Enhancement of detection mechanisms for HTTP based DoS/DDoS attacks

M.Kiruthika* [1], Jitin John Charivukalayil[2], Shreya Chavan[3], Jerin John Mathew[4], Christopher Cardoza[5]

[1]M. Kiruthika *
Associate Professor, Department of Computer Engineering,
Agnel Charities' Fr.C.Rodrigues Institute of Technology, Vashi, Navi Mumbai, India
m.kiruthika@fcrit.ac.in

[2]Jitin John Charivukalayil
Department of Computer Engineering,
Agnel Charities' Fr.C.Rodrigues Institute of Technology, Vashi, Navi Mumbai, India
charivukalayil.jitin@comp.fcrit.ac.in

[3]Shreya Chavan
Department of Computer Engineering,
Agnel Charities' Fr.C.Rodrigues Institute of Technology, Vashi, Navi Mumbai, India
chavan.shreya@comp.fcrit.ac.in

[4]Jerin John Mathew
Department of Computer Engineering,
Agnel Charities' Fr.C.Rodrigues Institute of Technology, Vashi, Navi Mumbai, India
jerin.john@comp.fcrit.ac.in

[5]Christopher Cardoza
Department of Computer Engineering,
Agnel Charities' Fr.C.Rodrigues Institute of Technology, Vashi, Navi Mumbai, India
christopher.cardoza@comp.fcrit.ac.in

Abstract*: *DoS (Denial of Service) and DDoS (Distributed Denial of Service) attacks are some of the vicious network layer attacks present in the world. More than 5.4 million DDoS attacks were reported in the first half of 2021. Hyper Text Transfer Protocol (HTTP)-based Dos and DDoS attack, a type of DoS and DDoS attack, is a threat to web applications as it damages the application and the business. This paper sheds light on the current detection mechanisms of HTTP-based DoS and DoS attacks and the limitations identified in these detection mechanisms. This paper focuses on the mitigation strategies for HTTP-based DoS and DDoS attacks, which includes exploring various algorithms in machine learning to find the optimal algorithm to detect DoS and DDoS attacks. And a model for small businesses to include in their architecture to defend against DoS and DDoS attack. It also discusses an architecture which can be used by the visually impaired community to defend against the HTTP based DoS and DDoS attack. It is imperative to create efficient solutions to defend against such cyber-attacks and ensure proper network security in workplaces.

***Keywords*: Network Security, Denial of Service, Distributed Denial of Service, Mitigation, HTTP-based attacks**

(Article history: Received: 26th April 2022 and accepted 11th June 2023)

I. INTRODUCTION

With the advent of the technological era, online services have become an important aspect of our life. Most of these services include banking, shopping, entertainment, etc. that making life easier even in tough times of lockdown. Hence there is a demand for the working of these services to be smooth. But one of the major factors that disrupt the flow is cyber-attacks like DoS/DDoS attacks. A denial of service (DoS) attack is an attempt by an attacker to render a target inaccessible to its customers, resulting in customers being unable to use the service. DoS attacks are highly damaging attacks that cause the system to crash or degrade the quality of service in an unanticipated manner. A Distributed Denial of Service (DDoS) attack as discussed by G Saleh et al. [1], on the other hand, is an attempt to flood a victim by generating a significant amount of traffic from a big number of devices. These essential services usually work on an application level hence HTTP protocol is majorly used. Machine Learning models reviewed by Verma et al. [2] are also assisting in the training of models to detect and prevent attacks before they cause maximum damage. Ivandro et al. [3] claimed that as cyber-attacks become more complex and common, an IDS must be able to detect and respond to anomalies as rapidly as possible. To accomplish so, researchers have invented several intrusion detection algorithms that have been published in the literature.

HTTP facilitates communication using a web user and a web server. It is a connection-oriented protocol based on TCP, meaning communication can only be initiated once the TCP connection (3-way handshake) has been established. This connection must be maintained till the end of the

communication. Attackers largely target HTTP, as almost all organizations own a website to provide uninterrupted services to the customers and have a wide range of integration with online services and usually. HTTP traffic is not blocked by any security equipment by default. As per Cloud flare's 2021 report prepared by Vivek Ganti et al., HTTP DDoS on government/ public sectors has increased by 491% making it the second most targeted industry after Customer Services which has increased by 684% [4]. Hence, efficient detection mechanisms are needed so that the concerned administrators can prevent such DoS/DDoS attacks without much damage to the resources.

