International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 4 ǁ 2021 ǁ PP. 58-62

A Reliable & Robust DDOS Attack Detection in IOT Using Neural Network

ABSTRACT:
Internet-of-things (IOT) plays a prominent role in the digital revolution . The rapid development of IoT leads to various emerging cybersecurity threats. This is because IoT devices are often limited in computing capability and energy, making them particularly vulnerable to adversaries Therefore, detecting and preventing attacks in IoT networks have to be noticed by the people in the industry.There are many attacks takes place out of them distributed denial-of-service(DDoS) attack is most challenging.
A distributed denial-of-service(DDoS) attack is a malicious attempt to disrupt normal flow of targeted server,service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.
Generally, these attacks work by drowning a system with requests for data. This could be sending a web server so many requests to serve a page that it crashes under the demand, or it could be a database being hit with a high volume of queries. The result is available internet bandwidth, CPU and RAM capacity becomes overwhelmed.
This paper presents Distributed denial-of-service attack detection using Neural network. The main contributions of this project are Data Analysis, Dataset Preprocessing,
Training the Model,Testing of Dataset.This method will produce better results compared to other techniques.

Date of Submission: 26-03-2021 Date of acceptance: 09-04-2021

I. INTRODUCTION

Application Layer Distributed Denial of Service (DDoS) attacks are very challenging to detect and mitigate. The various possible application layer attacks are HTTP flooding, XML attack, DNS attacks, etc. The most common and renowned application layer attack is HTTP flooding. The HTTP flooding detection and mitigation is an interesting research topic in computer networks.

There are various research solutions proposed by validating against HTTP flooding; using tools such as Golden Eye, LOIC, proprietary tools, etc. HTTP flooding attacks generated using any existing tools may not exhibit similar characteristics of the real time HTTP flooding attack.

Various methods were used to defend these attacks based on distributed schemes with certain difficulties to count the packets or duplicates sent by a node. This is due to lack of communication infrastructure. Two limits are used to mitigate packet flood and replica flood attacks, respectively. Violation of both the limits can be easily noticed by claim-carry-and- check. The inconsistency check against full claims is trivial. This is designed to work in a distributed system. Moreover, it allows tolerating a little amount of attackers for collision.

II. LITERATURE SURVEY

[1] Title: SDN-Assisted Slow HTTP DDoS Attack Defense Method .Authors: KIWON HONG , YOUNJUN KIM , CHOI AND JINWOO PARK. Published Year: 2017.Description: A Slow HTTP Distributed Denial of Service (DDoS) attack causes a web server to be unavailable, but it is difficult to detect in a network because its traffic patterns are similar to those of legitimate clients. In this paper, we propose a network-based Slow HTTP DDoS attack defense method which is assisted by a Software-Defined Network (SDN) that can detect and mitigate Slow HTTP DDoS attacks in the network. Simulation results show that the proposed Slow HTTP DDoS attack defense method successfully protects web servers against Slow HTTP DDoS attacks.Efficiency: Defeat application-level DDoS attacks,Use cross-layer traffic analysis,Bound to a variety of transport protocols.**[2]**Title:Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges Authors: QIAO YAN, F.RICHARD YU QINGXIANG GONG, JIANQIANG LI .Published Year: 2015 Description: In this paper, we discuss the new trends and characteristics of DDoS attacks in cloud computing, and provide a comprehensive survey of defense mechanisms against DDoS attacks using SDN. In addition, we review the

studies about launching DDoS attacks on SDN, as well as the methods against DDoS attacks in SDN. To the best of our knowledge, the contradictory relationship between SDN and DDoS attacks has not been well addressed in previous works. This work can help to understand how to make full use of SDN's advantages to defeat DDoS attacks in cloud computing environments and how to prevent SDN itself from becoming a victim of DDoS attacks, which are important for the smooth evolution of SDN-based cloud without the distraction of DDoS attacks. Efficiency: It is cost effective, by allowing reuse of information extracted during detection,It makes no compromise of QoS, Reduces the consumption of hardware resources.**[3]**Title: Botnet in DDoS Attacks: Trends and ChallengesAuthors: NAZRUL HOQUE, DHRUBA K BHATTACHARYYA AND JUGAL K KALITA. Published Year: 2015 Description: Botnets pose a major threat to network security as they are widely used for many Internet crimes such as DDoS attacks, identity theft, email spamming and click fraud. Botnet based DDoS attacks are catastrophic to the victim network as they can exhaust both network bandwidth and resources of the victim machine. This paper presents a comprehensive overview of DDoS attacks, their causes, types with a taxonomy and technical details of various attack launching tools. A detailed discussion of several botnet architectures, tools developed using botnet architectures and pros and cons analysis are also included. Furthermore, a list of important issues and research challenges is also reported in the paper. Efficiency: Integrates multiple traceback mechanism with customization support, Effectively block Slow HTTP DDoS attacks, allowing a web server to sustain its normal operation, Supports distributed architecture.**[4]** Title: DDoS Tools: Classification, Analysis and Comparison Authors: BHARTI NAGPAL, PRATIMA SHARMA, NARESH CHAUHAN, ANGEL PANESAR. Published Year:2015 Description: In the last few years, it is recognised that DDoS attack tools and techniques are emerging as effective, refined, and complex to indicate the actual attackers. Due to the seriousness of the problem many detection and prevention methods have been recommended to deal with these types of attacks. This paper aims to provide a better understanding of the existing tools, methods and attack mechanism. In this paper, we commenced a detailed study of various DDoS tools. This paper can be useful for researchers and readers to provide the better understanding of DDoS tools in present times.Efficiency: Detecting either low-rate or high-rate DDoS attacks,can leverage network-wide knowledge of its own network to detect DDoS attacks through techniques such as traffic pattern analysis, or machine learning,Achieved competitive performance on various datasets.

