2017 16th IEEE International Conference on Machine Learning and Applications

DDoS Attack Modeling and Detection Using SMO

Salva Daneshgadeh
Department of Information Systems
Middle East Technical University
Ankara, Turkey
salva.daneshgadeh@metu.edu.tr

Nazife Baykal
Department of Information Systems
Middle East Technical University
Ankara, Turkey
nzbaykal@metu.edu.tr

Şeyda Ertekin
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
seyda@ceng.metu.edu.tr

Abstract*— Over the last decade, Distributed Denial of Service (DDoS) attacks have been employed to cause huge financial and prestige loss to different kinds of e-business. Attackers also target governmental websites using DDoS attacks as a new weapon in the world of cyber war. The importance of the issue has inspired many researchers from academia and the industry to provide solutions to this type of challenging attack. In this study, we simulated DDoS attacks in a virtual lab and then collected firewall logs from the Security Information and Event Management (SIEM) platform of a company in the field of security management solutions. We extracted 14 research features from firewall logs and applied a SMO algorithm to train our data using 10 fold cross-validation. The SMO with PolyKernel was able to create a prediction model without any false alarm. We also tested our model with two different datasets. This research is an ongoing multistep study. Future research will concentrate on online DDoS detection.

Keywords—DDoS attack detection; machine learning; feature extraction

I. INTRODUCTION

Denial of Service (DoS) attack aims to make information or services unavailable to its legal users. Distributed Denial of service (DDoS) attack employs many compromised systems named Zombies which can be controlled by an attacker. Subsequently, a group or network of these Zombies is developed to attack a single system. In 2014, SANS Institute surveyed more than 15,000 organizations inside and outside the USA from various industry sectors. The report revealed that 53% of DDoS attacks were less than 2 GBs in bandwidth. However, 13% of these attacks averaged more than 4 GBs. According to the survey, ports 80 (HTTP) and 443 (HTTPS) were the most common ports for DDoS attacks. Ports 53 (DNS), 445 (Microsoft Active Director and SMB) and 21 (FTP) were also targeted in some attack cases. DDoS attacks have normal traffic characteristics and the only factor which differentiates the malicious DDoS attack traffics from the normal ones is the volume of the flooding traffic during attack time. There are several types of denial of service attacks including: Smurf attack, Ping Flood, Ping of Death, TCP SYN flood and UDP flood.

A. Smurf attack

The attacker broadcasts a large number of ICMP echo requests with the spoofed IP addresses from the victim. Therefore, all reply messages target and overwhelm the victim's system.

B. Ping flood and ping of death

The attacker floods thousands of ping packets toward the victim. It is very similar to the Smurf attack.

C. TCP SYN flood

Initially, the attacker sends a SYN packet to start the handshaking process for a connection request. The server replies with an Acknowledgement packet, but the attacker does not send a SYN ACK packet to finalize the connection request. The attacker floods thousands of SYN packets and leaves all of them as half open connections. The server waits for the SYN ACK replies until it goes out of memory.

D. UDP flood

The attacker floods thousands of UDP packets toward the victim. It is very similar to the ping attack, but can be more effective if the size of the packets is huge.

E. Low-rate TCP targeted Denial of service attacks:

It is also called the Shrew attack. The huge number of packets are only sent in a short period of time, then this pattern is repeated for several intervals. This attack utilizes TCP congestion-control algorithm. When the TCP Reno detects loss (either by a timeout or by a receipt of a triple-duplicate ACK); the TCP waits for a period of time equal to a retransmission timeout (RTO), reduces its congestion window value (cwnd) to half (cwnd/2), sets a slow start threshold (ssthresh) and finally resents the packet. After each timeout, the amount of RTO doubles. If an attacker is able to calculate the

DOI 10.1109/ICMLA.2017.0-123

RTO and only flood the network just before the RTO timer expires, he/she can push the TCP to another 2* RTO waiting time. This type of DoS attack has a low average traffic rate which makes it difficult to identify [14].

