IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue, 7, March 2019 E-ISSN: 2347-2693

Application Layer Denial of Service Attack Detection using Deep Learning Approach

A.B. Mahagaonkar[1*], A.R. Buchade[2]

[1,2]Department of Computer Engineering, PICT, Pune, India

[*]*Corresponding Author: akash.mahagaonkar@gmail.com, Tel.: +91-7028012611*

Available online at: www.ijcseonline.org

***Abstract*—** Denial of Service attack, is one of the deadliest attacks of the Internet era. It's major objective is to prevent legitimate users from accessing services over a network. DoS attacks can be broadly classified into network layer and application layer attacks. In this paper focus is on detection of well-known HTTP based application layer DoS attacks. We have proposed an integrated solution for detection of both volumetric and non-volumetric HTTP based application layer DoS attacks. The proposed system uses an in-memory analytics mechanism to extract the input feature set from the live traffic. On the basis of its learning from the training phase the deep neural network identifies the attacker using the feature set. We have used the TensorFlow to build the deep neural network. We have built a conformation mechanism to further reduce false positives. The result reveals that the proposed system can achieve 99.92% classification accuracy with only 0.003% false positives.

***Keywords*—** Denial of Service (DoS) Attack, Neural Network, Machine Learning, Deep Learning, Supervised Learning, Network Security, Application Layer, TensorFlow.

I. INTRODUCTION

Denial of Service (DoS) attack has been in existence for a long time. There are two major types of DoS attacks, volumetric and non-volumetric attacks. A majority of volumetric attacks are network layer attacks. Volumetric attacks tries to consume entire network bandwidth of the victim. To achieve this attacker sends a large volume of packets to the victim. But this abnormal increase in traffic volume can be easily detected. As a result, it is easy to neutralize the network layer volumetric attack. The majority of application layer attacks are non-volumetric in nature. In this type the attacker exploits, protocol and system implementation vulnerabilities to carry out the attack. As these attacks don't consume large chunk of the network bandwidth, they bypass the traditional DoS attack detection systems. It has increased the popularity of application layer DoS attacks. Many recent reports suggest that not only the frequency but also the duration of application layer attacks has increased.

Many efforts have been made for detection of various application layer DoS attacks. To detect the attack some approaches require computationally expensive features, most of them can detect either volumetric or non-volumetric DoS attacks. Contemporary approaches also spent a large amount of time and energy in training phase. As a result one has to spend a considerably large period of time to retrain the model if they came across a new attack. These challenges motivated us to explore a solution based on computationally less expensive features.

Organization of the rest of the papers is as fallows. Section II analyzes the related work, Section III briefly describes the dataset, Section IV contains the architecture of proposed system, Section V presents the experimental results and Section VI concludes the paper.

II. RELATED WORK

H. Beitollahi et al. [2] have proposed a system that creates benign user reference profile using various statistical attributes. In this paper focus of authors is on detection of HTTP flood attack. The proposed system uses reference profile to calculate the connection score. The score of each attribute is calculated individually. Sum of all attributes is the final score of the connection. All connections with lower scores are considered as attack and are dropped. To create reference profile of the benign user the system requires private data of the website such as request rate of a legitimate user, uptime and downtime of the legitimate user, the type of the page being accessed, page access rate and popularity of the page being accessed.

M. Shtern et al. [3] have proposed a system to mitigate application layer low and slow attacks by utilizing Software Defined Infrastructure capabilities. The proposed system uses detection sensor to determine how suspicious traffic will affect the protected application's behaviour. They have created baseline performance metrics of the protected application by monitoring it's execution performance during non-attack period. In case of deviation from the baseline performance metrics, detection sensor notifies to automation controller, then the automation controller creates a shark tank and redirects traffic to it. A shark tank absorbs the damage from DDoS attacks. To create baseline performance metrics they have used parameters like workload, CPU utilization, CPU time, disk utilization, disk time and waiting time.

S. Yadav et al. [4] have proposed a system for detection of both volumetric and non-volumetric attacks. The proposed system learns abstract features to understand behaviour of benign users. They have trained their stacked autoencoder during non-attack period. The trained model detects the abnormal behaviour and flags it as attack. But the proposed system suffers from large false alarms.

