IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

DENIAL OF SERVICE SIMULATION AND DETECTION

MUDUNURI URMIKA,KANCHIMIREDDY LASYA,AASMI KOTHARI,SK.HASEENA
COMPUTER SCIENCE AND ENGINEERING KALASALINGAM ACADEMY OF REASEARCH AND EDUCATION

ABSTRACT:

The goal of a Denial of Service (DoS) assault is to prevent legitimate users from accessing a target system, network, or service by flooding it with excessive traffic or resource requests. By depleting resources like bandwidth, CPU, or memory, a denialof-service attack aims to disrupt regular access to services.
Numerous detrimental consequences can result from a Denial of Service (DoS) assault. Users may be unable to access websites or online services as a result of it crashing. Downtime and disgruntled consumers can cost businesses money. Attacks that occur frequently can damage a business's reputation and erode consumer confidence. (DoS) assaults are occasionally used by attackers as a diversion from more serious cyber crimes, such data theft. All things considered, (DoS) attacks have the potential to impair operations, cause monetary losses, and pose security threats. Network security is seriously threatened by denial of service (DoS) attacks, which cause service interruptions by flooding target systems with excessive requests. Two of the most common methods employed by attackers are the TCP/SYN flood and the Slow Loris attack. In order to successfully detect malicious traffic, we model these threats in this study and put in place a machine learning- base detection system. To distinguish between malicious and normal network traffic, we use three machine learning algorithms: Random Forest, k-Nearest Neighbors (KNN), and Logistic Regression. To guarantee realistic and varied attack patterns, the training and testing dateset is created using simulated attack situations, such as Slow Loris and TCP/SYN flood. To increase classification accuracy, features are extracted from network traffic metrics.

INTRODUCTION:

Cyber security threats, particularly denial-of-service (DoS) assaults, have increased dramatically as a result of the expansion of internet-based services. By overtaxing network resources, these attacks interfere with services. Traditional security methods have trouble detecting complex attacks. Machine learning provides an adaptable approach to intrusion detection by analysing historical attack patterns and efficiently classifying network data. This paper aims to provide a comprehensive analysis of how machine learning models can enhance network security by effectively detecting and mitigating denial-of-service attacks. The complexity of Cyber attacks has led to the necessity for sophisticated systems that can quickly distinguish between malicious and genuine traffic.

Simulating DoS assaults, extracting pertinent network properties, and assessing the effectiveness of several machine learning models using important metrics like accuracy, precision, recall, and F1-score are the main objectives of this study. Finding the best model for detecting DoS attacks in real time and helping to build stronger cybersecurity defences are the objectives.

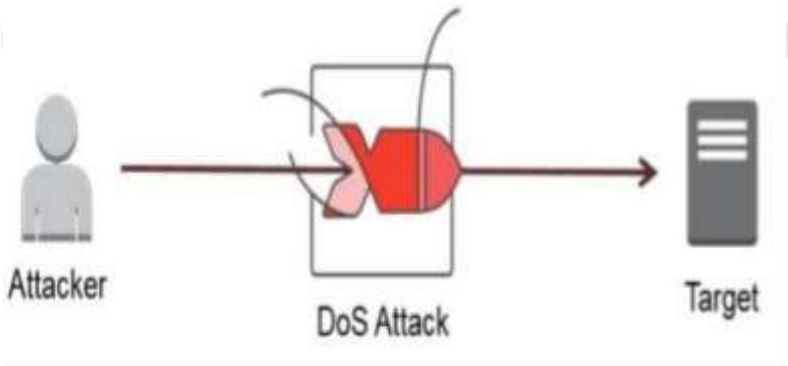

Fig1: denial of service

NEED OF THE STUDY.

Because of the growing dependence on internet-based services, network security has become a major concern for both consumers and enterprises. Denial of Service (DoS) attacks are among the most disruptive threats because they try to prevent authorized users from accessing a system or network resource. The TCP SYN Flood and Slow loris assaults are two of the most popular and successful types of denial-of-service attacks.

