Analysis and Derivation of Optimum Data-Driven Approach for Detecting DDoS Attacks

R. Sathya[1], R Siva Sandeep Reddy[2], P Sandeep[3], R Sai Kiran[4]

[1] Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science Technology, Chennai, Tamil Nadu, India.

[2] Student, Department of Computer Science and Technology, SRMIST, Chennai, Tamil Nadu, India

[3] Student, Department of Computer Science and Technology, SRMIST, Chennai, Tamil Nadu, India

[4] Student, Department of Computer Science and Technology, SRMIST, Chennai, Tamil Nadu, India

SRM Institute of Science and Technology, Chennai.

Abstract: DDoS (Distributed Denial of Service) attacks at the application layer are exceedingly difficult to detect and minimise. HTTP flooding, XML attacks, DNS attacks, and other application-layer attacks are all possible. HTTP flooding is the most well-known and well-known application-layer attack. In PC organisations, HTTP flooding detection and relief is a fascinating research subject. Various approaches based on distributed networks with some problems counting packets or redundant submissions sent from a malicious device are used to protect against these attacks. Owing to a lack of communication equipment, this is the case. Two limitations are used to mitigate all packet flood and imitation flood attacks. Claim-carry-and-check can quickly detect violations of both limits. The search for inconsistency against full statements is easy. This was created with a distributed system in mind. Furthermore, it allows for a small number of attackers to collide. A new vulnerability known as Ad Hoc Flooding Attack triggers a denial of service when used by all on-request ad hoc networks routing protocols. To preoccupy bandwidth and clog up the link, the malicious user either transmits a substantial percentage of route request packets for devices that are not present in networks or delivers a large number of data packets.

Keywords: Distributed Denial of Service (DDoS), Hypertext Transfer Protocol (HTTP), Domain Name System (DNS).

1.Introduction

The internet has become ubiquitous in recent years. The systems interact with one another through a variety of networks. A network is made up of n processors, routers, servers, and other devices. Both legal and unauthorised users (hackers) can be found on a network. A hacker is someone who illegally accesses and exploits the data of others. A hacker may use a variety of methods to accomplish this. Active and passive attacks are the two types of tactics used. Hackers do not change resources in passive attacks; instead, they sit back and evaluate all data. On the other hand, active attacks include hackers altering data and preventing a user from taking a specific action [1]. DDoS attacks are disruptive attempts to interrupt the daily traffic of a targeted worker, government, or organisation by flooding the victim or its adjacent infrastructure with Internet traffic. Since DDoS attacks use a large number of compromised Computer hardware to attack web traffic, they are viable.

PCs and additional network devices, such as IoT devices, are examples of underutilised machines. A DDoS attack resembles a sudden gridlock that shuts down the highway to an undeniable degree, preventing normal traffic from reaching its destination. The impact may range from minor annoyances caused by faulty administrations to the complete disconnection of entire websites, applications, or even entire businesses. There are three types of Distributed Denial of Service attacks. 1. Volume-based exploits will utilise massive amounts of false traffic to overload an asset, such as a website or a worker. ICMP, UDP, and satire package flood attacks are among them. A volume-based exploit's size is calculated in bits per second (bps). 2.DDoS attacks on the network or organisation layer send large packets to targeted organisation foundations and board computers.

SYN floods and Smurf DDoS, for example, are examples of convention assaults. In packets per second (PPS), their size is calculated. 3. Flooding applications with maliciously generated demands is how application-layer attacks are launched. Identifying DDoS attacks can be challenging. The most evident result of the DDoS attack is that either a website or administration becomes unavailable or moderately. Further investigation is usually needed because different factors, such as an actual increase in rush hour gridlock, can trigger comparative execution issues. Any of these signs of a DDoS attack can be identified through the use of traffic analysis tools.

