An Improved Model for Alleviating Layer Seven Distributed Denial of Service Intrusion on Webserver

IOP Conf. Series: Journal of Physics: Conf. Series **1235** (2019) 012020 doi:10.1088/1742-6596/1235/1/012020

An Improved Model for Alleviating Layer Seven Distributed Denial of Service Intrusion on Webserver

M Odusami[1*], S Misra[1,2] , E Adetiba[1] , O Abayomi-Alli[1], R Damasevicius[3], R Ahuja[4]

[1]Department of Electrical and Information Engineering, Covenant University Ota, Nigeria.

[2]Atilim University, Ankara Turkey.

[3]Kaunas university of Technology, Kaunas, Lithunia

[4]University of Delhi, New Delhi, India

* Modupe.odusami@covenantuniversity.edu.ng

Abstract- Application layer or Layer Seven Distributed Denial of service (L7DDoS) intrusion is one of the greatest threats that intrusion a webserver. The hackers have different motives which could be for Extortion, Exfiltration e.t.c Researchers have employed several methods to prevent L7DDoS intrusion especially using machine learning. Although Machine learning techniques has proven to be very effective with high detection accuracy, the approach still find it difficult to detect Hyper Text Transfer Protocol (HTTP) based botnet traffic on web server with high false positive rate. The adoption of deep learning based technique using Long Short Term Memory (LSTM) will alleviate this problem.

1. Introduction

One of the greatest threats on the internet today is the Denial of service (DoS) intrusion and is called a DDoS intrusion if it is launched from multiple distributed sources. The sophisticated nature of intrusion tools which are man- made computerized tools are used to carry out the DDoS intrusion on the target server [1]. DDoS intrusions arise from a large number of distributed hosts which are compromised and they launch the intrusion by flooding the target server with large amount of illegitimate flows [2]. This intrusion downtrend the target server in replying to legitimate users' request [3]. The intruder commonly known as bot-master infects systems that are vulnerable through command & control (C& C) by sending malware. One of the major tasks of botnet is DDoS in which the main goal is to drain the resources of the server which includes CPU, I/O bandwidth, sockets and memory etc. DDoS intrusion at layer seven is often referred to as Application layer DDoS intrusions which are more complicated to detect. There two main categories of protocols in application layer [4]. These protocols are service users' protocol and support protocols in which most L7DDoS intrusions exploit them. Examples of these protocols are: HTTP, File Transfer Protocol, etc [5]. Generally, L7DDoS intrusion can be classified into High-bandwidth intrusions with HTTP GET or POST requests and Low-bandwidth intrusions with slow HTTP POST or slow READ which are difficult to distinguish. In this paper, the focus is on both high rate and low rate DDoS. The rest of the paper is arranged as follows: Related work is described in Section 2. In Section 3, the proposed method to detect L7DDoS intrusion is presented. Section 4 explained the proposed implementation and the performance evaluation of the proposed method. In section 5, the conclusion of the paper is presented.

2. Related Works

Authors in [6] introduced Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) in which users have to click an image before accessing the resources. The major limitation of this approach is puzzle solving annoys normal users and can cause huge overhead. Giralte et al. [7] modeled normal user behavior using statistics, HTTP graph and HTTP caches. The DDoS intrusion is detected by considering the recurrence of HTTP activity performed on the target server. Authors in [8] proposed Real time frequency vector to detect L7DDoS intrusions by calculating the entropy of legitimate and illegitimate traffic. Wang et al. [9] employed a new relative entropy L7DDoS intrusion detection method by defining the click ratio of the web objects and suspicious sessions are detected using relative entropy. In [10] a trust management scheme to alleviate session flooding L7DDoS intrusion was introduced. Based on users browsing history, trust is calculated and higher trust value is assigned to a user that has better browsing history. Authors in [11] proposed a machine learning based detection system using Support Vector Machine classifier. Bots masking after the identity of legitimate users are exposed by utilizing the behavioral characteristics of network traffic. In [12] machine learning based on ensemble neural classifier to detect DDoS intrusion was employed. The results showed that the model detection accuracy is 99.4% with low false positive rate. Most of the incoming characteristics based techniques are machine learning in which neural network had been greatly explored. Due to the drawbacks of hand engineered feature extraction and false positive rates, deep neural network approach is utilized by some researchers. Authors in[13] proposed a stacked Auto- Encoder deep learning architecture to detect L7DDoS intrusion. The stacked auto encoder highly learned the features and logistic regression classifier was used for the classification. The results showed a detection accuracy of 98.99% with 1.27% false positive rate. This paper focuses on proposing an L7DDoS detection technique with high detection accuracy with improve false positive rate.