F. Intrusion detection and Machine Learning:

Network Intrusion Detection System (NIDS) is an application which is responsible for monitoring network activities and investigating abnormal activities and behaviors. Signature based and Anomaly detection are the two major types of NIDS. The current taxonomy of Network Anomaly detection techniques is classified as statistics, clustering, classification and information theory. Statistical methods develop a profile of the normal data sample and take a baseline of the normal traffic and activities on the network. Network administrator is responsible for declaring a threshold to raise an alarm. Mean, Standard deviation, Multivariate, Markov process and Time Series Models are examples of statistical algorithms. A clustering approach is a type of unsupervised algorithm which does not require any labeled data sample. The main idea behind these techniques is the assumption of the high frequency of the normal data compared to the anomalies. K-means, Furthest First, Maximization Expectation, K-NN and hierarchical clustering are some examples of clustering algorithms. A classification approach is a type of supervised algorithm which requires data samples labeled as normal and attack. Therefore, network experts play a significant role in labeling the training dataset. Subsequently a predictive model and a normal traffic baseline is developed. Finally, any incoming traffic data with a deviation from the baseline is considered as anomaly. Support Vector Machine (SVM), Bayesian Networks, Logistic Regression, Decision Trees, Neural Networks are major examples of classification algorithms. Information theory based methods operate in an unsupervised mode and calculate some measures of information metrics such as entropy, relative entropy, conditional entropy and information cost in order to extract information content in data. These methods assume that outliers change the information content in a dataset. Therefore, they investigate data instances which alter information metrics.

II. RELATED WORKS

There are hundreds of studies in the field of Network Anomaly detection in the literature. The continuing and growing trend of the DoS and DDoS attacks encouraged us to concentrate only on the studies which applied clustering or classification methods to find DDoS and DoS attacks. DoS/DDoS attacks are collective type of anomalies. They can be distinguished by analyzing huge collections of a single connection or a request to the web server however, a single connection or a request is by itself legal.

Zhang Z, Li J, Manikopoulos CN, Jorgenson J, and Ucles J.[1] combine statistical methods and neural networks to propose a detection method. They created their own sample data by simulating the UDP flooding attack as an anomaly in their virtual lab. They reported fast coverage for their neural network classifier and low misclassification rate. The study of Portnoy L, Eskin E, and Stolfo S. [2] proposed width based single-linkage clustering. Different portions of the KDD'99 dataset was used as testing and training dataset and the performance of the algorithm was reported in each step. However, the number of actual and detected DoS attacks was not mentioned in any of the steps. Eskin E, Arnold A, Prerau M, Portnoy L, and Stolfo S. [3] applied K-NN, unsupervised SVM and cluster based estimation algorithms on the KDD'99 dataset. They compared the performance of algorithms for detecting anomalies, but they did not separately report what percent of the detected anomalies were DoS attacks. Arshad M, and Chan P. [4] applied the density and distance based clustering method to the DARPA dataset. They did a good job in reporting the performance of their algorithm, especially when it came to the DoS attack. Their study revealed that their algorithm was able to find 59% of DoS attacks (25 from 42). Guan Y, Ghorbani A, and Belacel N. [5] developed a method based on k-means which is called X-means. They ran their algorithm over KDD'99 and reported the overall performance of the algorithm. Leung K, and Leckie C. [6] proposed a density based clustering algorithm. KDD'99 was the evaluation instrument of their research as well. Nevertheless, they did not report the performance of their method regarding the DoS attack detection. Laskov P, Gehl C, Kruger S, and Muller K. [7] developed an incremental SVM (ISVM) algorithm using the U-RBF kernel function which is called RS-ISVM. In their work, a detection rate of 88.79% was stated for DoS attacks in the subset of KDD'99 using C-SVM with U-RBF kernel. Petrovic S, Alvarez G, Orfila A, and Carbo J. [8] proposed a method which combined Davies-Bouldin index of the clustering with the comparison of the centroid diameters of the clusters. They also used the KDD'99 dataset and demonstrated that their method is powerful enough to label data when there are not massive attack events in the dataset. Münz G, Li S, and Carle G.[9] applied k-mean (k=2) to the training dataset by assuming normal and attack traffic from two different clusters. They created a network traffic in their testbed and utilized the traffic generator npag to create the ping and UDP flood against port 53. They reported discovery of anomalies including one ICMP ping flood, but the number of actual and detected DoS attacks is not reported explicitly. YU W, and Lee H. [10] proposed an incremental learning method which was called incremental tree inducer (ITI). They stated the performance of ITI, K-mean+ ITI, SMO+ ITI for DoS detection on KDD'99 as 92.38%, 91.31% and 91.07% respectively. Poojitha G, Kumar K, and Reddy P. [11] applied neural network to train samples from KDD'99. Their method was able to simply feed forward neural networks trained by the back-propagation algorithm to classify the abnormal events. They reported the power of their algorithm to find all 1500 DoS attacks in the testing dataset. Su Y. [12] collected its own attack data using one laptop that sent DoS attacks against the victim machine in the LAN. The amount of traffic range was between 0-80 Mbps during the simulation. He initially applied MLBG clustering algorithm to reduce the amount of sample data. Afterwards, he employed K-NN algorithm and reported the overall accuracy of 96.25% in the case of 2-flod validation. Papalexakis EE, Beutel A, and Steenkiste P. [13] utilized the soft clustering to find different types of attacks in KDD'99. They reported an overall accuracy of 75% and 85% for normal and attack respectively. Ahmed M and Mahmood A.[15] applied the X-mean algorithm to detect anomalies in the DARPA dataset. The majority of the attack in