K. Prasad et al. [5] have proposed an ensemble classifier for detection of volumetric application layer attacks. The proposed system creates tuples of traffic flow such that each tuple reflects distribution diversity from the other tuples and engaged to independent classifier for attack detection.

M. Najafabadi et al. [6] have proposed a method for detection of a volumetric application layer DDoS attack, HTTP GET. The proposed system is based on the concept of user behaviour anomaly detection. The system extracts user browsing behaviour instances from web server logs, where each instance represents resource access pattern of user during particular time interval. The proposed system utilizes Principle Component Analysis for the classification purpose.
A. Alsirhani et al. [7] have proposed system that utilizes naïve bayes, decision tree and random forest algorithms for detection of the volumetric DoS attack. The proposed system uses fuzzy logic to determine the next classification algorithm to be used. The selection of next classification algorithm is based on following three parameters. First is average of previously store accuracies, second is average of delays of algorithms and third is last minute traffic volume.
C. Kemp et al. [8] have employed 8 different classifiers for detection of non-volumetric slow read attacks. They have carried out slow read attack in 3 different configurations. They have used self-generated dataset to train their system. The dataset consists of values for 12 different features.

V. Katkar et al. [9] have proposed an intrusion detection system based on naïve bayes algorithm for detection of both volumetric and non-volumetric application layer DoS attacks. They have used self-generated dataset to train the detection algorithm. The system suffers from large false alarms.

III. BRIEF DESCRIPTION OF THE DATASET

In our experiment we have generated the dataset using the pcap file provided by the University of New Brunswick's Canadian Institute for Cyber Security [1]. The pcap file contains both volumetric and non-volumetric application layer DoS attacks.

Volumetric attacks are often referred to as flooding. They are characterized by a large volume of application-layer requests. Non-volumetric DoS attacks are characterized by small amounts of attack traffic transmitted strategically to the victim. In the pcap file generated application layer DoS attacks were intermixed with the attack-free traces from the ISCX-IDS dataset. The resultant pcap file contains 24 h of network traffic with total size of 4.6 GB. More details about the pcap file can be found in paper [1].

We have generated the dataset using the pcap file provided by Canadian Institute for Cyber Security. We have extracted following features for each communication between the client and the server. We have kept the window size to 1 second.

1. Total unique connections between the client and the server.
2. Total packets exchanged between the client and the server.
3. Total bytes exchanged between the client and the server.
4. Max packet length.
5. Min packet length.
6. Max TTL.
7. Min TTL.
8. Total Count of URG Flag.
9. Total Count of ACK Flags.
10. Total Count of PSH Flags.
11. Total Count of RST Flags.
12. Total Count of SYN Flags.
13. Total Count of FIN Flags.
14. Bytes exchanged per packet.
15. Class.

IV. METHODOLOGY

The architecture proposed in this paper [Figure 1] is based on Deep Feed Forward Artificial Neural Network. We have used TensorFlow a deep learning framework developed by Google for implementation of our deep neural network.
A typical FF-ANN consists of the input layer, hidden layers and the output layer. Inputs are fed through the input layer, output is generated at the output layer and hidden layers are placed between input and output layers.

A. The Attack Detection Module.

FF-ANN is a supervised machine learning algorithm. The detection module learns to detect the attack during the training phase. In this paper we have used the ReLU activation function at hidden layers and the Sigmoid activation function at the output layer.

The detection module was trained using carefully labeled attack and legitimate records. In this phase the detection module learns to classify the communication as attack or legitimate communication. During training phase the detection module tries to increase the prediction accuracy by minimizing mean square error in each epoch. Mean square error is square of difference between predicted output and actual output. Based on the error value, weights and biases are updated to decrease the error and increase the classification accuracy.

B. Network Traffic Monitor.

The network traffic monitor constantly captures the live network traffic and sends it to the feature extractor for the feature generation.

C. Feature Extractor.

For each communication the module extracts 14 features using raw network traffic forwarded by the Network Traffic Monitor. It performs standard scalar on each feature value before forwarding it to the attack detection module. We have implemented an in-memory analytics mechanism in this module to avoid the disk access latency.