3. Methodology

In this study, Long short Term Memory (LSTM) deep learning approach with Tensor flow backend is proposed to develop an improved detection model for L7DDoS intrusion. A group of techniques that employs deep composition to learn high level feature illustration is referred to as deep learning and it is made up of several levels of neural network layers.

3.1 LSTM

LSTM is a type of Recurrent Neural Network (RNN), RNN process sequential information and has the ability to carry out the same process for each element of a sequence and the output depends on the previous estimations. Figure 1 indicates RNN states.

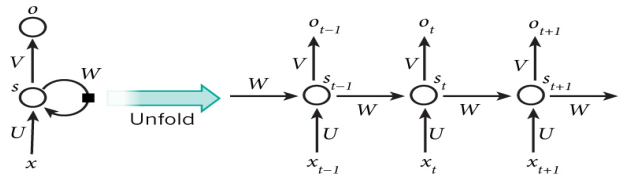

Fig 1: RNN state [14]

Figure 3 shows a RNN complete sequence in which input sequence is map into output sequence. The meanings of the parameters used in figure 4 above are explained as follows: U represents the weight of neurons between the input x and hidden state S, W represents the weight of neurons between hidden

states, and V represents the weight of neurons between hidden states S and the output O. The same parameters U,V and W are shared at the different hidden layers since the same task are carried out at all the hidden layers, only the inputs to the hidden layers are different. The weight is updated from one hidden layer to another using the back propagation. At time step t, the input is x_t, the hidden state is S_t. At time step t-1, the previous hidden state is S_{t-1}.The value of S_t is determined based on input at the current state and S_{t-1}.

$$S_t = f(Us_t + Ws_{t-1}). \quad (1)$$

Where f = tanh or ReLU (rectilinear function) and $S_{t-1} = 0$
The output at step t is O_t and it is given by equation 2.

$$O_t = \text{softmax}(Vs_t) \quad (2)$$

Error calculation for the RNN unfolding in figure 4 is shown in equation 3

$$E_{Total} = \sum \frac{1}{2}(Given_{output} - Actual_{output})^2 \quad (3)$$

The weight of neurons parameters are shared by all the time steps during the training of RNN

3.2 LSTM Algorithm

LSTM algorithm is designed based on the four interacting layers in the repeating module as shown in figure 2.The algorithm followed a four phase process as discussed below.

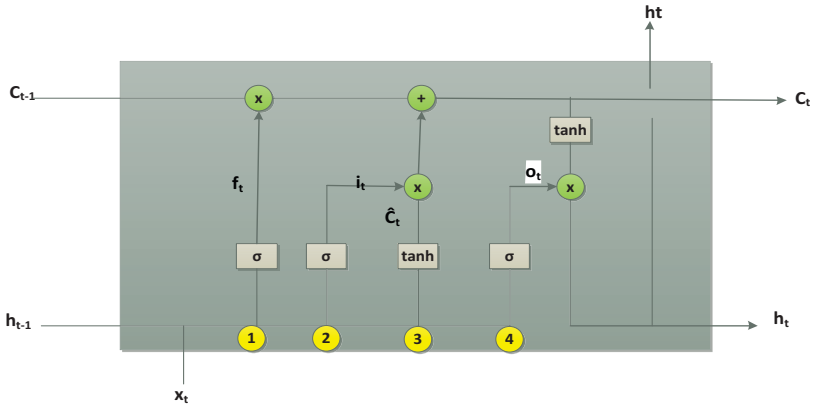

Fig 2. Interacting layers for LSTM Algorithm design Architecture

Phase 1:Based on the Sigmoid layer σ or forget gate layer, LSTM decides which information to discard from the cell state. Equation 4 represents phase 1or layer 1.

