DDOS attacks per day take place, more than 80 GBPs bandwidth is utilized for these attacks and more than 900 active botnets are ready to flood the Internet and disrupt the legitimate services. Monitoring and detecting such types of attacks has become increasingly demanding since the DDoS attacks correspondingly have become sophisticated and the Internet traffic due to the increasing number of the new devices has become enormous

D. Naccache and D. Sauveron (Eds.): WISTP 2014, LNCS 8501, pp. 190–199, 2014.

[3] and thus difficult to monitor. DDoS attacks as the abovementioned statistics reveal are launched by very big botnets or computers infected with software that allows their remote control by the attacker. All the infected hosts, which typically are some thousands, then attack one or different legitimate services by sending thousands or millions of requests in a few seconds. The attacked host is flooded with different types of packets such as SYN, FIN or other types of packets and as a result stops responding and offering the legitimate service. Several known internet sites such as Yahoo, Dell, eBay, Amazon, ZDNet, British Telecom and countries such as Georgia, Estonia [4] and more recently Syria and Ukraine have been targeted by DDoS attacks that caused either financial losses or serious problems in the operation of public services correspondingly.

The protection of critical infrastructure and services against DDoS attacks has occupied researchers working in the fields of networking and cybercrime a lot. Several different techniques have been proposed either to prevent or to detect DDoS attack. A task that is not easy due to the existence of diverse characteristics of the attack. More specifically, J. Mikrovic [6] in her PhD thesis identified the characteristics of DDoS attacks that make DDoS defense very complicated. The characteristics mentioned are the diverse methods the attackers use, e.g., the different stream of packets they sent, the coordination of the distribution of the attack makes very complicated the detection of the attack e.g., geo-dispersed botnets are used, the sophisticated coverage of the traces of the attacker e.g., through IP address spoofing, the availability of several tools that can launch DDoS attacks, e.g., Trinoo, Stacheldraht, etc. and the illiteracy of the Internet users e.g., users who do not update their operating systems in order to address potential security holes. These characteristics of DDoS attacks have obliged researchers to propose different approaches in order to either prevent the infrastructure from DDoS attacks or early detect the DDoS attack and safeguard the infrastructure. A classification of DDoS Mechanisms [5] suggests two generic categories, the preventive and reactive methods. Preventive methods can be further distinguished to Attack prevention methods that increase the security of the hardware or software resources of an organization e.g., by deploying automatic updating schemes, by continually monitoring the access rights, by deploying security related infrastructure such as firewalls or intrusion detection systems. Another type of preventive methods attempt to prevent specifically DDoS attacks. These methods either balance the load of an attack intelligently or utilize a very large number of resources that can endure DDoS attacks. The reactive methods on the other hand focus on the early detection of a DDoS attack and the elimination of its impact to the infrastructure. Reactive methods are further distinguished in pattern detection methods or anomaly detection methods. The pattern detection methods usually monitor the system under protection by identifying and comparing possible patterns against stored signature of known attacks. The anomaly detection methods on the other hand attempt to identify anomalies in the normal or standard operation of the network under protection. All these different types of preventive or reactive methods cannot provide 100% protection of a system against DDoS attacks and that is why the research in the

field of defense against those types of attacks is on-going and new methods are constantly introduced. The contribution of this paper is the proposal of an innovative DDoS detection method that combines anomaly with pattern detection. A data mining technique developed by the authors which can identify all the repeated patterns of a sequence is applied to the data received in the network. When several IP addresses from the same domain are detected by this technique a potential DDoS attack may occur. Based on the experiments, the time needed to identify the launch of the DDoS attack ranges from 1-4 seconds depending on the initial parameters provided to the algorithm.

The rest of the paper is organized as follows: Section 2 presents a review of pattern and anomaly detection methods. Section 3 presents the approach proposed using the pattern detection method that is developed by the authors. Section 4 presents the experimental results by the application of the proposed methodology to an existing publicly available dataset with DDoS attack data and discusses the experimental results. Finally, the conclusions and future work is presented.