III. PROPOSED SYSTEM

The proposed system presents a period-based defense mechanism (PDM) scheme is based on the periods and uses a blacklist to efficiently prevent the data flooding attack, by checking the data packet floods at the end of each period in order to enhance the throughput of burst traffic. Therefore, it can guarantee the Quality of Service (QoS) of burst traffic. As a result of which many data packets are forwarded at a high rate for the whole duration.

Flood attacks are launched by malicious or selfish nodes. Malicious nodes, which can be the nodes purposely setup by the opponent or subverted by the opponent via mobile phone worms begin attacks to congest the network and misuse the resources of other nodes. Selfish nodes may also develop flood attacks to increase their communication throughput. In DTNs, a single packet usually can only be carried to the destination with chances smaller than 1 due to the opportunistic connectivity. If a selfish node floods many replicas of its own packet, it can increase the probability of its packet being delivered, since the delivery of any replica means successful delivery of the packet. With packet flood attacks, selfish nodes can also boost their throughput, although in a subtler manner.

In the proposed Single-copy routing after sending a packet out, a node deletes its own copy of the packet. Thus, each packet only has one copy in the network. In the proposed Multicopy routing to the source node of a packet sprays a certain number of replicas of the packet to other nodes and each copy is separately routed using the single-copy strategy. The maximum number of copies that each packet can have is set.

In the proposed, Propagation routing (when a node locates it appropriate (according to the routing algorithm) to send a packet to another encountered node, it replicates that packet to the encountered node and keeps its own fake. There is no preset limit over the number of copies a packet can have. In Propagation, a node duplicates a packet to another encountered node if the latter has more regular contacts with the destination of the packet.

IV. SYSTEM ARCHITECTURE

Graphical representation of predicting the ddos attack.Here the flow chart shows how the hhtp requests are filtered and formated for the analyisation of flooding operation and atlast detecting the attackers activity .

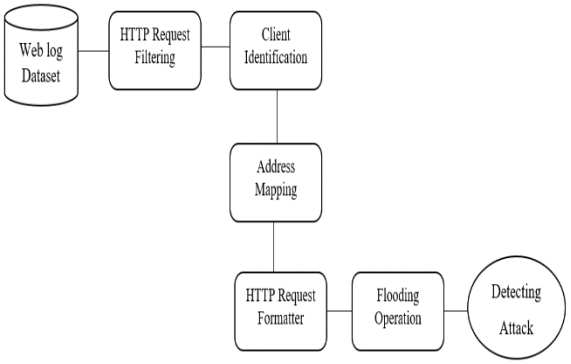

Fig 1. Block Diagram

V. MODULE DESCRIPTION

EXPLORATORY DATA EVALUATION

Exploratory Data Analysis (EDA) is the first step in your data analysis process. Here, you make sense of the data you have and then figure out what questions you want to ask and how to frame them, as well as how best to manipulate your available data sources to get the answers you need. You do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in your existing data, using visual and quantitative methods to get a sense of the story this tells.

Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and data scientists but can be very informative about a particular business.

EDA is performed in order to define and refine the selection of feature variables that will be used for machine learning. Once data scientists become familiar with the data set, they often have to return to feature engineering step, since the initial features may turn out not to be serving their intended purpose. Once the EDA stage is complete, data scientists get a firm feature set they need for supervised and unsupervised machine learning.

PRE-PROCESSING

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. Obviously you could remove the entire line of data but what if you are unknowingly removing crucial information? Of course we would not want to do that. One of the most common idea to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data.

The library that we are going to use for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data.

Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import test_train_split from model_selection library of scikit

FEATURE ENGINEERING

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. For basic guidance, you can refer to the following table for defining correlation co-efficients.

Pearson's Correlation: It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1.

LDA: Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

ANOVA: ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

Chi-Square: It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution

PREDICTION

Once training is complete, it's time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

A good rule of thumb I use for a training-evaluation split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If you have a lot of data, perhaps you don't need as big of a fraction for the evaluation dataset.