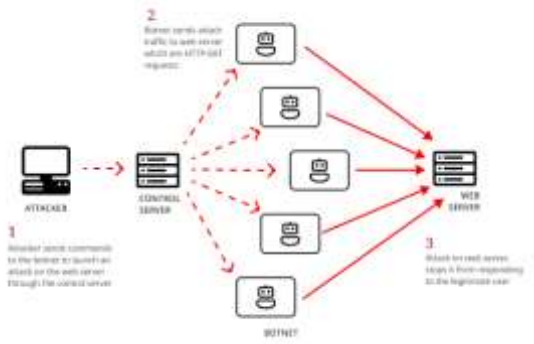

Fig. 1. HTTP Flood DDoS Attack on web architecture.

Fig. 1 shows a DDoS attack on a web server. The attack as shown in the figure is divided into 3 parts. First is the attacker, who controls and sends commands to the botnets to launch the attack on the server through the control server. Second is the botnet network, which waits for the command to come from the control server to attack the web server. Third is the web server, under the DDoS attack the web server does not respond to any incoming requests from any clients.

II. LITERATURE REVIEW

In this study, a few detection mechanisms have been chosen that use various methods to detect application layer attacks, particularly HTTP Dos/DDoS attacks. For this, a thorough literature survey has been performed to identify the gaps and drawbacks in the existing models after which improvements have been proposed for these models.

There are currently many detection models to detect HTTP DoS/DDoS attacks as mentioned before and many researchers have conducted a survey of all those methods over the years. Praseed et al. [5] discussed the critical aspects that aid in understanding how these assaults can be executed have been used to examine the complete range of application layer DDoS attacks. The defense mechanisms against various types of attacks are also explored, with a focus on traits that aid in the detection of different types of attacks. This discussion is expected to help researchers understand why a particular group of features are useful in detecting a particular class of attacks.

Oluwatobi et al. [6] have made a broader study on only slow HTTP attacks, their types, and the detection models that are present. The paper presents that more research is needed on slow HTTP attacks compared to volumetric attacks. Classification of DDoS flooding attacks and their detection methods were comprehensively carried out by Zargar et al. [7]. Khalaf et al. [8] believed that combining source address authentication, capability mechanisms and filtering methods along with stricter cyber law policies should be established in the systems to address the DDoS attacks. Statistical and Artificial Intelligence approaches to detecting and preventing DDoS attacks were discussed along with the advantages and disadvantages of each approach. The researchers have pointed out various gaps in achieving solutions in making a full-proof solution which includes a lack of updated datasets, the inclusion of new features to thwart new age attacks, etc.

When a particular user wants to acquire some services from the internet, the user initiates communication by sending requests from the web browser to the web server and in return, the web server sends either the result of the request made or an error page as per the situation. Hence, to derail this communication or process, the attacker tries to attack the server by sending many HTTP requests. Hence, methods safeguarding the web server were studied in which Muhammad et al. [9] have built a two-stage mechanism that protects the web server from slow HTTP attacks through firstly the NGINX reverse proxy which senses the sudden surge of attack flow, and secondly white-list based admission control policies which separate the attack flow from the normal flow.

Ndibwile et al. [10], introduces the use of three servers is made which are namely Real, Bait and Decoy server, and Snort NIPS, where the Real Server consists of the fully loaded website, the Bait server primarily listens to all the incoming requests, and as per the traffic, routes it to Real Server if the traffic is normal, else to the Decoy Server when there is attack flow. The major advantage of this method is it gives the attacker a false pretext that they are still attacking the real server.

Apart from protecting the server, understanding user behavior has been studied using various methods. Logistic Regression was used by Yadav et al. [11] to detect request flooding, session flooding, and asymmetric attack using 17 features. The model attained a detection rate of 98.64% and a False Positive Rate of 1.41%. Chengxu Ye et al. [12] discussed how a hierarchical clustering model is used to describe user browsing behaviors. 4 features namely average size of all objects in the session, request rate, average object popularity, and average transitional probability of objects in a session. EPA-HTTP dataset was used to train the user behavior and the model acquired an accuracy of 90-93%.