TCP SYN By issuing several SYN requests without establishing a connection, Flood takes advantage of the TCP three-way handshake mechanism and depletes server resources. By delivering partial HTTP requests, Slow loris, on the other hand, keeps a large number of connections to the target web server open and keeps them open for as long as possible, preventing the server from processing valid traffic

Considering the dynamic character of these attacks and the range of suggested detection methods, a thorough analysis is necessary to:

Recognize the Threat Landscape: Examining historical occurrences and patterns aids in determining the vulnerabilities that these assaults target and how they are changing.

Examine Simulation Techniques: To test detection techniques in realistic settings, it is essential to properly simulate DoS attacks. A survey offers information about different simulation platforms and tools (e.g., LOIC, Hping3, Metasploit, SlowHTTPTest).

Compare Detection Techniques: There are many different detection techniques, ranging from machine learning-based anomaly detection to signature-based intrusion detection systems (IDS). A survey aids in assessing their adaptability, scalability, and efficacy.

Determine Gaps and Challenges: Researchers can identify shortcomings in present solutions, such as high false positives, the incapacity to detect low-and-slow attacks, or the absence of real-time detection, by examining the body of existing literature.

Encourage Future Research: By identifying areas that require more investigation or development, a well-structured survey provides the groundwork for future research.

RESEARCH METHODOLOGY

a) DENIAL OF SERVICE SIMULATION

Depending on the kind of assault being evaluated, there are various methods for simulating a denial-ofservice attack. One popular technique is to simulate real-world DoS circumstances by flooding a target system with excessive network requests using tools like Hping3 or LOIC (Low Orbit Ion Cannon). Another strategy is TCP/SYN flood simulation, in which a tool or script continuously sends a server unfinished connection requests, using up all of the server's resources. A straightforward Python script can be used to launch several sluggish connections for Slow Loris attacks, which hold them open for a long period and stop new connections from being made.

We used two types of Dos Attacks mainly to simulate dos

1) TCP SYN FLOOD ATTACK
2) SLOW LORIS ATTACK

1) TCP SYN FLOOD ATTACK

SYN Flood on TCP is a form of Denial of Service (DoS) attack known as a TCP SYN flood uses a portion of the standard TCP three-way handshake to overload the targeted server's resources and cause it to become unavailable. Network saturation results from SYN flood DoS, in which the attacker sends TCP connection requests more quickly than the victim computer can handle them.

Using a fictitious IP address, the attacker repeatedly sends SYN packets to each port on the targeted server in a SYN flood attack. Several seemingly valid requests to establish connectivity are sent to the server. It sends a SYN-ACK message from each open port in response to every attempt.

If the IP address is fake, the attacker either never receives the SYN-ACK at all or fails to send the anticipated ACK. In any case, the compromised server will have to wait a while for its SYN-ACK packet to be acknowledged. The connection remains open during this period since the server is unable to terminate it by sending a RST packet. Another SYN packet will come before the connection can time out. This results in a growing number of connections remaining partially open.

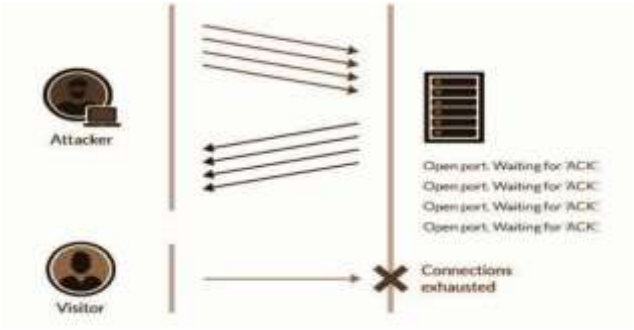

fig2: tcp syn flood

TCP SYN FLOOD EXECUTION

Follow these steps to simulate a **TCP/SYN flood attack** in a controlled environment using Metasploit:

1. Open Metasploit Run the following command to start Metasploit in quiet mode: ●**sudo msfconsole –q**
2. Search for TCP/SYN Flood Module Search for available TCP/SYN flood attack modules:●**search tcp/synflood**
3. Select the First Module from the List Use the first available module (if it's auxiliary/dos/tcp/synflood, for example):●**use 0**
4. Set the Target IP Address Replace <metaip> with the actual IP address of your target system: ●**set RHOSTS <metaip>**
5. Show Module Options Verify the required parameters: ●**show options**
6. Run the SYN Flood Attack Launch the attack: ●Run
7. Check the Website Availability Try to access the target machine's web service from a browser using: ●**http://metaip**