2 Related Work

Several researchers have focused on the detection of patterns or signatures of DDoS attacks using various methods such as statistical methods, artificial neural networks, data mining techniques, hybrid techniques, etc. Statistical-based methods [7] monitor and model normal traffic patterns by using advanced statistical analysis techniques and are able to detect anomalies based on a predefined threshold. An example of this method is described in [7]. This type of technique may provide relatively accurate results depending on the statistical analysis technique used. However, the selection of threshold is very important since either real DDoS attacks may not be detected or legitimate requests may be tagged as DDoS attacks. If the threshold selected is rigid it may increase the false positive detections while it will decrease the false negative detections [10]. Especially in cases where there is an increase of traffic as for example when an event such as a music concert occurs the corresponding website of the event may be visited by several people who wish to learn information about the event, the accessibility of the venue, etc. and thus the traffic will increase as the event approaches. The increasing traffic may be identified as DDoS attack because it does not follow the normal traffic patterns processed by the statistical based detection method and thus a false alarm may be issued.

Neural network based methods aggregate already identified patterns related to DDoS attacks and by applying machine learning techniques, develop a neural network that can analyze the traffic in a network and decide whether a DDoS attack is in progress or not. Such a technique is presented in [9]. The algorithm described operates in two stages. In the first stage it monitors various features of the traffic and estimates the likelihood ratios for a DDoS attack. In the second stage the algorithm combines the result of each feature identified and the results are forwarded to the neural network that provides the final decision whether a

DDoS attack has been detected or not. The result of such techniques are heavily dependent on the selection of features. If the features selected do not correspond to the type of DDoS attack carried then the detection will not be possible.

Data mining based techniques have also been introduced in the detection of DDoS attacks. Data mining algorithms can be employed in the automatic feature selection for monitoring and the classification of the traffic patterns as in [8] in which the decision tree algorithm is used to select the traffic attributes that need to be analyzed. Then using a classification algorithm it is possible to decide whether there is a DDoS attack or not. Other data mining algorithms such as C4.5 algorithm association rule mining have been applied to detect attack patterns in [11]. The C4.5 algorithm is first applied to develop a learning model for known attack types and then association rule mining is used for in-depth semantic interpretation of the attack type. This approach combines different data mining techniques for the detection and analysis of the monitored traffic and notifies the network managers regarding possible DDoS attacks.

Hybrid methods finally combine elements of the above mentioned techniques in order to improve the positive detection rates of DDoS attacks. Such a system is proposed in [12] that combines anomaly detection with weighted association rules in order to produce signatures of attacks. These signatures are used in order to identify similar new and unknown future attacks. The combination of these techniques according to the results reported in [12] outperforms the corresponding individual methods used. In another work statistical based methods and data mining techniques are used to propose a multistage detection of DDoS attacks [13]. The method is based on various statistical analysis model e.g., Markov based prediction at the first stage and wavelet based singularity detection for sending DDoS attack alerts.

Most of the methods presented in this section are addressing three stages of the DDoS defense process: the detection phase, the classification phase whether it is a DDoS attack or not and finally the response [14]. The proposed method in this paper focuses on the first phase of the defense process by detecting an anomaly in the network traffic using a novel pattern recognition algorithm which discovers all the repeated patterns in a given sequence. In other words the proposed method acts as an early warning system and reports abnormal activity in network traffic.

3 Our Approach

The method proposed in this paper is based on the Suffix Array data structure that is used to detect all repeated patterns in a sequence. More specifically, the ARPaD Algorithm [19] is used as it has been derived by COV Algorithm [15], [16], [17]. A Suffix Array is a data structure that contains an array of all suffixes of a string [20] and it is mainly used for pattern detection. However, with the use of the actual suffix strings we can construct a similar to a suffix array data structure for fixed width substring such as the IP strings of length 12 by adding leading zeros in octets that do not have length three. By doing this, ARPaD Algorithm can analyze the strings of the IPs and detect all repeated patterns,

which in this case are domains, subnets or actual hosts when the string is a full IP address of length 12.

