Machine Learning Classifier for Detecting Slowloris Attacks

Suat Tarık Demirel

***Abstract*—Slowloris is a type of distributed denial of service (DDoS) attack that targets websites or servers by sending them with incomplete HTTP requests to make the server unresponsive. This paper proposes a machine learning (ML) classifier for detecting Slowloris attacks. The proposed classifier uses features that are extracted from the TCP and HTTP packets captured by Wireshark to train the model that can accurately identify Slowloris attacks. The results of the training and real-life scenery demonstrate that the proposed classifier can successfully detect these attacks. The proposed classifier could be used to create an effective tool to protect online services from Slowloris attacks.**

***Index Terms*—Slowloris, DDoS, ML Classifier, Feature extraction**

I. Introduction

DDoS attacks are a type of cyberattack that aims to increase the traffic volume of a website or server in order to make them unavailable to legitimate users. Usually, individual devices referred to as bots were used to create this high volume since these bots are legitimate Internet Devices and it can be difficult to distinguish these attackers' requests from legitimate users' requests. In this way, it has become a major concern for organizations, since it can cause significant disruption to online services and cause financial losses.

Slowloris is a type of DDoS attack that targets websites or servers by repeatedly sending partial HTTP requests to fill up the server's maximum connections and resources. It opens multiple connections to the target and leaves the connection open. In this way, the server's maximum concurrent pool is filled and legitimate users' connection attempts will be denied. This attack allows a single machine to take down a website or server by itself. This attack causes the server to become unresponsive or it could significantly increase CPU and memory usage.

For preventing Slowloris attacks several methods are proposed such as limiting the number of connections for a single IP or using an intrusion detection system to detect the attack. In this paper, it was proposed a machine-learning classifier for detecting Slowloris attacks by using TCP and HTTP packets captured by Wireshark. Features from the TCP and HTTP packets are extracted and used to train an ML model that can accurately identify Slowloris attacks.

II. Related Work

In the literature, researchers have proposed and evaluated various methods for detecting and preventing the Slowloris attacks. For instance, Giralte [1] used consecutive analyses named as statistics, HTTP graphs, and HTTP path caches in order to detect suspicious activity. They proposed that Slowloris attacks can be detected with the HTTP path cache analysis. Aiello [2] proposed a similarity-based approach to detect Slowloris attacks. They identified some protocol dependent parameters. Then, statistically, analyzed these values to distinguish different network scenarios. In this work, we also tried to extract features from both protocols, the length of the packets, and the time to classify and distinguish different network scenarios.

III. Method

A. Data and Preprocessing

In order to train a machine-learning classifier a wide dataset that includes both plenty of slow loris attacks and regular network traffics. The data sets contain traffic in and out of the web server of the Student Union for Electrical Engineering (Fachbereichsvertretung Elektrotechnik) at Ulm University was used for training. It can be retrieved from GitHub - vs-uulm/2017-SUEE-data-set.

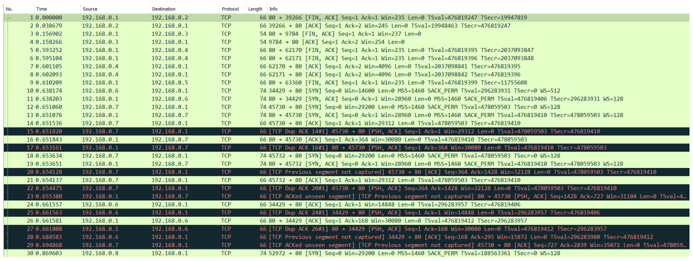

Fig. 1: Sample portion of the training data from the Wireshark window.

The data set includes TCP and HTTP packets of both Slowloris attacks and regular traffic packets. We only take TCP and HTTP packets in specified time intervals for training. We take several timelines of each attack flow and a similar number of regular flow data to get a more balanced dataset. Each data consist of 30 packets between two different IPs. Source, Destination, and Info columns are dropped since they are not in the features list that we want to train on. Also, a new column named as the label is added by looking at the source IP. This column can either be 0 (regular traffic) or 1 (attack traffic). The result data is saved in CSV format for easy usage.

B. Model Architecture

The model architecture consists of several layers. The first layer is the feature layer which is used to create feature columns from the input data. We have three distinguished feature columns as Time, Protocol, and Length. Each feature column is converted to an embedding columns structure of TensorFlow and passed to the feature layer.