1. Untrustworthy traffic measurements that start with one IP address or a set of IP addresses. 2. The increase in user traffic matches a specific profile, such as type of device, metadata, or user agent. 3. The desire for one page or destination was surprisingly strong. 4. Unusual traffic form, such as peaks at odd times during the day or variations, seem to be out of the ordinary. Other, more obvious indicators of a DDoS attack differ considerably on the attack type. The aim of any form of attack is the same: to make online resources slow or unresponsive. DDoS attacks can look like many non-malignant things that can cause accessibility issues like a brought down worker or framework, too many open solicitations from authentic clients, or even a cut link. It regularly requires traffic examination to figure out the thing is correctly happening.

2. Related Work

Models trained with the more popular Distributed Denial of Service tools, such as hping3, were examined, but they may not be able to identify DDoS attacks triggered by many Distributed Denial of Service tools [1]. Information on different forms of distributed DDoS can be found at. The network is examined, and packet data collected using the NetFlow protocol is shown [3]. As compared to some current approaches, this paper's hybrid approach for detecting Distributed Denial of Service attacks provides the highest detection accuracy [2]. In this proposed system, a new architecture uses network traffic data from correlation functionality to create a new framework. The correlation functions are determined by the variability of the entropy calculated between both the features [13]. The variance of entropy is used to construct the function representative. The threshold value is then determined by taking the median of each function. During the training stage, the controller is given relevant information to distinguish between request packets and usual requests.

During the testing process, the featured representative of the test sample is equal to the existing knowledge using the Euclidean distance. Categorise the test results as standard or attack conditions based on this contrast. The data from CAIDA 2007 was used to simulate the system. The findings show that the detection accuracy and time are far superior to other methods currently in use.[4]. They have developed a network security model for detecting DDoS attacks at the application layer. They create a website for collecting the data and hold a log of both attacking and non-attacking users. When a user accesses the server's logs, the values of the features are saved in a MySQL database and translated to a CSV format with Weka. They added two new features: DT (differences between two consecutive times of website requests from a specific IP address) and BTS (indicate dissimilarity and similarity in byte size). Using SMOTE, the data is resampled. The dataset is broken down into three sections: 70% planning, 15% evaluation, and 15% cross-validation to avoid overfitting.

The naive Bayes technique, which has a 99% accuracy rate in recognising DDoS attacks and legitimate requests, is used to categorise the instances [5]. They created a model that combines an evolutionary neural network with a PSO-based neural network (Particle Swarm Optimization). They used the PSO algorithm, which specifies the optimal connection weights for the Feed Forward Neural Networks, to boost the ANN's efficiency in detecting the intrusion. The PSO maintains a particle swarm in which each particle represents one of the swarm's potential solutions. Regular packet, User Datagram Protocol flood, and Transmission Control Protocol/Internet Protocol SYN attacks are included in the proposed hypervisor-based intrusion dataset for an experiment [6].

3. Proposed System

Here in the proposed system, we will be using Period based Defence Mechanism (PDM). In the PDM approach, we will employ a blocklist and PDM System based on cycles that check for the similar data packets present in the blocklist and help us prevent data flooding attacks by analysing each data packet at the end of each period during the flood to improve the incoming storm traffic. As a result, storm traffic Quality of Service will be assured. Consequently, for the length of the transfer, multiple data packets are delivered at a significant

level. Nodes that are malicious or greedy initiate flood attacks. Malicious nodes can be set up on purpose or overturn using cell phone worms to clog up the network and misuse the capabilities of other end devices. Selfish nodes can use flood attacks to increase the communication bandwidth. Because of Delay-tolerant networking (DTN) opportunistic networking, a single packet may only be sent to the destination with a chance of less than one when a greedy node floods a packet with multiple replications, the odds of the packet being sent increase.