The first step to apply the ARPaD algorithm in the log files of traffic is to convert the IP addresses found to actual strings that will be used in order to detect all repeated patterns. For this reason, each one of the triplets of the IPs that are not full (i.e. have less than three digits) is converted to a full triplet by adding in front of each number the necessary number of zeros. This transformation will allow ARPaD Algorithm to search into IP addresses as they were simple strings.

The second step that is required is to sort all the IP addresses alphabetically since now all have been converted to strings. This is needed for the ARPaD Algorithm in order to perform the analysis as the strings have directly come from a Suffix Array data structure. This is the most time consuming part since it has complexity $O(nlogn)$, while ARPaD Algorithm has been proven experimentally to have on average complexity $O(n)$ [16], [18], [19]. Therefore, the total complexity of the method is on average $O(nlogn)$ which allows a very fast analysis of the IP addresses data.

The last step in the proposed methodology is to execute ARPaD Algorithm on the sorted array of IP address strings and retrieve as results all the repeated substrings (IP prefixes of the domains or subnets) or strings (full IP addresses). Having the results a Network Administrator can set a threshold in the occurrences of the substrings (IP prefixes) that are detected in order to characterize the traffic from a specific domain, or subnet or host as possible abnormality and, therefore, a potential DDoS attack. The specific threshold has to be set depending on the type of analysis, hits number or time. Based on the defined threshold and the detected traffic, the proposed method can send a warning of a potential DDoS attack to the administrator. Furthermore, the Network Administrator can use the proposed method to continue monitoring the traffic and can opt to perform further analysis based on specific time interval or a specific number of hits on the router. This is something that a specialist can decide, yet, the proposed methodology allows both implementations to be applied interchangeable. Depending on the traffic and the potential DDoS attack warning, these intervals (time or numbers) can change dynamically to accelerate the analysis and prevent an attack at the beginning. For example, in a normal traffic situation you can analyze the hits per minute but when a warning is issued instead of time interval, a specific number of hits e.g., 100,000 can be analyzed. In a DDoS attack situation this is expected to be reached in a few seconds depending on the magnitude of the attack.

4 Experimental Results

For the experiments a laptop with Intel i5 quad core processor and 8Gb RAM has been used. The code of ARPaD algorithm has been written in C# and a 64bit operating system has been used. The data come from the Computer Science Department of University of California Los Angeles (UCLA) website that holds

information about packet traces. We have used Trace Set 2 for the UDP packets that includes 16 files. Each one of the first 15 files has 100,000 hits generated from a DDoS attack. The last file holds less information and we haven't used it in order to have a uniform distribution of the hits per file and thus to be comparable. There have been contacted in total three major experiments. In each experiment different number of hits (100,000 IPs, 500,000 IPs and 1,500,000 IPs) has been used. For the first experiment, the ARPaD algorithm run 15 times to analyze all the 15 files. In the second experiment the algorithm run three times and in the third one the algorithm run once. For each experiment, two different versions of the algorithm run in order to just detect all repeated patterns (IPs prefixes) or all repeated patterns including the positions of each one in timeline. The latter is more time consuming that the first yet can provide more detail information for further analysis.