their selected subset of the DARBA dataset was the DoS attacks and they researched 97% accuracy to detect anomalies in the dataset. Ahmed M and Mahmood A. [16] proposed a collective anomaly detection method using a partitioned clustering technique. They also used the KDD'99 /DARPA datasets to train and test their method. They reported the ability of their algorithm to find all available DoS attacks in test data. The very recent study by Hoque M, Kashyap H, and Bhattacharyya D.K. [17] proposed a new DDoS detection framework which is implemented on software as well as hardware using the Field Programmable Gate Arrays (FPGA) device. Currently, the proposed method solely considers the DDoS attack detection as a 2-class problem. The proposed model creates the normal traffic profile during the analysis period. When a new input traffic instance comes, the attack detection module first computes the correlation value by analyzing the three distinct features of the incoming instance and normal profile. If the calculated correlation value surpasses the predefined threshold, the system generates an alarm

III. METHODOLOGY

Fig. 1, presents the way that this study was conducted.

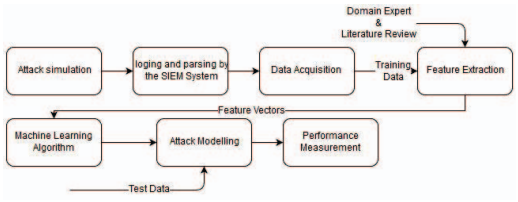

Fig. 1.The framework of the study

IV. DATA COLLECTION

We made a literature review in order to find the comprehensive datasets which include DoS and DDoS attacks. To the best of our knowledge the KDD cup 1999, DARPA 1998, CAIDA DDoS attack 2007, ISCX 2012, ADFA 2013, PREDICT 2014, DEFCON 2014, NSL-KDD 2014, KYOTO 2014, ICS Attack 2014, TUIDS 2015 are the available intrusion datasets. Most of the datasets include raw network packet information which are captured by tools like TCPdump and Wireshark. Unfortunately access to most of them are prohibited because of privacy issues. DARPA/KDD datasets are publicly available, but they are old fashioned databases which are still used in many of today's studies. We planned to make our analyses based on firewall logs. We designed a virtual lab as it can be seen in Fig. 2. We installed 3 virtual machines in our virtual environment: offensive system (Kali), victim system (Windows server 2012) and firewall (open source Pfsense). We also made an interface from firewall to our LAN where our SIEM is located in order to forward the firewall Syslogs to the SIEM and collect the parsed data from SIEM.