Every replica's delivery means the packet will be delivered successfully. Packet flood attacks can also help selfish nodes maximise their throughput, albeit in a more subtle manner. A node in the proposed Single-copy routing erases its copy of the packet after sending it back [7]. As a consequence, there is only one copy of each packet in the network. The planned Multicopy routing sprays a certain number of replications to other nodes to a packet's source node. Using the single-copy technique, each copy is routed separately. The utmost number of copies per packet is calculated. When a node decides that sending a packet to another experienced node is necessary (as determined by the routing algorithm), it mimics the packet and keeps the duplicate. The number of replicas in a packet is not set in stone [12]. If another met node has more reliable communications with the packet's destination, a node duplicates the packet.

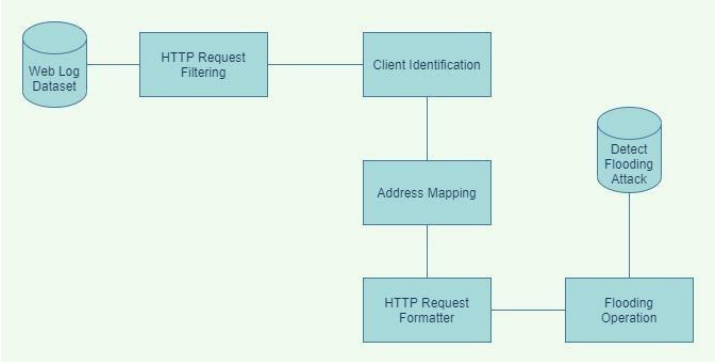

Figure 1: Architecture Diagram

3.1. Project Modules

3.1.1. Module 1: Exploratory Data Analysis

The first step in the data analysis process is exploratory data analysis (EDA). Here, we will find out how to make sense of the data we have and what questions we want to ask and how to frame them, and better manipulate the data sources to get the answers we need. We can do this by looking at patterns, trends, outliers, unusual outcomes, and other items in our current data. To get a sense of the story, this says, we used both visual and quantitative approaches. Exploratory Data Analysis is beneficial to data science ventures because it helps them get closer to knowing that the potential findings will be accurate, correctly interpreted, and relevant to their desired business contexts. Only after raw data has been verified and reviewed for irregularities, ensuring that the data collection was obtained without errors, can such a degree of certainty be achieved [11].

EDA also helps explore insights that may not have been apparent or worth exploring to market stakeholders and data scientists but are highly insightful about a specific business. EDA is used to describe and fine-tune the set of function variables used in machine learning [10]. Once data scientists have a good understanding of the data set, they may need to go back to the feature engineering stage because the initial features may not be fulfilling their intended function. When the EDA stage is over, data scientists have a solid feature set to work with it.

3.1.2. Module 2: Pre-Processing

Any data in the dataset may be missing. When we encounter a challenge, we must be prepared to deal with it. We might erase the entire data line, but what if we erase critical information without realising it? Of course, we will never do anything like that. Taking the average of all the values in the same column and filling in the gaps

is one of the most common solutions. Scikit Learn pre-processing is the name of the library we will use for this mission. There is a section called Imputer in there that can help us fill in the blanks. Our information is often qualitative, i.e., we use texts as information. We may identify categories in the text type. Since the simulations are based on mathematical equations and calculations, computers have a harder time interpreting and processing texts than numbers.

As a consequence, we must encode categorical data [8]. Our dataset must now be split into two parts: a training set and a test set. We will use our training collection to train our machine learning models. They will attempt to understand any correlations in the data. The models will then be tested on our test range to see how accurate they are predicting [9]. As a general rule, 80% of the dataset should be assigned to the testing group. In comparison, the remaining 20% should be assigned to the evaluation set. For this mission, we will use a test train split from the Scikit-learn model selection repository.

3.1.3. Module 3: Feature Engineering

Filter methods are commonly used in the pre-processing phase stage. Any machine learning algorithms have no bearing on feature selection. Instead, features are chosen based on their association with the outcome variable measured by various statistical tests. The word "correlation" is used here to refer to a subjective concept. We may use the table below to define correlation coefficients as a starting point. Pearson's Correlation: The Pearson's Correlation metric is used to determine the linear relationship between two continuous factors, X and Y. It has a value that varies from -1 to +1. LDA is a tool for evaluating a rectilinear set of features that characterises or categorised into two or even more categorical variable types. ANOVA (Analysis of Variance) is a statistical method for comparing two or more variables. ANOVA is short for Analysis of Variance. It functions in the same way as linear discriminant analysis (LDA), but with one or more categorical individualistic attributes and one continuous depending on function [9].