The couple following layers are dense layers with 128 neurons each. The Relu activation function was used on each of these layers.

The last two layers are the dropout layer and a dense layer with one single neuron to get an output. The Dropout layer is used to prevent overfitting by randomly setting a fraction of input units to 0. The model is compiled with an Adam optimizer, binary cross-entropy loss function, and accuracy metric from TensorFlow.

Finally, end of the training the model is saved to make predictions.

IV. Training and Evaluation

A. Training

The dataset that was preprocessed and saved as CSV was loaded as a data frame of the pandas library. Then dataset divides into three different subsets as train dataset, the train validation dataset, and the test dataset by using the sci-kit-learn library, and convert to the TensorFlow data. Three feature columns are created by using embedding columns of TensorFlow. Then, the model was trained a couple of times to select the best hyperparameters for the best results. The best-trained model with a testing accuracy of about 75% is saved. All implementations were done with pyhton3 and TensorFlow.

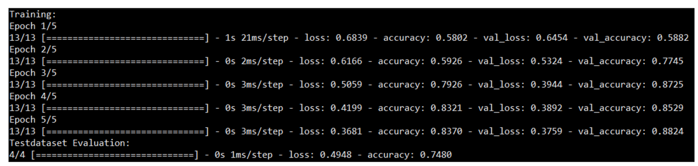

Fig. 2: Example evaluations of training process.

B. Evaluation

In order to evaluate our pre-trained model, a custom setup was created. Since preprocessing of data only take packets from IP pairs and Slowloris attacks can be done from one machine easily, our setup consists of only two machines. Two machines are created to simulate the simple Slowloris attack. Since it does not require to have two physical machines, the VMware environment was used to create a virtual machine that has a ubuntu os attacker. The windows11 os host machine behaves as a server in this setup. The NAT network configuration is selected from VMware for the virtual machine. This configuration allows virtual machines to communicate with the host machine.

For attacking SlowHTTPTest tool is used. It can be retrieved from GitHub - shekyan/slowhttptest: Application Layer DoS attack simulator. It is very easy to use. For the server side, in order to receive HTTP requests and simulate a normal web server, http.server 80 command of python was used. It can create a simple HTTP server that implements the basic security checks we need.

In order to monitor these attacks and gather data for the prediction Wireshark was used. Wireshark is one of the most known free and open-source packet analyzers. It captured the attack, filtered the TCP and HTTP packets, and saved the flow in CSV format. Its packet consists of time, source, destination, protocol, length and info columns. Since our model does not need the source, destination and info columns, then our preprocessor is used to create prediction data. In the last step, the pre-trained model is used to make the prediction. In this scenery, our model can successfully classify the traffic as an attack from our custom network.

C. Results

We implemented and evaluated the model that has been described in the previous sections. As a result of the training, the model achieved a training accuracy of approximately 75%. In order to evaluate the model in real-life scenery, we created a custom network and made an attack. The model successfully classified the network that has been captured from this attack scenery as an attack.

D. Future Work

As a future work, the model can be trained with a more comprehensive dataset to achieve a higher testing accuracy. Also, different types of DDoS attack flows can be used to train the model to evaluate its performance for detecting different kinds of DDoS attacks. Moreover, this model can be used to classify attacks in real-time systems by pipelining the model to real-time monitoring tools.

V. Conclusion

In this paper, we proposed a machine-learning classifier to detect Slowloris attacks by investigating TCP and HTTP packets. We implemented and evaluated our methods by using a real-life attacking scenery.

References

[1] Luis Campo Giralte, Cristina Conde, Isaac Martin de Diego, Enrique Cabello, Detecting denial of service by modelling webserver behaviour, 2013

[2] Maurizio Aiello, Enrico Cambiaso, Silvia Scaglione, Gianluca Papaleo , A Similarity Based Approach for Application DoS, 2013

[3] Thomas Lukaseder, Shreya Ghosh, Frank Kargl, Mitigation of Flooding and Slow DDoS Attacks in a Software-Defined Network, 2018

[4] Z. Wang, H. You, J. Chen, Y. Zhang, X. Dong and W. Zhang, Prioritizing Test Inputs for Deep Neural Networks via Mutation Analysis, 2021

[5] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, X. X. Zhu, A Survey of Uncertainty in Deep Neural Networks, 2022