We used Hping3 to perform SYN flood, UDP flood and Smurf attacks toward the Windows Server 2012 from the Kali. Our attack packets have different data sizes and transfer rates. Additionally, we sent fragmented parts of large DDoS packets during the attack. On the other hand, we collected the normal traffic from the firewall logs of the SIEM. In order to make sure that there is not any attack in the firewall logs, network experts examined the logs manually. In addition, we applied clustering algorithms (for more details refer to Table I) to investigate the possible availability of attacks among firewall logs.

Subsequently, we combined attack and normal firewall logs for creating one training (27,090,739 total logs and 24,768,581 attack logs) and two testing datasets. Finally, the time-date attribute of attack logs was shifted to match the time-date attribute of normal firewall logs.

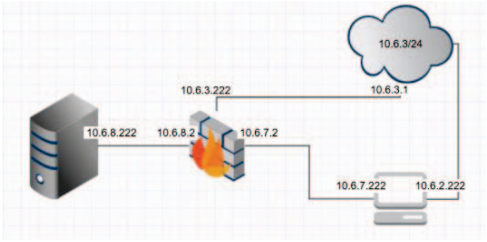

Fig. 2. Virtual lab architecture

V. FEATURE CONSTRUCTION AND EXTRACTION

We selected date-time, destination IP addresses, source IP addresses, destination ports, total bytes/packets sent, total bytes/packets received, protocol, duration and action (close, timeout) attributes from the firewall logs. As the DoS/DDoS are collective type anomalies, we needed to look at the behavior of logs in an aggregative manner. In a nutshell we developed the following feature vectors for every 24 hours (1440 seconds). Finally, we came up with 326,593 vectors (326,416 normal and177 attack):

$\vec{F}$ = (Time_Interval, Distinct_Destination_IP, Count_of_Source, Total_Receivedbytes, Total_Receivedpackets, Total_Sentbytes, Total_Sentpackets, Count_of_Distinct_Destination_Port, Count_Close, Count_Timeout, Average_Packet_Size_Received, Average_Packet_Size_Sent, Byte_Rate_Sent, Packet_Rate_Sent)

For example, $\vec{F}$ = (668, 10.100.190.100, 8, 2484, 27, 2944, 46, 2, 0, 8, 92, 64, 19, 23) is interpreted as the following:

In time interval 668, 8 connections were established to 10.100.190.100. All of 8 connections were timeout. In total, 46 packets sent to 10.100.190.100. The average size of packets which were sent were 64 bytes and the average rate of sent packets was 23 bytes/s.

VI. EXPERIMENTAL RESULTS

We used the open source Weka package as a machine learning software. Initially, we normalized all data in training and testing datasets. Consequently, we applied Furthest First clustering algorithm to investigate whether any normal vector was clustered with any attack vectors or not. As it can be seen in Table I, attack vectors are clustered separately except one

which was grouped by normal vectors. It was related to the part of the DDoS attack traffic which 3595 connections were established to the victim's machine with a time of 15:34. This number of connections was too low to consider this vector as an attack. Accordingly, we removed this vector from the training dataset and applied the Sequential Minimal Optimization (SMO) algorithm to the training data using 10 Folds Cross-Validation. SMO with PolyKernel produced the 100% detection performance without any false alarm rate.