Luis et al. [13] have combined the usage of graphs, statistics, and analysis of HTTP requests to characterize user behavior in a multi-site web server. To detect a suspicious attacker, three analyses are conducted: statistics, HTTP

graph, and HTTP paths. This model proves to detect multiple categories of attacks but does not consider cases where the adversaries may change randomly among high workload state attacks. It could detect bots and security scanners as well as tools like Slowloris.Ranjan et al. [14] made a framework to classify various types of Layer 7 DDoS attacks and developed a counter-mechanism model which assigns a value to a suspicion session if there is a deviation from legitimate behavior. The detection model used statistical distribution and probability which reduces the computational complexity but due to this, it does not consider the request sequence of the packets sent which would be a limitation to the existing model. In [15], the researchers have used HTTP GET requests and their entropy to construct a time series model. Through their studies, they found out that HTTP Request per source IP address (HRPI) of DDoS is less than HRPI of normal traffic which means that HTTP GET requests are more converged during an attack hence a drop in HRPI value. Kalman smoothing filter and SVM classifier are used to further enhance the model.

A multilayer framework was proposed and designed by Saleh et al. [16], is evaluated based on optimal specifications of a protective framework and can fight all sorts of HTTP DoS/DDoS attacks. Emphasis on detection and mitigation of HTTP-based DoS attacks in the cloud architecture was carried out by Karnwal et al. [17]. The game theory approach was used by Mahsa et al. [18] to detect HTTP DoS to address the decision-making of attacker strategies. The model was tested on only small attacks and was focused mainly on two factors which were request rate and the workload which might create false positives during detection.

From the above review, limitations in the detection mechanisms have been identified and are tabulated in Table 1:

Name of Research Paper	Author/s	Limitation
Detection of application layer DDoS attack by modeling user behavior using logistic regression [11]	Yadav, S., & Selvakumar, S.	Limited datasets have been analyzed.
A Practical Approach and Mitigation Techniques on Application Layer DDoS Attack in Web Server [9]	Muhammad Yeasir Arafat, Muhammad Morshed Alam, Mohammad Fakrul Alam	Only Slowloris attack was considered.
Web Server Protection against Application Layer DDoS Attacks using Machine Learning and Traffic Authentication [10]	Ndibwile, Jema David, Govardhan, A., Okada, Kazuya, Kadobayashi, Youki	The current model does not favour people with disabilities. To address that, a speech recognition model is to be introduced

Table. 1. Limitations identified

To address the above limitations the following functionalities/objectives are proposed in the research study:

• To analyze various predictive models like Decision Tree, Random Forest Classifier, Support Vector Machine, Neural Networks for detection of HTTP based DoS/DDoS attacks with the help of datasets available online. Performance metrics of these models like accuracy, precision, and recall are compared with the existing models.
• To analyze the current infrastructure to find whether the model detects HTTP-based DoS attacks. The current infrastructure is modified and a R.U.D.Y attack is introduced, and results are collected using Wireshark.
• Introducing speech recognition module with the help of Web Speech API to the existing system to route traffic per packet characteristic i.e., Attack or Normal. If the user is legitimate, he/she is redirected to the real website; others mostly consisting of attack packets are redirected to the Decoy server. The model is also simulated in real time and resulting logs are collected.

III. METHODOLOGY

A. *Detection mechanism 1*

Data is collected from various sources and preprocessed. Necessary features are extracted and given to various models for detecting attack traffic. Some part of the data is later used to validate the model and the results like accuracy and other performance metrics are analyzed and compared for further study as depicted in fig. 2.

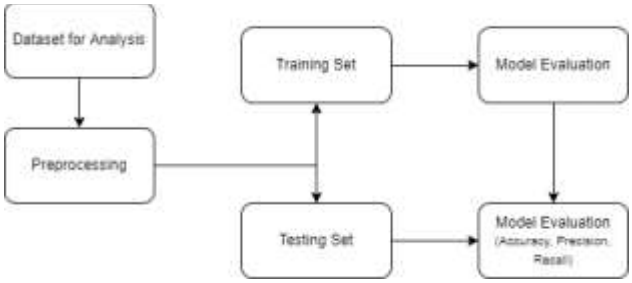

Fig. 2. Dataset generation and analysis.

B. *Detection mechanism 2*

1) Normal Scenario: In a normal scenario, the source would be a legitimate user and will try to visit the hosted web application, the stream of packets will be irregular due to the usual "thinking factor" by the user. This stream of packets reaches the NGINX and will be treated as normal traffic since there is no unusual change in the number of incoming packets. The packets will later reach the destination as usual.