Finally the proposed system examines each communication for 5 consecutive time. If a communication is classified as attack 3 out of consecutive 5 times by the detection module, it notifies the attacker IP address to the administrator.

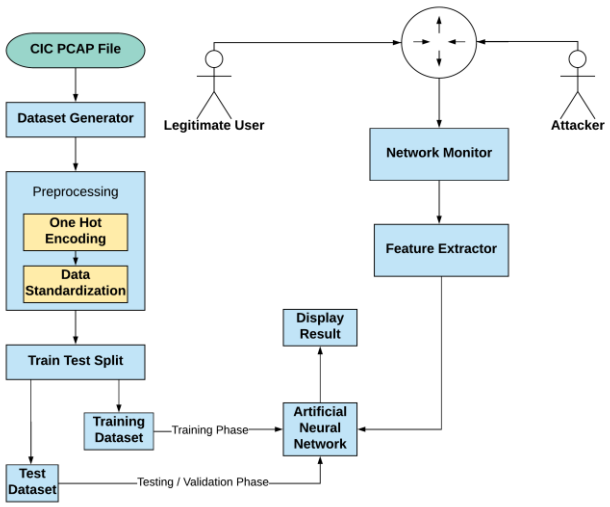

Figure 1. System Architecture

V. RESULTS AND DISCUSSION

While comparing the performance of the system, we have used following evaluation metrics. Precision, recall, sensitivity and specificity.

For the detection tasks, the terms true positives (Tp), true negatives (Tn), false positives (Fp), and false negatives (Fn) compares the results of the proposed system under test with evaluation data. The terms positive and negative refer to the detection algorithm's prediction and the terms true and false refer to, what that prediction corresponds to the external judgment.

Table 1. Experimental Results.

	Proposed System.
Total Legitimate User Records	50,455
Total Records Classified as Legitimate User (Tn)	50,426
Total Attack Records	769
Total Records Classified as Attack (Tp)	760
False Negative (Fn)	9
False Positive (Fp)	29

1. Precision.

Equation (1) presents the formula for precision calculation. In is the fraction of correctly identified attack records out of total identified attack records. It measures ability of the system to correctly identify the attack records.

$$\text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}} \quad (1)$$

2. Recall.

Equation (2) represents the fraction of correctly identified attack records out of total attack records present in the dataset.

$$\text{Recall} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}} \quad (2)$$

3. Specificity.

It is also called as the true negative rate. Equation (3) measures the proportion of correctly identified legitimate users records out of total legitimate user records present in the dataset.

$$\text{Specificity} = \frac{\text{Tn}}{\text{Tn} + \text{Fp}} \quad (3)$$

4. Accuracy.

Accuracy of the system is determined by the ratio of correct classification of attack and legitimate records out of total records. Equation (4) presents the formula for it.

$$\text{Accuracy} = \frac{\text{Tp} + \text{Tn}}{\text{Tp} + \text{Tn} + \text{Fp} + \text{Fn}} \quad (4)$$

Table 2. Performance Parameters.

Performance Parameters	**Proposed System.**
Precision	0.963244613
Recall	0.988296489
Specificity	0.99942523
Accuracy	0.99925616

Table 3. Performance comparison with contemporary systems.

Author	**Dataset**	**Acc. %**	**Approach**	**Focus**
H. Beitollahi et al. [2]	ClarkNet server	91.7	Connection Score approach based on Statistical Analysis	Volumetric Application Layer Attack
S. Yadav et al. [4]	NIT Trichy.	98.9	Stacked Autoencoder.	Non-Volumetric & Volumetric Application Layer Attack
K. Prasad et al. [5]	CAIDA Dataset	91.5	Ensemble of ML Classifiers	Volumetric Application Layer Attack.
M. Najafabadi et al. [6]	Florida Atlantic Uni.	99.6	Principal Component Analysis	Volumetric Application Layer Attack.
C. Kemp et al. [8]	Self-Generated Dataset	96.7	Multiple ML Classifiers	Non-volumetric Application Layer DoS Attack.
Proposed system	Self-Generated Dataset.	99.92	Deep Learning.	Volumetric and Non-volumetric Application Layer DoS Attacks.