TABLE I. RESULTS OF FF CLUSTERING ALGORITHM

K and seed values	Cluster 1	Cluster 2	Cluster3
K=2, seed= 10	326,592	1	-
K=3, seed= 10	326416	176	1

VII. PERFORMANCE MEASUREMENT

The huge difference between the number of normal and anomalous instances in the training dataset results is in the imbalance data problem. Frustrated accuracy is the most important problem which threads imbalance datasets. In the imbalance data samples, the models cleverly decide that the best thing to do is to always predict the class with the highest accuracy. There are several ways to combat imbalanced training data, such as changing the performance metrics and re-sampling. Table II, demonstrates the results of different performance measurements for the PolyKernel based SMO. All performance metrics depict the perfect ability of the model to classify both normal and attack events.

TABLE II. PERFORMANCE METRICS OF SMO

Performance Metrics[a]	Normal	Attack
TP Rate	1.00	1.00
FP Rate	0.00	0.00
Precision	1.00	1.00
Recall	1.00	1.00
F-Measure	1.00	1.00
MCC	1.00	1.00
ROC Area	1.00	1.00
PRC Area	1.00	1.00

[a] Tolerance parameter: 0.001, c= 1.00, Epsilon= 1.0E-12

We selected SMO as a classification algorithm, because of its fundamental advantageous such as the following:

A. *Decision boundary*

When SVM establishes a decision boundary, it only needs to keep the set of core points which are required to demonstrate the boundary. These data points are called support vectors. Therefore, most of the training data becomes redundant in keeping track of the decision boundary.

B. *None-linearly dependent data*

The real power of SVM is seen when data are none-linearly dependent. SVM take advantage of the kernel trick to transfer all training data to higher Hilbert space (where the number of features will be equal to number of training samples) and then investigate the relationship among data points.

C. *Finding global minimum*

SVM is good for finding the global optimum, as its optimization problem is convex optimization.

VIII. TESTING MODEL

We tested our classifier with two more testing datasets.

A. *Test_data1(2,635,981total logs & 520,099 attack logs)*

The corresponding dataset of feature vectors had 3 rows which were marked as attack. We increased the amount of attack vectors to 1536, using the EMOTE re-sampling algorithm, then applied the prediction algorithm. As it can be seen in the Table III, the performance of the classifier was 100%.

TABLE III. RESULTS OF SMO ON TEST_DATA1

Negative	Positive	Class
290215	0	Normal
0	3	Attack

B. *Test_data2 (2,392,673 total logs & 1,699034 attack logs)*

The corresponding dataset of feature vectors had 5 rows which were marked as attack. We tested the data with the prediction algorithm. As it can be seen in the Table IV, the classifier could detect the DDoS attack in 5 time intervals.

TABLE IV. RESULTS OF SMO ON TEST_DATA2

Negative	Positive	Class
89167	0	Normal
0	5	Attack

IX. CONCLUSION AND FUTURE WORKS

Network log files are very useful for understanding abnormal events in computer systems. Today, most of the organizations have a firewall as a network security system. We took advantage of the basic attributes in almost all firewall logs to create the best feature set which increases the detection rate. We also decreased the training dataset size by selecting the effective vectors of features which in turn reduces the training cost and time. We collected real firewall logs and prepared our own training and testing datasets. We applied SMO algorithm to our training data and developed a prediction model. Subsequently we used our model to predict data in testing datasets. We developed a model which is able to predict incoming DDoS attacks with high detection accuracy and no false positive alarms. Also, our model requires a small set of features and low training cost. We are currently working on an online DDoS attack detection method, using recursive kernel based algorithms in order to remove the continuous need for re-training the model to adapt to the ever-changing nature of network traffics. We also plan to develop a model to detect DDoS attacks by finding correlations among different logs of the SIEM system such as apache server, IDS, and IPS.

ACKNOWLEDGMENT

The authors would like to thank MAY Cyber Technology company for its contribution as a network security consulter and data provider. Their underlying virtual lab was used to simulate DoS/DDoS attacks during this study. In addition, parsed data was collected from their commercial SIEM product named "MAY Cyber SIEM". This study is supported through the "Network Anomaly detection systems" project and the results will be used to add a machine learning component to their "MAY Cyber Analytic Engine" product.