2) Attack And Detection Scenario: In this scenario, now the attacker will be sending the packets which would be an incoming attack, and on the hosted web application, the stream of the incoming packets would be now constant and streamlined without any change in TTL values. Now, when

this traffic reaches the NGINX server, the decision engine will notice a surge in the incoming packets and send the traffic through an analyzer. The analyzer performs the challenge-response mechanism to check the legitimacy of the traffic and if it detects the DoS Attack then that source IP address of the traffic would be blacklisted in the iptables thus mitigating/reducing the effect of the attack. Both scenarios have been explained below in fig. 3. In the figures, the flow of data packets is shown in fig. 3, where two scenarios are considered. First being the normal scenario in which the flow of data when the web application is accessed by a normal user is shown. Second being the flow of data when an attacker attacks the web application through a DoS or DDoS attack is shown.

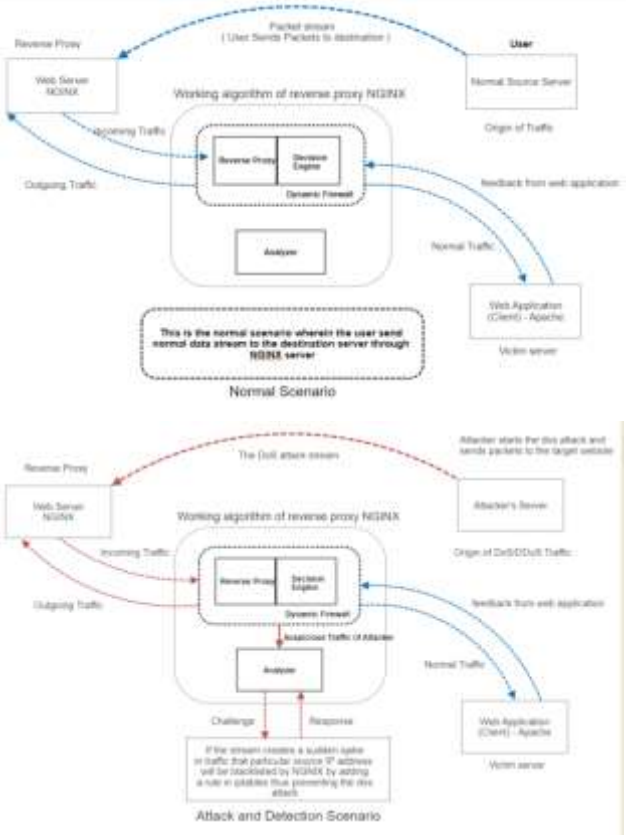

Fig. 3. Working of the Reverse Proxy NGINX in both normal and attack scenario.

C. Detection mechanism 3

1) Normal Scenario: Considering we have a normal user using a normal network with an IDS, the user might send too many requests to the server and due to this, the packets sent by them would be rejected thus signaling a false positive or at worst even getting blocked by the server itself. To solve this issue, a JavaScript authenticator is introduced in our proposed model which would test the legitimacy of a user by asking a simple question. The user would initially be authenticated by a speech recognition captcha before visiting the original or real website and once upon successful authentication the user can use the website just like any normal user.

2) Attack Scenario: In an attack scenario, the adversary would directly send huge streams of packets over the network or even mix the huge streams with normal traffic to evade the IDS, in both cases, since the attacker would not be authenticated through the bait server, the IDS would automatically reject such streams thus safeguarding the original web server from a DoS attack. The bait server can also be modified into a honey pot which can create a fake impression on the attacker that he has successfully initiated the attack but, the adversary is tricked into believing that they are on the original site.

The architecture used for this method is shown in fig. 4 with steps. The architecture consists of a UFW (Uncomplicated Firewall) firewall with Snort IDS implemented in the Bait server which uses Ubuntu operating system. Now during an attack, the incoming traffic will pass through the firewall and the Snort IDS will analyze the incoming packets and their frequency, when a higher surge of packets is detected, the packets will be diverted directly to the decoy server and since the attacker would not be able to solve the speech captcha, the user will be declined access to the real web server.

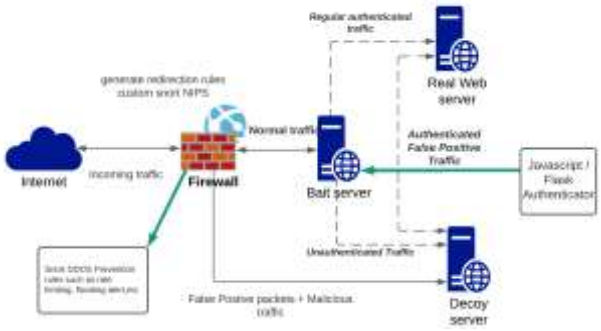

Fig. 4. The Mitigation Architecture

IV. Results and Discussion

A. *Detection Mechanism 1: Dataset analysis using different ML Algorithms*

In this detection mechanism, work is done with different data sets, and a comprehensive analysis is performed to understand the better approach for analysis of HTTP-based DoS/DDoS attacks. Google Collaboratory has been used for reading and analyzing data. The first dataset was a .arff file while the second dataset was a .csv file which was loaded with the help of the Pandas library and the classification models used are from the Sklearn library. The above analysis is performed using the following models with the mentioned datasets to understand the effective and suitable model.