It performs a mathematical test to see if the means of several classes are equivalent. Chi-Square: This is a mathematical measure that is used to determine the likelihood of similarity or relationship between classes of categorical features depending on the frequency distribution of those features.

3.1.4. Module 4: Prediction

After the preparation, it is time to use Evaluation to see if the model is any good. This is where the previously set-aside dataset comes in handy. We may use evaluation to evaluate our model against data that has never been used before. This metric helps us to see how the model can do with data it has not seen before. This is supposed to represent how the model will work in the real world. We use an 80/20 or 70/30 split as a reasonable rule of thumb for a training-evaluation split. The size of the initial source dataset influences a lot of this. If we have a large amount of data, we will not need such a large fraction for the assessment dataset. After we have completed our test, we may want to see if there is some way we can enhance our training somehow. This can be accomplished by fine-tuning our parameters. When we did our preparation, we implicitly believed in a few parameters. Now is an excellent time to test those assumptions and try different values.

4. Implementation

First, we have visualised the data we have. We have removed unnecessary data from the dataset. By visualising, we have observed the pattern, trends, and information that could help us better. We have tested our dataset from basic algorithms to advanced algorithms like logistic Regression, Decision Tree, Random Forest, and KNeighbors and compared them with our approach. We have gained a good amount of ground. We used a confusion matrix and a Receiver Operating Characteristic (ROC) Curve with a True Positive Rate on the Y-axis and a False Positive Rate on the X-axis to visualise the output of each algorithm.

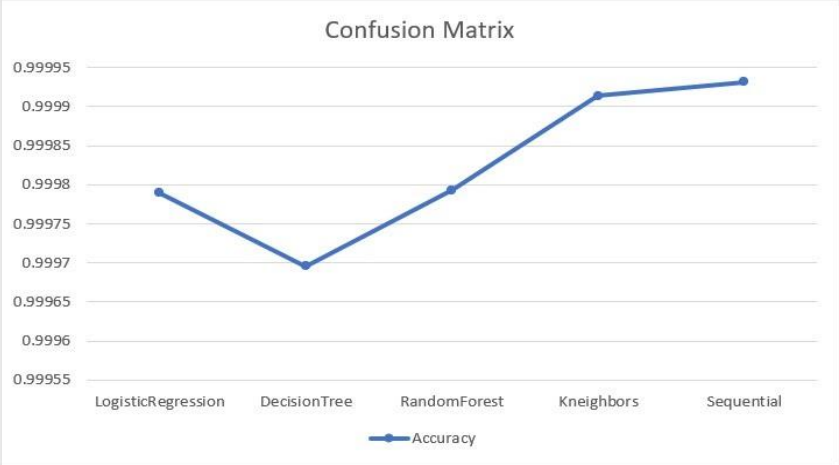

Figure 2: Accuracy

4.2. Result

When we tested the dataset with Sequential, we got an accuracy of 0.9999318756893532. Below we can see the Confusion matrix and Receiver Operating Characteristic (ROC) Curve.