$$f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f) \quad (4)$$

Where: w_f = Weight, b_f = biases ,w_f and b_f remain the same for the four iterations.
Phase 2: This phase decides which information to store in the cell state and it is made up of two units. The input layer often called sigmoid layer decides on which information will be updated. Equation 5 described

the sigmoid layer.Further, a tanh layer converts the new information into vector form which can be included in the cell state. Equation 6 described the tanh layer.

$$i_t = \sigma(w_f \times [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{C}_t = \tanh(w_c . [h_{t-1}, x_t] + b_c) \quad (6)$$

Phase 3: This phase update the former cell state C_{t-1} into the recent cell state C. The former cell state C_{t-1} is amplified by f_t and the result from this vector multiplication is added to it x C_t as shown in equation 7

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (7)$$

Phase 4: This is the output phase and it selects on the information to output and before this selection is done, a sigmoid layer of the cell state makes the decision as shown in equation 8 and the cell state passes through a tanh and the result amplifies the output of the sigmoid gate as indicate in equation 9.

$$O_t = \sigma(W_0 \times [h_{t-1}, x_t] + b_0 \quad (8)$$

$$C_t = O_t \times \tanh(C_t) \quad (9)$$

3.3 Proposed Workflow

Proposed methodology workflow consists of four stages: collection of data, preprocessing of data, training/ testing data and model training, and classification of intrusion as shown in figure 3.

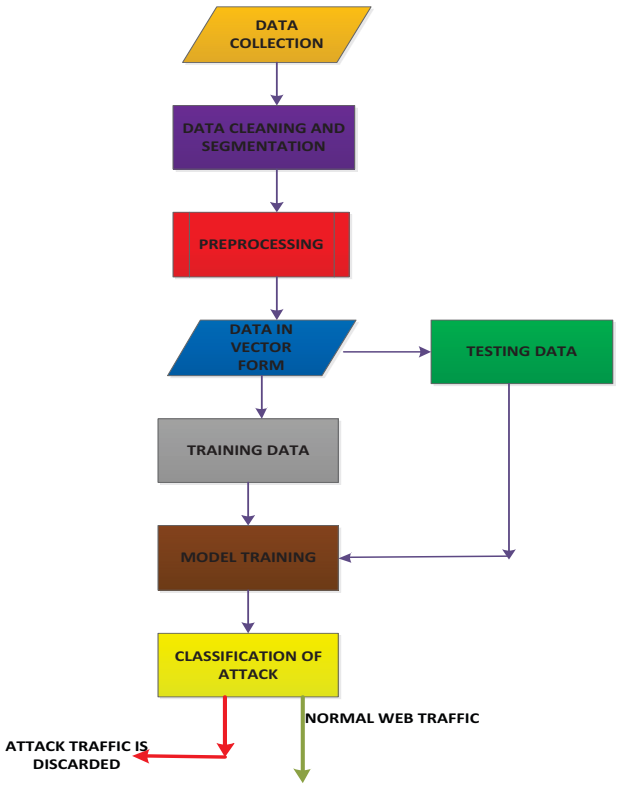

Fig 3: Proposed workflow

Data collection phase is very important. Centre for Applied Internet Data Analysis (CAIDA) DDoS attack dataset will be used to evaluate the proposed study.

4 Proposed Implementation and Performance metrics

The proposed methodology will be implemented using python 3.6 running on windows 10. Ananconda3 software will be installed, followed TensorFlow. Keras library a popular deep learning library will be used to create CPU based tensor flow. The tensor flow backend will be used to code the LSTM algorithm. The major aim of the proposed system is to detect L7DDoS intrusion with high detection accuracy and no false alarm rate. The following metrics will be used to evaluate the performance of the model: precision, specificity and accuracy.