- Decision tree algorithm
- Random forest algorithm
- SVM algorithm
- Neural Networks

```
df['PKT_CLASS'].value_counts()

Normal        1935959
UDP-Flood      201344
Smurf           12590
SIDDOS           6665
HTTP-FLOOD       4110
Name: PKT_CLASS, dtype: int64
```

Fig. 5. Class variable counts in dataset 1

The labels in dataset 1 [19] were Normal, UDP- Flood, Smurf, SIDDOS, and HTTP-FLOOD with the value counts as shown in fig. 5.

```
[4]   train_data['Label'].value_counts()

BENIGN          370623
DoS Hulk        310126
DoS slowloris   128612
Name: Label, dtype: int64

[7]   test_data['Label'].value_counts()

BENIGN          159295
DoS Hulk        132394
DoS slowloris    55180
Name: Label, dtype: int64
```

Fig. 6. Class variable counts in dataset 2

Dataset 2 [20] consisted of BENIGN which is the Normal traffic and DOS Hulk and Slowloris which are the attack traffic with their value counts of both train and test data as shown in fig. 6.

The dataset 1 was 75% training and 25% testing set. The first dataset consisted of 24 attributes with 2160668 rows while in the second dataset, the training set had 809361 rows and the testing data had 346869 rows. Table 2 and Table 3, depict the results of implementation of detection mechanism 1.

Dataset 1					
Model	Accuracy	Precision		Recall	
Decision Tree	97.3247	Normal 0.99	Attack 0.93	Normal 0.99	Attack 0.92
Random Forest	98.2742	0.99	0.95	1.00	0.95

Table 2: Result found for dataset 1

Dataset 2					
Model	Accuracy	Precision		Recall	
Decision Tree	99.9199	Normal 1.0	Attack 1.0	Normal 1.0	Attack 1.0
Random Forest	99.9963	1.0	1.0	1.0	1.0

Table 3: Result found for dataset 2

From the above tables we can see that Random Forest Classifier is the most accurate model in detecting normal and attack traffic followed by Decision Tree Classifier. The Support Vector Machine model and Neural Network model didn't provide any remarkable results and hence was not included in our results.

B. Detection Mechanism 2: Mitigation Mechanism using NGINX reverse proxy

For this method, an effort was made to create the mitigation mechanism and to try newer and effective attacks i.e., R.U.D.Y have been tried on the existing mechanism to prove its versatility as a DDoS prevention architecture.

For the first setup, OWASP Switchblade, which is an open-source DDoS tool known for simulation of Slow HTTP DDoS attacks was used which was then installed on a Windows instance which will be used for initiating the attack. Furthermore, we have two Ubuntu Servers, such that one server has no defense mechanism against DDoS and the other one has a NGINX reverse proxy which will aid towards defending the attack. According to fig. 7, Slow HTTP attack was performed on the bare Apache server without any defenses and the site was halted down due to the attack. The attack statistic is enclosed in the red box in the figure where most of the attack traffic affected the server.
Whereas, in fig. 8, the NGINX reverse proxy was able to easily mitigate the attack as the load was diverted from the real page hosted on port 8080 by the reverse proxy mechanism. Out of 400, 396 packets were disconnected hence the NGINX reverse proxy was able to easily mitigate the attack as the load has been diverted from the real page hosted on port 8080 by the reverse proxy mechanism. NGINX HTTPS reverse proxy is an intermediate proxy service that receives a client request, forwards it to one or more servers, and then sends the server's response back to the client. Most notable benefits of using NGINX as a reverse proxy are its load balancing and increased security mechanisms.

Fig. 7. Implementation of slow HTTP attack on bare Apache server

Fig. 8. Implementation of slow HTTP attack on Apache server with NGINX reverse proxy

Later, for the second setup, the R.U.D.Y Script was used as the attack vector, and Kali-Linux was used as the attack instance. In this scenario, the observations found out was that the script was successful in taking down the website hosted on Apache, meanwhile in the other machine where the reverse proxy has been implemented, the attack script failed to launch a successful attack after a few seconds of launch, thus showing that this method is also effective on other types of HTTP DDoS attacks. Both comparisons have been depicted in fig. 9 & fig. 10 using I/O graph for R.U.D.Y attack.