Table 1. IP Detection Time per File (100,000 rows per file)

From	To	Detect IPs	Detect IPs & Positions	DDoS Attack
1	100	0.992	3.22	9.57
100,001	200	0.988	3.26	8.93
200,001	300	0.991	3.22	8.86
300,001	400	0.994	3.23	9.77
400,001	500	0.988	3.17	11.15
500,001	600	0.989	3.13	9.81
600,001	700	1.006	3.23	11.00
700,001	800	0.992	3.22	10.65
800,001	900	1.013	3.17	10.56
900,001	1,000,000	0.975	3.23	10.68
1,000,001	1,100,000	0.993	3.20	10.77
1,100,001	1,200,000	0.988	3.22	9.75
1,200,001	1,300,000	1.021	3.24	10.73
1,300,001	1,400,000	0.995	3.19	10.79
1,400,001	1,500,000	0.991	3.17	13.10

In Table 1 we can see the time analysis per single file. The table includes the time ARPaD Algorithm needs to simply detect all repeated patterns and the time the Algorithm needs to detect the patterns and their position in the timeline. The position of each pattern can be further used to detect density or increase in hits or other attributes per pattern (IP prefix) that might be useful to detect DDoS attack. Moreover, the table includes the time the attack last for each one of the 100,000 hits based on the data provided in the files from UCLA. As we can see from Table 1 the time ARPaD Algorithm needs to detect all repeated patterns is approximately 1 second on average, including the sorting process time when we do not record the actual positions of the hits in timeline. If the time factor needs to be calculated then the Algorithm needs approximately 3.2 seconds on average for each file. However, what is very important to be mentioned here is that the attack needs approximately 10 seconds while the analysis can be

performed in 1 or 3.2 seconds depending on which variation of the algorithm is used. Therefore, the analysis can be performed faster than the attack as in case there is a DDoS attack with 100,000 hits per second, the algorithm can provide the results of the analysis the next second. As a result, we can run a pattern detection analysis every few seconds and have an early warning when something abnormal is happening and before the next sequence of IPs will need to be analyzed. The time interval to analyze the data can change automatically depending on the results found in the previous run of the algorithm.

Table 2. IP Detection Time per 5 Files (500,000 rows)

From	To	Detect IPs	Detect IPs & Positions	DDoS Attack
1	500	4.32	14.30	48.29
500,001	1,000,000	4.31	14.53	52.69
1,000,001	1,500,000	4.33	14.21	55.15

Table 3. IP Detection Time for the Whole Data Set (1,500,000 rows)

From	To	Detect IPs	Detect IPs & Positions	DDoS Attack
1	1,500,000	11.67	43.20	156.12

In Table 2 the results of the second experiment (the hits per 500,000 IPs) are presented. Again we have the same information as in Table I regarding times and we can observe that the detection of all repeated patterns is approximately 4.3 seconds on average while when all repeated patterns and their position in timeline is detected the ARPaD Algorithm needs approximately 14.3 seconds. In this experiment, the total attack time per 500,000 hits is approximately 52 seconds, time considerably longer than the time ARPaD Algorithm needs to detect all repeated patterns. ARPaD Algorithm it has been proved to be linear on average [15], [16], [19] and that is why it preserves the ratio of 1/10 for the simple pattern detection and approximately 1/4 for the full pattern detection (including positions). Finally, we run one more experiment for the whole data set for the 1.5 million IPs (Table 3). The time for the single detection is 12 seconds, for the full detection is 43 seconds while the whole DDoS attack lasted 156 seconds approximately according to the files provided by UCLA.

The fact that our method can perform a very fast analysis in real time can have several benefits and can also allow variation of implementations of the method. One way is to have a fixed width analysis per time or number of hits as we have already described with the experiments and the results in Table 1 through Table 3. However, we can apply a dynamic execution of the method and allow the Network Administrator to fully parameterize it and decide how the pattern detection will be executed. The process can be the following: In a normal environment we run checks based on a fixed, wide, interval which can be based on either on time or number of

IPs. If the system detects an abnormality then it can manually or automatically decrease the width of the intervals in order to prevent a DDoS attack. For example, we can have a fixed width of 500,000 hits. The analysis of these needs approximately 4.3 seconds while the time needed for the hits is 52 seconds. So, in time t we execute an analysis for the 500,000 hits and the system detects an abnormality in time t+4.3. Now the system can change the time interval and perform an analysis per 100,000 hits. This will happen at t+10 or 5.7 seconds after the first analysis has been conducted by the ARPaD Algorithm. Therefore, the network administrator can specify intervals that can easily be executed without overlapping or without lags in order to have a flexible, dynamic early warning DDoS attack detection system. When the traffic will return to normal level then the system can again increase its intervals.