2) SLOW LORIS ATTACK:

Web servers are the subject of a Slow Loris attack, a kind of Denial of Service (DoS) assault that exhausts the server's capacity to process new requests by maintaining several connections open for a lengthy amount of time. Slow Loris is a stealthy, lowbandwidth attack that is hard to detect, in contrast to standard DoS attempts that overload a server with traffic. It operates by delivering the target server incomplete HTTP requests that are never completed, causing the server to maintain those connections open while it awaits the completion of the request.

This eventually prevents authorized users from using the service because most web servers can only handle a certain number of concurrent connections. To keep the connections from timing out, Slow Loris delivers tiny pieces of data continually. This persistent and progressive attack is quite effective against poorly setup servers since it can shut down websites without disrupting other network services. Web application firewalls that identify and terminate questionable connections, rate limitation, and connection timeouts are some of the defences against Slow Loris

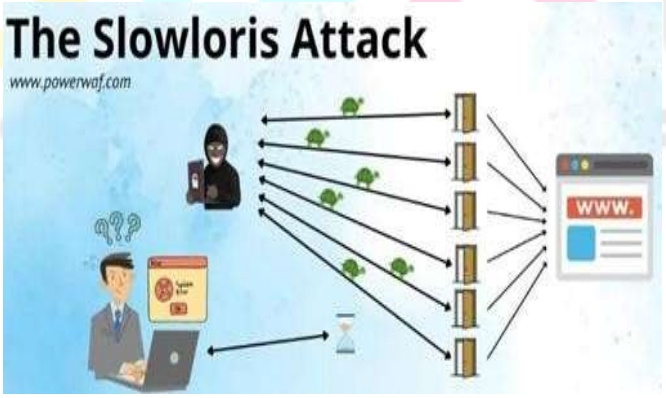

fig 3: slow lories attack

EXECUTION OF SLOW LORIES ATTACK

1) Execute the below command in the terminal to clone the slowloris.pl tool from GitHub.

- Command: git clone https://github.com/amittttt/slowloris.pl.git

2) Now navigate into the slow Loris directory and List the files in the directory.

- Cd SLOWLORIES/ ●ls

3) Add executable permissions to slowloris.pl file by executing below command

- COMMAND: CHMOD +X SLOWLORIS.PL

4) To perform DoS attack on the target system execute the below command in terminal

- ./SLOWLORIS.PL –DNS<TARGETIP> ● COMMAND: ./SLOWLORIS.PL --DNS <METASPLOITABLE IP>

5) Try to access the target machine's web service from a browser using:

- http://metaip

Approach:

3.1 : Collecting and Preparing Information

The NSL-KDD dataset was used to test and train our models. The collection includes labelled instances of network traffic that are categorized as either normal or attack types, including denial-of-service assaults. The data was pre-processed by properly naming the columns. Protocol type, service, and flag are examples of categorical variables that are encoded using one-hot encoding. Label mapping attack: DoS attacks were separated into discrete types to enhance classification. separating data into training and testing sets. Numerical characteristics are handled and normalized to improve model performance.

3.2 Features of Engineering

To improve model performance, feature selection and encoding techniques were applied. Numerical attributes such as duration, src_bytes, and dst_bytes were included. To ensure interoperability with machine learning methods, one-hot encoding was used for categorical characteristics. Principal Component Analysis (PCA) and Recursive feature selection approaches are used to identify the most relevant attributes impacting attack classification.