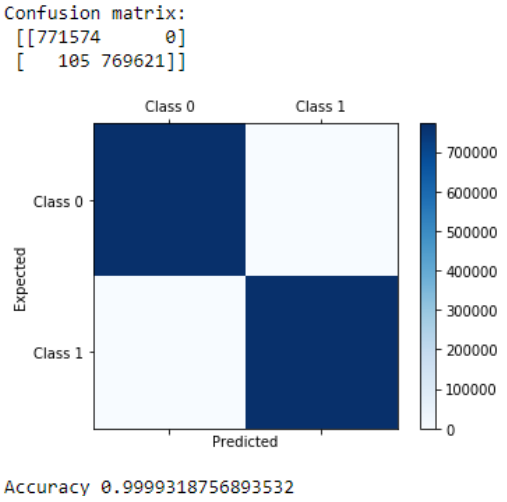

Figure 3: Confusion Matrix

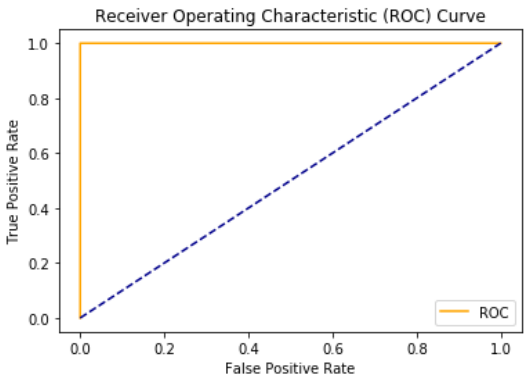

Figure 4: Receiver Operating Characteristic (ROC) Curve.

5. Conclusion

Reduce flood attacks by using rate limiting and optimising the claim-carry search to identify the number of attackers probabilistically. The cumulative number of packets that meet the rate limits is calculated using the learning automata algorithm. This research is being carried out on a decentralised basis. It can also handle a small group of attackers working together. They will easily lower it by relating the basic threshold to the throughput of outburst traffic. Consequently, during a data flooding attack, the prime focus is to maximise the throughput of outburst traffic. This is accomplished by employing the proposed method, which also ensures QoS compared to the previous design.

6. Future Work

In the future, our model may be extended to other attacks involving massive datasets, such as Application-Layer Attacks, Protocol Attacks, and Volumetric Attacks.

References

[1] S.Shanmuga Priya, M.Sivaram, D.Yuvaraj, A.Jayanthiladevi "Machine Learning-based DDOS Detection", 2020 International Conference on Emerging Smart Computing and Informatics (ESCI) *AISSMS Institute of Information Technology, Pune, India. Mar 12-14, 2020*

[2] Nandi, S., Phadikar, S., & Majumder, K. (2020). Detection of DDoS Attack and Classification Using a Hybrid Approach. 2020 *Third ISEA Conference on Security and Privacy (ISEA-ISAP).2020.*

[3] J. Hou, P. Fu, Z. Cao, and A. Xu, "Machine Learning-Based DDoS Detection Through NetFlow Analysis," MILCOM 2018 - 2018 *IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 1-6.*

[4] T. V. Sindia, and J. P. M. Dhas, "SBS-SDN based Solution for Preventing DDoS Attack in Cloud Computing Environment," *vol. 12, pp. 3593-3599, 2006*.

[5] V. Kumar and H. Sharma, "Detection and Analysis of DDoS Attack at Application Layer Using Naïve Bayes Classifier," Journal of Computer Engineering & Technology, *vol. 9, no. 3, pp. 208-217, 2018.*

[6] A. Rawashdeh, M. Alkasassbeh, and M. Al-Hawawreh, "An anomaly-based approach for DDoS attack detection in cloud environment," International Journal of Computer Applications in Technology, *vol. 57, no 4, pp. 312-324, 2018.*

[7] 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.*

[8] Afanasyev, & al., "Interest flooding attack and countermeasures in named data networking," in IFIP Networking Conference. *IEEE, 2013, pp. 1–9.*

[9] 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.*

[11] 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.*

[12] 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.*

[13] Yi, & al., "A case for stateful forwarding plane," Computer Communications, *vol. 36, no. 7, pp. 779–791, 2013.*

[14] Anurag Busha, Vakeesh Kanna, Sagar Naidu, Sathya R, ""Network Analysis Of Intrusion Detection Based On Machine Learning And Deep Learning", *International journal of engineering and advanced technology (IJEAT), ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019*