We can see in Table 4 the full list of results for the domains and subnets in the whole 1.5 million hits that the 15 files have. The first and third column contains all repeated patterns detected and the second and fourth column contains the number of occurrences of each pattern (IP prefixes) correspondingly. The table is sorted based on the IP prefixes. The domain 1.1.139.x has been detected 1,500,00 times during the DdoS attack. For each subnet of the previous domain we have 588,537 for the 1.1.139.0x, 583,884 for the 1.1.139.1x and 327,579 for the 1.1.139.2x. The subnet with the most occurrences 59,941 has IP Prefix 1.1.139.17x and the subnet with the least occurrences 29,826 has IP Prefix 1.1.139.25x. The single IP addresses have not been included because the list is very large and almost all the possible IP addresses from the specific domain have been used in the DDoS attack in this data set. The IP with the most occurrences (hits) is the IP with address 1.1.139.149 with 6,140 hits and the IP with the least hits is the 1.1.139.181 with 5,752 hits. From the results presented in Table 4 we can detect the hits from a specific domain and further how this can be analyzed per subnet or even per host. For example, we can observe in Table 4 the hits per subnet are almost uniformly distributed which it cannot be a real life situation. Therefore using this analysis, it is possible to determine if a system is under DDoS attack or not.

Table 4. Full IPs List Ordered By Occurrences

IP Prefix	Occurrences	IP Prefix	Occurrences	IP Prefix	Occurrences
1001139	1,500,000	100113906	59,909	100113916	53,74
10011390	588,537	100113907	59,675	100113917	59,941
10011391	583,884	100113908	59,796	100113918	59,291
10011392	327,579	100113909	59,539	100113919	53,886
100113900	53,548	100113910	59,352	100113920	59,616
100113901	59,151	100113911	59,396	100113921	59,22
100113902	59,088	100113912	59,654	100113922	59,213
100113903	59,444	100113913	59,679	100113923	59,913
100113904	59,173	100113914	59,777	100113924	59,791
100113905	59,214	100113915	59,168	100113925	29,826

5 Conclusions

We have proposed in the current paper a novel methodology that can allow a network administrator to prevent a DDoS attack manually or automatically. Our method, based on an advanced data mining technique, takes advantage of the very fast ARPaD Algorithm that can detect all repeated patterns in a sequence. Using this algorithm, abnormal number of hits from specific domains or subnets can be detected and characterized as a potential DDoS attack. Such analysis allows to use our method as an early warning system of DDoS attack that can help to stop the attack at the beginning. The method has been applied in a 1,500,000 hits data set from the Computer Science Department of UCLA and the results showed that the method can analyze and detect the potential DDoS attack in 1/10 of the total time of the attack.

In future work, we can also use more characteristics of the ARPaD Algorithm and more specifically the detection of each IP hit in the timeline. This can help detecting any further attributes of the hits such as density and increase of the hits over time. For example, we may have periodic attacks from different hosts. The proposed method can detect the periodic pattern and correctly alert the network administrator for further action. Additionally, after the early warning of potential DDoS attack the system can also continue further investigation in order to determine if abnormal traffic from specific domain or IPs can be defined as a positive DDoS attack. This can be accomplished by analyzing further attributes included in the traffic, e.g., packets, time, etc. Finally, analyzing the IPs it is possible to very fast identify whether the traffic from a specific host is legitimate or spoofed and as a result has a very good indication if the traffic is in the context of a DDoS attack or not.