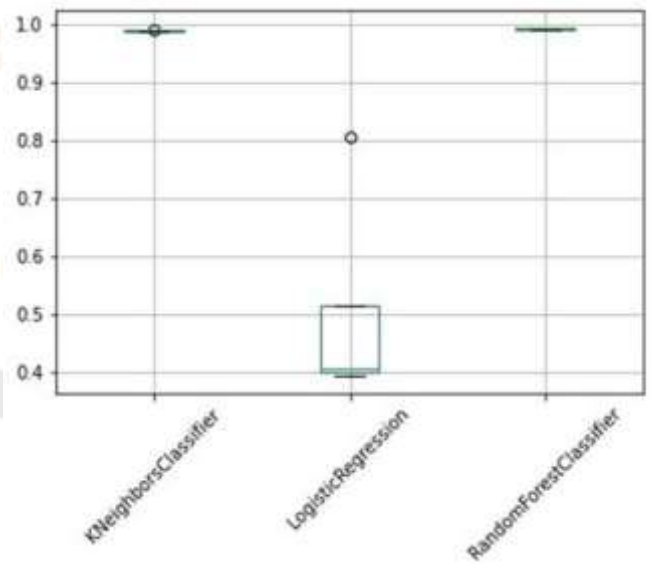

figure 5: random forest-based visualization of feature importance

3.3 Model Training and Selection

Three models for machine learning were put into practice:

The Random Forest Classifier is an ensemble learning technique that builds several decision trees and has a high accuracy and resilience to over fitting.

For non-linear classification applications, K-Nearest Neighbors (KNN) is a distance-based classifier that finds patterns by comparing similarities.

A statistical model for binary classification that is helpful in determining baseline performance is logistic regression.

The model's performance was compared using cross-validation. In order to maximize model performance, Grid Search CV was also used for hyper parameter tuning. To evaluate how well the models handled skewed attack distributions, they were trained on both balanced and imbalanced datasets.

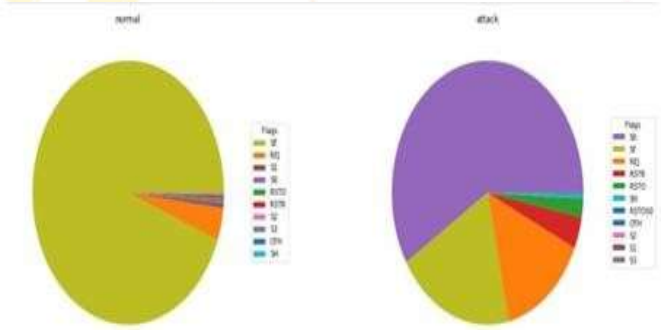

figure 4: workflow for model training and evaluation

4. FINDINGS AND CONVERSATION

4.1 MEASURES OF PERFORMANCE

Accuracy, precision, recall, and confusion matrices were used to assess the models. The findings showed that: The most dependable model is Random Forest, which has 99% accuracy and great recall and precision values. KNN: 98% Accuracy, high computing resource requirements, good performance. Although it performs worse than tree-based models, logistic regression is nevertheless helpful for comparing baselines.

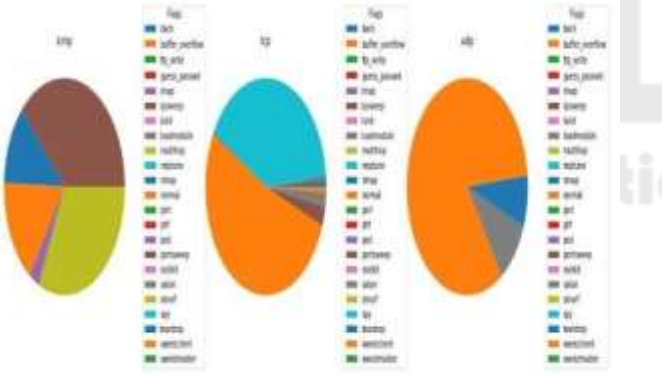

figure 5: comparison of various machine learning models' accuracy

4.2)Analysis of Confusion Matrix

The confusion matrix was shown using a heat map, which demonstrated how well the Random Forest model reduced false positives and false negatives. The model was quite successful in differentiating between attack and genuine traffic, which decreased the possibility of misclassification, according to the confusion matrix study.

V. RESULTS AND DISCUSSION

Dos attacks indiscriminately attacked numerous network services, emphasizing the need for strong security mechanisms.
While assaults propagated across various protocols, HTTP traffic dominated regular network activity, making feature engineering essential. Model performance was greatly enhanced by feature selection and encoding, which decreased training time and raised classification accuracy.
Scalability issues and computational complexity make real-time machine learning model implementation difficult.