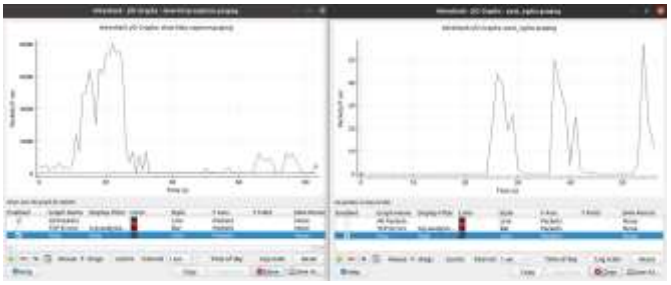

Fig. 9. I/O Graph before and after implementation of the model for Slow POST Attack

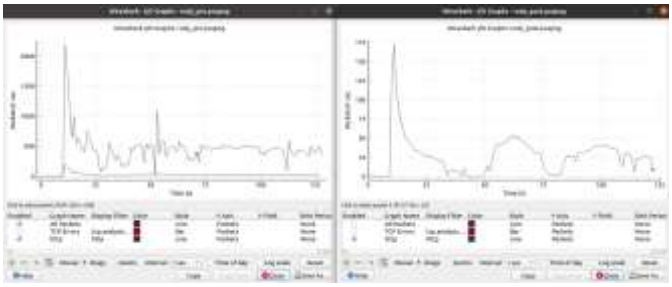

Fig. 10. I/O Graph before and after implementation of the model for R.U.D.Y attack

As far as results are considered, the proposed architecture triumphed in both methods of testing and was easily able to defend the incoming DDoS attacks from an adversary's PC. fig. 9, shows us the results of the packet capture analysis during the implementation of the first setup, wherein we can observe that without the architecture, the bare Apache server felt a maximum stress of around 8000 packets/sec (peak) whereas the NGINX reverse proxy scaled down such a big attack to a mere 50-60 packets/sec.

In fig. 10, the RUDY attack shows a similar trend wherein, the attack was successful on a normal Apache server providing a continuous stream of packets which hindered the site to load, whereas after the defense was enabled, a rise in packets are observed initially but later it becomes inconsistent as the attack proceeds, this mechanism wins over the RUDY attack as the NGINX server serves the hosted web page asynchronously, so incomplete requests are simply moved to the background while NGINX's event loop keeps working on other things. Hence, even though the attack was initiated, it was not able to successfully complete the task of taking down the server.

C. Detection Mechanism 3: DDoS Mitigation mechanism using Snort and speech recognition captcha

The set-up consists of three machines in the same network namely Bait, Real and Decoy in our Institute Laboratory. All incoming traffic comes through the Bait Server even though the attacker thinks that he/she is attacking the Real Server which consists of the website that has been targeted for the attack. The Bait Server consists of an audio CAPTCHA to prove the authenticity of the user to avail the services of the website. If it is a real user, they are redirected to the real website (Real server) else they are sent to the Decoy Server. The CAPTCHA also has a 30s timeout failing which will also lead to Decoy Server. The CAPTCHA used was speech recognition and the questions asked to the user are programmable as per requirement and in this model the days of the week were kept as the CAPTCHA questions.

The traffic is also authenticated using Snort IDS with the help of local rules and resultant logs are collected. Stress testing was done on this system by attacking it with help of HULK tool which is written in Python. All the machines used have 8GB RAM and Ubuntu 20.04 LTS operating system and the IDS used is Snort3. All the applications on the servers are made with the help of Flask.