REFERENCES

[1] Z. Zhang, J. Li, CN. Manikopoulos, J. Jorgenson, J. Ucles, "Hide: A hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," in Proceedings of IEEE workshop on information assurance and security, pp.85-90, 2001.

[2] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion Detection with Unlabeled Data Using Clustering," in Proceedings of ACM CSS workshop on data mining applied to security (DMSA), 2001.

[3] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo," A Geometric Framework for Unsupervised Anomaly Detection," Applications of data mining in computer security, vol.6, pp. 77-101, 2002.

[4] M. Arshad, and P. Chan P, "Identifying outliers via clustering for anomaly detection," Florida Institute of Technology, Florida, Tech.Rep., June, 2003.

[5] Y. Guan, A. Ghorbani, N. Belacel, "Y-means: A Clustering Method for Intrusion Detection," IEEE CCECE 2003 Canadian conference on electrical and computer engineering, vol.2, pp.1083-1086, 2003.

[6] K. Leung and c. Leckie, "Un Supervised Anomaly Detection in Network Intrusion Detection Using Clusters," in Proceedings of the twenty-eighth Australasian conference on computer science, vol. 38, pp.333-42, 2005.

[7] P. Laskov, C. Gehl, S. Kr¨uger, and K.-R. M¨uller, "Incremental Support Vector Learning: Analysis, Implementation and Applications," Journal of Machine Learning Research, vol. 7, pp. 1909-1936, 2006.

[8] S. Petrovic, G. Alvarez, A. Orfila, and J. Carbo, "Labelling Clusters in An Intrusion Detection System Using a Combination of Clustering Evaluation Techniques,"in Proceedings of the 39th annual Hawaii international conference on System Sciences, vol.6, 2006.

[9] G. Münz,S. Li, and G. Carle, "Traffic Anomaly Detection Using Kmeans Clustering," InGI/ ITG Workshop MMBnet, 2007.

[10] W. Yu and H. Lee, "An Incremental-Learning Method for Supervised Anomaly Detection By Cascading Service Classifier And ITI Decision Tree Methods," in Proceedings of the Pacific Asia Workshop on Intelligence and Security Informatics, pp. 155-160, 2009.

[11] G. Poojitha, K. Kumar, and P. Reddy, "Intrusion Detection Using Artificial Neural Network," 2010 International conference on computing communication and networking technologies(ICCCNT), pp.1-7, 2010.

[12] M-Y. Su, "Using Clustering To Improve The KNN-Based Classifiers For Online Anomaly Network Traffic Identification," Journal of Network and Computer Applications, vol.34(2), pp.722–30, 2011.

[13] E. Papalexakis, A. Beutel, and P Steenkiste, "Network Anomaly Detection Using Co- Clustering," In:Proceedings of the 2012 international conferenceonadvances in social networks analysis and mining(ASONAM2012),ASONAM'12, IEEE Computer Society, Washington, DC,USA, pp.403-10. 2012.

[14] L. Xiao-ming, C. Gong, L. Qi, and Z. Miano, "A Comparitive Study on Flood DoS and Low-Rate DoS Attack," The Journal of China Universities of Posts and telecommunications. Vol. 19, pp. 116-121, June 2012.

[15] M. Ahmed and A. Mahmood, "Network Traffic Analysis Based On Collective Anomaly Detection,"in:2014 IEEE 9th conference on industrial electronics and applications (ICIEA), pp.1141-1146, 2014.

[16] M. Ahmed and A. Mahmood, "Novel Approach for Network Traffic Pattern Analysis Using Clustering-Based Collective Anomaly Detection," Annals of Data Science, vol.2(1), pp.111-130, 2015.

[17] N. Hoque, H. Kashyap, and D.K. Bhattacharyya, "Real-time DDoS attack detection using FPGA," Computer Communications, vol.110, pp. 48-58, 2017.