Vi .OUTPUTS

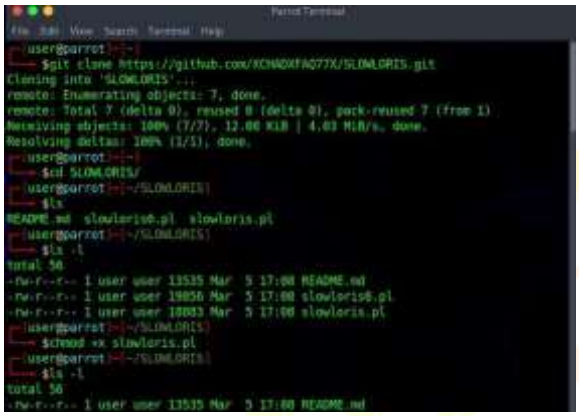

Fig 6 Slowlories(a)

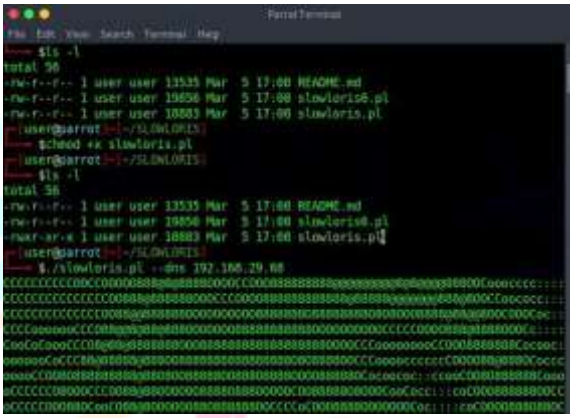

Fig7: Slow lories(b)

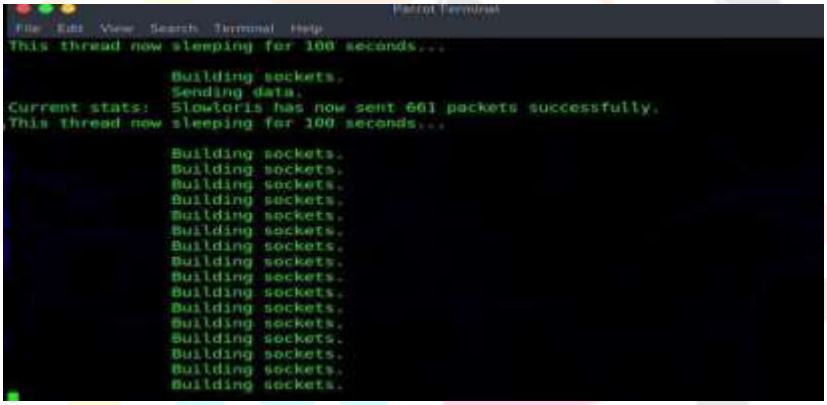

Fig 8: Slow lories output

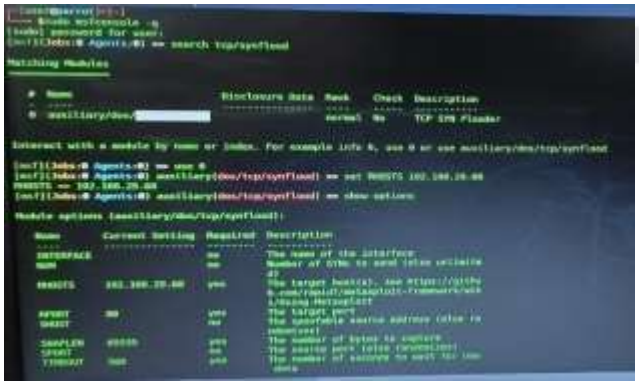

fig9:tcp syn flood (a)