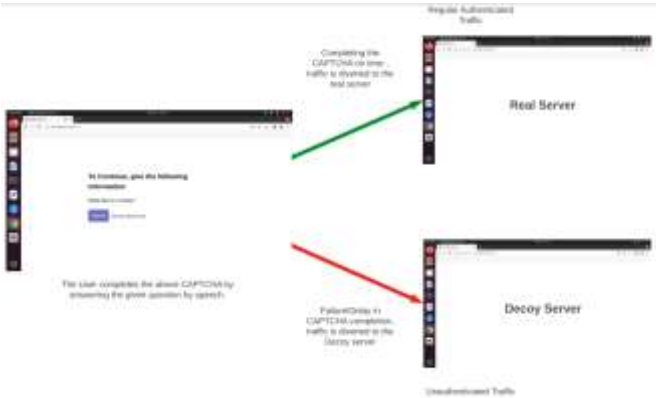

Fig. 11. The Speech CAPTCHA implementation

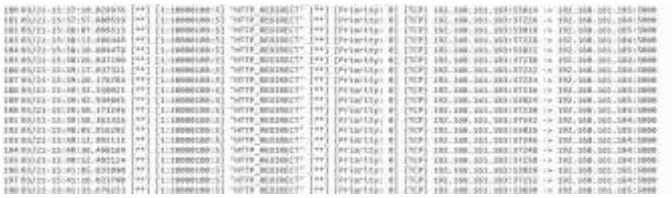

Fig. 12. Logging of HTTP traffic in normal scenario in Snort

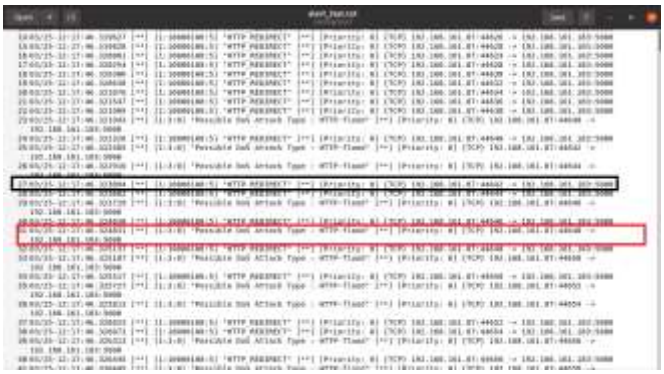

Fig. 13. Logging of HTTP traffic in attack scenario in Snort

Fig. 14. I/O Graph during HTTP Attack

As discussed earlier in the methodology section, the user attempts a CAPTCHA as shown in Fig.11, to gain access to the real web server, and in the case of an attacker, they fail to do so. The CAPTCHA acts as a gateway to access the real web server and the security measures implemented made sure that proper pivoting between servers is performed in both normal and attack scenarios. In Fig. 12, It was observed that the Snort rules are working, and logs are collected during normal scenario. An HTTP-based DoS attack was performed on the system using the HULK DoS tool and we could still use our systems even though it was getting attacked. The attack traffic is getting logged which is enclosed in red box while normal traffic is enclosed in black rectangle in Fig.13.

Further study was done on the system through Wireshark tool, which captures packets and analyzes them for more understanding. Through Fig.14, we can see that the system was still working under stress where the number of requests reached 1500-2000 per second. The logs provide the IP address of the traffic source which can be later blacklisted by the administrators using tools such as fail2ban of various SIEM systems.

V. Conclusion

With major services going online, the use of web and web services had increased. This proved to be a major area for attackers to wreak havoc and create disruptions for the people. DoS/DDoS attacks of various protocols were one of the cyber-attacks carried out frequently. Hence it is very important to detect such attacks and prevent consequences. This paper focuses on HTTP protocol and specifically the detection mechanisms. Various detection mechanisms and their limitations were reviewed. There were three limitations chosen and solutions to these limitations were discussed. The first limitation talks about the need for analyzing more datasets, for which we analyzed two different datasets with different algorithms. According to this analysis, decision tree and random forest algorithms are the favored algorithms for detecting HTTP-based DoS attacks with an accuracy of 97-99%. The second limitation talks about the architecture which can only detect slowloris attacks, for which the NGINX model was implemented on the network. The model is successful in reducing the impact of the attack on the web server and it sustains even other types of HTTP-based DoS attacks which were not possible in the previous literature. The third limitation consists of implementing a model for inclusivity of disabled people. The speech recognition model, a novel method that when implemented will help people with disabilities too, to access the website. Hence the proposed solutions have overcome the observed limitations. Our future work consists of including more methods and providing corrective measures.

References

[1] G Saleh, M. A., & Abdul Manaf, A. (2014). ”Optimal specifications for a protective framework against HTTP-based DoS and DDoS attacks”.2014 International Symposium on Biometrics and Security Technologies(ISBAST).

[2] Verma, Vishal and Kumar, Vasudha, DOS/DDOS Attack Detection using Machine Learning: A Review (April 24, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021.

[3] Ivandro Ortet Lopes, Deqing Zou, Francis A Ruambo, Saeed Akbar, Bin Yuan, "Towards Effective Detection of Recent DDoS Attacks: A Deep Learning Approach", Security and Communication Networks, vol. 2021, Article ID 5710028, 14 pages, 2021.