Once you've done evaluation, it's possible that you want to see if you can further improve your training in any way. We can do this by tuning our parameters. There were a few parameters we implicitly assumed when we did our training, and now is a good time to go back and test those assumptions and try other values.

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression.

VI. SYSTEM IMPLEMENTATION

HARDWARE ENVIRONMENT

Hardware	Minimum Requirement
Disk Space	32 GB or more,10 GB or more for Foundation Edition
Processor	1.4 GHz 64 bit
Memory	512 MB

Display (800 × 600) Capable video adapter and monitor

SOFTWARE ENVIRONMENT

BACKEND TECHNOLOGIES

Python, Numpy, Sci-learn, Jupyter notebook

FRONTEND TECHNOLOGIES

WebTechnologies, Bootstrap

VII. SCREENSHOT

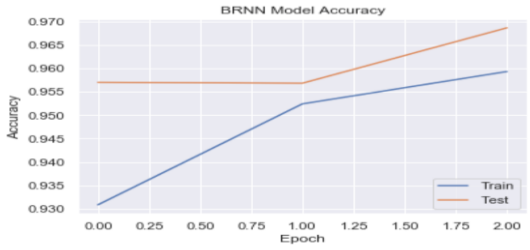

Fig 2 .BRNN Model accuracy

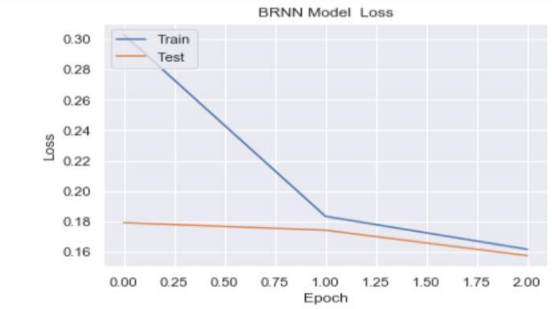

Fig 3 .BRNN Model loss

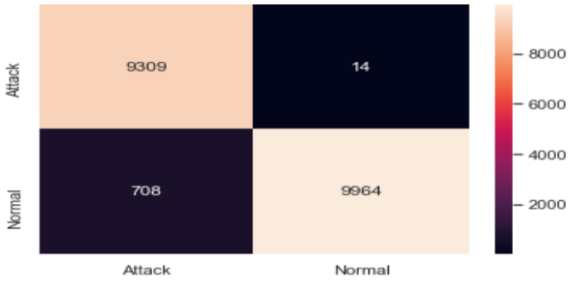

Fig 4 .Output

VIII. CONCLUSION

Reduce the attacks by employing rate limitations and probablistically detect the number of packets and the Long-short term memory algorithm is used to detect the approximate counting of packets which are violating the rate limits . This works are implemented in distributed manner. They easily reduce the throughput of burst traffic by comparing with the simple threshold . This is achieved by using proposed scheme and more over it is better than old scheme.

REFERENCES

[1]. Compagno, & al., "Poseidon: Mitigating interest flooding ddos attacks in named data networking," in Local Computer Networks (LCN), Intl' Conf. on. IEEE, 2013, pp. 630– 638.
[2]. Afanasyev, & al., "Interest flooding attack and countermeasures in named data networking," in IFIP Networking Conference. IEEE, 2013, pp. 1–9.
[3]. Ghali, & al., "Closing the floodgate with stateless content-centric networking," in 2017 26th International Conference on Computer Communication and Networks (ICCCN), July 2017, pp. 1–10.
[4]. T. Zhi, H. Luo, and Y. Liu, "A gini impurity-based interest flooding attack defence mechanism in ndn," IEEE Communications Letters, vol. 22, no. 3, pp. 538–541, March 2018.
[5]. Y. Xin, & al., "Detection of collusive interest flooding attacks in named data networking using wavelet analysis," in IEEE Military Communications Conference (MILCOM), Oct 2017, pp. 557–562.
[6]. Yi, & al., "A case for stateful forwarding plane," Computer Communications, vol. 36, no. 7, pp. 779–791, 2013.
[7]. K. Wang, & al., "On the urgency of implementing Interest NACK into CCN: from the perspective of countering advanced interest flooding attacks" in IET Networks, vol. 7, no. 3, pp. 136–140, 2018.
[8]. S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), April 2016, pp. 164–16
[9]. Amandeep Verma Manpreet Singh Gujral "A Comprehensive Appraisal of Ad hoc Networks" International Journal of Computer Applications (0975– 8887) Volume 49– No.22, July 2012.
[10]. Shahanaz Begum I, Geetharamani G, "DDoS Attack detection and Prevention in Private Cloud Environment ",International Journal of Innovations in Engineering and Technology (IJIET), Vol.7 Issue.3, Oct 2016, pp. 527- 531
[11]. S. Umarani, D. Sharmila, ''Predicting Application Layer DDoS Attacks Using Machine Learning Algorithms", International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol.8, No.10, 2014, pp. 1912-1917