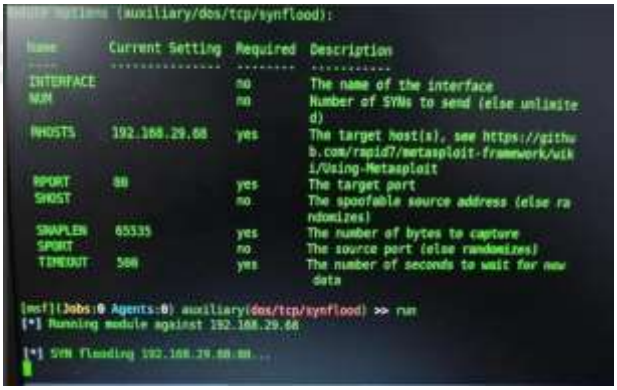

fig 10: tcp syn flood(b)

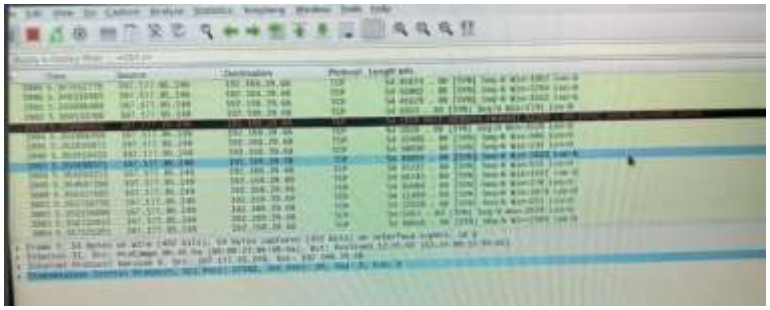

Fig11:tcp output

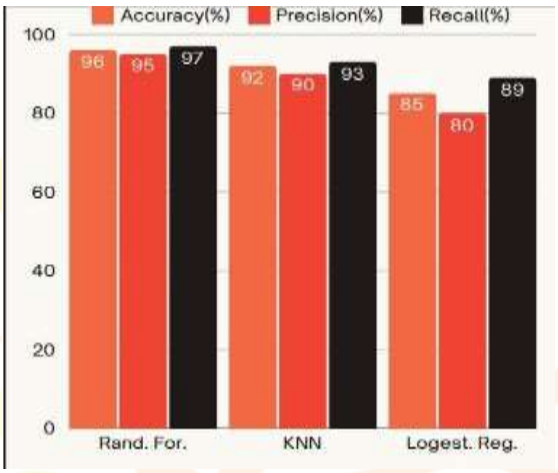

FIG12 : Dos detection using machine learning output

CONCLUSION:

Networks are at serious risk from Distributed Denialof-Service (DDoS) and Denial-of-Service (DoS) attacks, such as TCP SYN Flood and Slowloris. To identify these assaults, a variety of machine learning models have been employed, such as Random Forest, Logistic Regression, and K-Nearest Neighbors (KNN). According to studies, Random Forest models have shown above 99% accuracy in recognizing anomalous network traffic, whereas KNN can detect packet-level attacks with up to 92.8% accuracy. Despite being more straightforward, logistic regression has shown successful in differentiating between attack and legitimate traffic, however it might not be as accurate as more sophisticated models. While TCP SYN Flood attacks are more difficult and can occasionally lower detection accuracy to about 85–90%, machine learning classifiers have been able to obtain detection rates of about 95% for Slowloris attacks.

REFERENCES:

Research Papers & Articles

Doshi et al. (2018): Machine Learning for DDoS Attack Detection DOI: 10.1109/SPW.2018.00013
Tegeler et al. (2012): Flow-Based Intrusion Detection Using Machine Learning Link: https://ieeexplore.ieee.org/document/6249338
Wang et al. (2020): ML-Based Intrusion Detection for IoT Networks Link: https://doi.org/10.1109/JIOT.2020.3031190
Kumar et al. (2021): Deep Learning for DDoS Detection in Cloud Networks Link: https://doi.org/10.1109/ACCESS.2021.3066485
Jeyanthi & Vishnu (2022): Comparative Analysis of ML Models for Network Intrusion Detection Link: https://doi.org/10.1016/j.comnet.2022.109020

Datasets for DoS/DDoS Attack

CIC-DDoS2019 Dataset (Canadian Institute for Cybersecurity) Link: https://www.unb.ca/cic/datasets/ddos-2019.html

NSL-KDD Dataset (A Benchmark for Intrusion Detection Systems) Link: https://www.unb.ca/cic/datasets/nsl.ht ml