[4] Vivek Ganti and Omer Yoachimik (2021, July 20) “DDoS Attack Trends for 2021 Q2” [Blog post] Retrieved from https://blog.cloudflare.com/ddos-attack-trends-for-2021-q2/

[5] Praseed, A., & Thilagam, P. S. (2018). ”DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications”. IEEE Communications Surveys & Tutorials, 1–1.

[6] Oluwatobi Shadrach Akanji and Opeyemi Aderiike Abisoye and Sulaimon A. Bashir and Oluwaseun

Adeniyi Ojerinde (2020). ”A Survey on Slow DDoS Attack Detection Techniques”.

[7] Zargar, Saman Taghavi; Joshi, James; Tipper, David (2013).” A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks”. IEEE Communications Surveys & Tutorials, 15(4), 2046–2069

[8] Khalaf, B. A., Mostafa, S. A., Mustapha, A., Mohammed, M. A., & Abduallah, W. M.(2019). ”Comprehensive Review of Artificial Intelligence and Statistical Approaches in Distributed Denial of Service Attack and Defense Methods”.

[9] Muhammad Yeasir Arafat and Muhammad Morshed Alam and Mohammad Fakrul Alam (2015). ”A Practical Approach and Mitigation Techniques on Application Layer DDoS Attack in Web Server.”

[10] Ndibwile, J. D., Govardhan, A., Okada, K., & Kadobayashi, Y. (2015).”Web Server Protection against Application Layer DDoS Attacks Using Machine Learning and Traffic Authentication”.2015 IEEE 39th Annual Computer Software and Applications Conference.

[11] Yadav, S., & Selvakumar, S. (2015). ”Detection of application layer DDoS attack by modeling user behavior using logistic regression”. 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions).

[12] Chengxu Ye & Keshong Zheng & Chuyu She (2012). ”Application layer ddos detection using clustering analysis.”

[13] Luis Campo Giralte and Cristina Conde and Isaac Martin de Diego and Enrique Cabello(2013). ”Detecting denial of service by modeling web-server behaviour.”

[14] Ranjan, Supranamaya & Swaminathan, Ram & Uysal, Mustafa & Nucci,Antonio & Knightly, Edward(2009). ”DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks.”

[15] Tongguang Ni & Xiaoqing Gu & Hongyuan Wang & Yu Li (2013). ”Real-Time Detection of Application-Layer DDoS Attack Using Time Series Analysis”.

[16] Saleh, M. A., & Abdul Manaf, A. (2015). ”A Novel Protective Frame-work for Defeating HTTP-Based Denial of Service and Distributed Denial of Service Attacks”.

[17] Karnwal, T., Sivakumar, T., & Aghila, G. (2012). ”A Comber Approach to Protect Cloud Computing against XML DDoS and HTTP DDoS attack”.

[18] Mahsa Emami-Taba and M. Amoui and L. Tahvildari(2015). ”Strategy Aware Mitigation Using Markov Games for Dynamic Application-Layer Attacks.”

[19] Dataset used for Detection Mechanism-1. https://www.kaggle.com/jacobvs/ddos-attack-network-logs.

[20] Dataset used for Detection Mechanism-1.https://www.kaggle.com/datasets/wardac/applicationlayer-ddos-dataset

AUTHOR PROFILE

M. Kiruthika
Associate Professor,
Computer Engineering Department,
Agnel Charities'
Fr.C.Rodrigues Institute of Technology,
Vashi, Navi Mumbai, India
m.kiruthika@fcrit.ac.in

Jitin John Charivukalayil
B.E Computer Engineering Student,
Computer Engineering Department,
Agnel Charities'
Fr.C.Rodrigues Institute of Technology,
Vashi, Navi Mumbai, India
charivukalayil.jitin@comp.fcrit.ac.in

Shreya Chavan
B.E Computer Engineering Student,
Computer Engineering Department,
Agnel Charities'
Fr.C.Rodrigues Institute of Technology,
Vashi, Navi Mumbai, India
chavan.shreya@comp.fcrit.ac.in

Jerin John Mathew
B.E Computer Engineering Student,
Computer Engineering Department,
Agnel Charities'
Fr.C.Rodrigues Institute of Technology,
Vashi, Navi Mumbai, India
jerin.john@comp.fcrit.ac.in

Christopher Cardoza
B.E Computer Engineering Student,
Computer Engineering Department,
Agnel Charities'
Fr.C.Rodrigues Institute of Technology,
Vashi, Navi Mumbai, India
christopher.cardoza@comp.fcrit.ac.in