The experimental result revels that the proposed system outperforms many contemporary application layer DoS attack detection systems in terms of classification accuracy. Nevertheless every system that we have compared with the proposed system has used different dataset and different approach for detection of the attack.

VI. CONCLUSION AND FUTURE SCOPE

In this paper we have proposed a system that provides an integrated solution for the detection of both volumetric and non-volumetric HTTP based DoS attacks. In the proposed system we have achieved 99.92% classification accuracy within 50 training epochs. Hence the system can be retrained within a short period of time for the detection of new attacks. The in-memory analytics feature provided in feature extractor module made the system scalable, this feature will be helpful while handling large volumes of service requests. The deep neural network generates very small fractions of false alarms. We have implemented a conformation mechanism to further reduce the false alarm. The proposed system will help organizations to detect well-known HTTP based attacks within fraction of seconds. As a result organizations will have enough time at their disposal to carry out the actions required for the mitigation of the attack. The proposed system will help organizations to avoid the degradation in quality of service, service downtime and reputation loss associated with it.

As the system can detect only known application layer HTTP based DoS attacks. In future one can extend the system's capabilities for detection of other application layer attacks that are not considered in this paper.

REFERENCES

[1] H. Jazi, H. Gonzalez, N. Stakhanova, A. Ghorbani, "*Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling*", Computer Networks, Vol. **121** pp. **25-36**, **2017**.

[2] H. Beitollahi, G. Deconinck, "*Tackling Application-layer DDo-S Attacks*", Procedia Computer Science, Vol. **10**, pp. **432-441**, **2012**.

[3] M. Shtern, R. Sandel, M. Litoiu, C. Bachalo, V. Theodorou, "*Towards Mitigation of Low and Slow Application DDoS Attacks*", In the Proceedings of the 2014 International Conference on Cloud Engineering, **USA,** pp. **604-609**, **2014**.

[4] S. Yadav and S. Subramanian, "*Detection of Application Layer DDoS Attack by Feature Learning Using Stacked Autoencoder*", In the Proceedings of the International Conference on Computational Techniques in Information and Communication Technologies, **India**, pp. **361-366**, **2016**.

[5] K. Prasad, A. Reddy, K. Rao, "*Ensemble Classiers with Drift Detection (ECDD) in Trafic Flow Streams to Detect DDOS Attacks*", Wireless Personal Communications, Vol. **99** pp. **1639-1659**, **2018**.

[6] M. Najafabadi, T. Khoshgoftaar, C. Calvert, C. Kemp, "*User Behavior Anomaly Detection for Application Layer DDoS Attacks*", In the Proceedings of the International Conference on Information Reuse and Integration, **USA**, pp. **154-161**, **2017**.

[7] A. Alsirhani, S. Sampalli, P. Bodorik, "*DDoS Attack Detection System Utilizing Classication Algorithms with Apache Spark*", In the Proceedings of the 2018 International Conference on New Technologies, Mobility and Security, **Spain**, pp. **1-7**, **2018.**

[8] C. Kemp, C. Calvert, T. Khoshgoftaar, "*Utilizing Netflow Data to Detect Slow Read Attacks*", In the Proceedings of the 2018 International Conference on Information Reuse and Integration, **USA**, pp. **108-116**, **2018**.

[9] V. Katkar, A. Zinjade, S. Dalvi, T. Bafna, R. Mahajan, "*Detection of DoS/DDoS attack against HTTP Servers using Naïve Bayesian*", In the Proceedings of the 2015 International Conference on Computing Communication Control and Automation, **India**, pp. **280-285**, **2015**.

Authors Profile

Mr. A. B. Mahagaonkar pursed Bachelor of Engineering from University of Pune, India in 2013. He is currently pursuing Master of Engineering in Pune Institute of Computer Technology, Pune, India. His main research work focuses on Neural Networks, Network Security, Machine Learning, Data Analytics.

Dr. A. R. Buchade presently working as assistant professor in the Department of Computer Engineering at Pune Institute of Computer Technology, Pune. He received B.E. and M.E. in Computer Engineering from WCOE, Sangli, India. He received Ph. D. in Computer Engineering from COEP, Pune, India. His research area includes Distributed System, Cloud Computing and Security.