References

1. Hoque, N., Monowar, H., Bhuyan, R.C., Baishya, D.K., Bhattacharyya, J.K.: Kalita, Network attacks: Taxonomy, tools and systems. J. Netw. Comput. Appl. 40, 307–324 (2014)
2. ARBOR Networks, DDOS and Security Reports Live Feed, `http://www.arbornetworks.com/asert/2014/03/pravail-security-analytics-packetloop/` (retrieved March 20, 2014)
3. Wang, D., Yufu, Z., Jie, J.: A multi-core based DDoS detection method. In: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), July 9-11, vol. 4, pp. 115–118 (2010)
4. Loukas, G., Oke, G.: Protection against denial of service attacks: A survey. Computer J. British Computer Society. 53, 1020–1037 (2010)
5. Mirkovic, J., Reiher, P.: A taxonomy of DDoS attack and DDoS defense mechanisms. SIGCOMM Computer Communication Review 34(2), 39–53 (2004)
6. Mirkovic, J.: D-WARD: DDoS network attack recognition and defense, PhD disseration prospectus. UCLA (January 23, 2002)
7. Thapngam, T., Yu, S., Zhou, W., Makki, S.K.: Distributed Denial of Service (DDoS) detection by traffic pattern analysis. Peer-to-Peer Networking and Applications, 1–13 (2012)

8. Kim, M., Na, H., Chae, K.-J., Bang, H., Na, J.-C.: A combined data mining approach for DDoS attack detection. In: Kahng, H.-K., Goto, S. (eds.) ICOIN 2004. LNCS, vol. 3090, pp. 943–950. Springer, Heidelberg (2004)
9. Oke, G., Loukas, G.: A Denial of Service Detector based on Maximum Likeli-hood Detection and the Random Neural Network. The Computer Journal 50(6), 717–727 (2007)
10. Rahmani, H., Sahli, N., Kamoun, F.: DDoS flooding attack detection scheme based on F-divergence. Computer Communications 35, 1380–1391 (2012)
11. Yu, J., Kang, H., Park, D., Bang, H.-C., Kang, D.W.: An in-depth analysis on traffic flooding attacks detection and system using data mining techniques. Journal of Systems Architecture 59(10-B),1005–1012 (2013)
12. Hwang, K., Cai, M., Chen, Y., Qin, M.: Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes. IEEE Transactions on Dependable and Secure Computing 4(1), 41–55 (2007)
13. Wang, F., Wang, H., Wang, X., Su, J.: A new multistage approach to detect subtle DDoS attacks. Mathematical and Computer Modelling 55(1), 198–213 (2012)
14. Oke, G., Loukas, G., Gelenbe, E.: Detecting denial of service attacks with bayesian classifiers and the random neural network. In: IEEE International Fuzzy Systems Conference, FUZZ-IEEE 2007, pp. 1–6. IEEE (2007)
15. Xylogiannopoulos, K., Karampelas, P., Alhajj, R.: Periodicity Data Mining in Time Series Using Suffix Arrays. In: Proc. IEEE Intelligent Systems IS12 (2012)
16. Xylogiannopoulos, K., Karampelas, P., Alhajj, R.: Exhaustive Patterns Detectio. In: Time Series Using Suffix Arrays (2012) (manuscript in submission)
17. Xylogiannopoulos, K., Karampelas, P., Alhajj, R.: Minimization of Suffix Arrays Storage Capacity for Periodicity Detection in Time Series. In: Proc. IEEE International Conference in Tools with Artificial Intelligence (2012)
18. Xylogiannopoulos, K., Karampelas, P., Alhajj, R.: Experimental Analysis on the Normality of pi, e, phi and square root of 2 Using Advanced Data Mining Techniques. Experimental Mathematics (2014) (in press)
19. Xylogiannopoulos, K., Karampelas, P., Alhajj, R.: Analyzing Very Large Time Series Using Suffix Arrays. Applied Intelligence (2014) (submitted for publication)
20. Manber, U., Myers, G.: Suffix Arrays: A New Method for On-Line String Searches. In: Proceedings of the first Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 319–327 (1990)