4.1 Expected Contribution

The expected contributions for the proposed system are as follows: Develop a generic model for detecting low rate and high rate L7DDoS intrusion using LSTM, Provide a tensor flow code for the LSTM algorithm and evaluate the performance of the algorithm.

5. Conclusion

The application of Machine Learning and Artificial Intelligence has revolutionized the fields of Image Recognition and Speech Recognition. Majority of the researchers have applied machine learning in the prediction of future intrusions in which the system is trained and tested in such a way that it learns by observing the behavioral pattern associated with the traffic and classify the incoming traffic as an intrusion or normal. Existing methods in literature have been designed to counter either low-rate or high- rate L7DDoS intrusions but not for both. Also most of the existing techniques detect with high false positive rate. This research work will make significant contribution in addressing these challenges by adopting LSTM approach.

References

[1] Kumar PAR, Selvakumar S. Distributed denial-of-service (DDoS) threat in collaborative environment - A survey on DDoS intrusion tools and traceback mechanisms. Adv Comput Conf (2009).
[2] Saravanan, R.D., Loganathan, S., Shunmuganathan, S. and Palanichamy, Y., 2017. Suspicious Score Based Mechanism to Protect Web Servers against Application Layer Distributed Denial of Service Intrusions, International Journal of Intelligent Engineering & Systems, 10(4), pp 147- 156.
[3] Sreeram, I. and Vuppala, V.P.K., 2017. HTTP flood intrusion detection in application layer using machine learning metrics and bio inspired bat algorithm. *Applied Computing and Informatics*.
[4] Abliz, M., 2011. Internet denial of service intrusions and defense mechanisms. University of Pittsburgh, Department of Computer Science, Technical Report, pp.1-50.
[5] Kumar, G., 2016. Denial of service intrusions–an updated perspective. *Systems Science & Control Engineering*, *4*(1), pp.285-294.
[6] Mehra, M., Agarwal, M., Pawar, R., and Shah, D., 2011, February. Mitigating denial of service intrusion using CAPTCHA mechanism. In Proceedings of the International Conference & Workshop on Emerging Trends in Technology (pp. 284-287). ACM.
[7] Giralte, L.C., Conde, C., De Diego, I.M. and Cabello, E., 2013. Detecting denial of service by modeling web-server behavior. Computers & Electrical Engineering, 39(7), pp.2252-2262.

[8] Zhou, W., Jia, W., Wen, S., Xiang, Y. and Zhou, W., 2014. Detection and defense of application-layer DDoS intrusions in backbone web traffic. Future Generation Computer Systems, 38, pp.36-46.
[9] Wang, J., Yang, X. and Long, K., 2010, June. A new relative entropy-based app-DdoS detection method. In Computers and Communications (ISCC), 2010 IEEE Symposium on (pp. 966-968).
[10] Yu, J., Fang, C., Lu, L. and Li, Z., 2010. Mitigating application-layer distributed denial of service intrusions via effective trust management. *IET Communications*, *4*(16), pp.1952-1962.
[11] Singh, K., Singh, P. and Kumar, K., 2018. User behavior analytics-based classification of application layer HTTP-GET flood intrusions. *Journal of Network and Computer Applications*, *112*, pp.97-114
[12] Kumar, P.A.R. and Selvakumar, S., 2011. Distributed denial of service intrusion detection using an ensemble of neural classifier. *Computer Communications*, *34*(11), pp.1328-1341.
[13] Yadav, S. and Subramanian, S., 2016, March. Detection of Application Layer DDoS intrusion by feature learning using Stacked AutoEncoder. In *Computational Techniques in Information and Communication Technologies (ICCTICT), 2016 International Conference on* (pp. 361-366).
[14] WILDML, [Online] http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/,