

DoS Attack



Project Report

**DENIAL OF SERVICE (DOS) USING MYSQL RELATIONAL DATABASE
STRUCTURE BASED ON NETWORK SECURITY**

Prepared by -
Mainak Roy

Guided by -
Zakir Hussain



TABLE OF CONTENTS

INDEX

1. DOS ATTACK DESCRIPTION

2. DATABASES USED IN THIS PROJECT

3. TABLES USED IN EACH DATABASES

4. QUERIES IDENTIFIED BY THE NETWORK INFRA
SECURITY TEAM

5. FINAL GOAL OF THIS PROJECT

6. CONCLUSION

DESCRIPTION

Denial of Service (DoS) attack is a cyberattack aimed at disrupting the normal functioning of a targeted server, service, or network by overwhelming it with a flood of illegitimate requests, making it unavailable to legitimate users. The primary objective of a DoS attack is to exhaust the target's resources, such as processing power, memory, or bandwidth, preventing normal operations.

Types of DoS Attacks

DoS attacks can be broadly classified into two types based on the method of attack:

- **Volume-Based Attacks**

ICMP Flood (Ping Flood):

The attacker sends a large number of ICMP Echo Request (ping) packets to the target, overloading the network's bandwidth.

UDP Flood:

The attacker sends a high volume of User Datagram Protocol (UDP) packets to random ports on the target. The target server tries to process these packets, but since they are invalid, it wastes resources trying to handle them.

DNS Amplification:

A reflection attack where small DNS queries are sent to an open DNS server with the victim's IP address, causing large responses to flood the victim's server.

- **Protocol-Based Attacks**

SYN Flood:

Exploits the TCP handshake process by sending a large number of TCP/SYN packets to the target without completing the handshake, consuming resources and leaving the server in a half-open state.

Ping of Death:

Involves sending malformed or oversized packets that the target server cannot process, causing it to crash or become unstable.

Smurf Attack:

The attacker sends an ICMP echo request (ping) to a network's broadcast address, using the target's IP address as the source, causing all devices in the network to respond to the target, overwhelming it.

- **Application Layer Attacks**

HTTP Flood:

Involves sending a large number of HTTP GET or POST requests to a web server, causing it to become overloaded and unable to process legitimate requests.

Slowloris:

Sends partial HTTP requests to the target server at a slow rate, keeping the connections open as long as possible. The server waits for the request to complete, which depletes its resources.

DNS Query Flood:

The attacker sends an overwhelming number of DNS requests to the target DNS server, overloading it and making it unresponsive to legitimate queries.

Key Features of DoS Attacks

Overload Resources: DoS attacks are designed to consume the target's resources, making it unavailable to legitimate users.

Interrupt Service Availability: By overwhelming the target, DoS attacks prevent normal operations, causing downtime.

Simple Execution: Many DoS attack types are relatively easy to launch with limited resources.

Targets Any Layer: DoS attacks can target any layer of the network stack, from the transport layer (TCP, UDP) to the application layer (HTTP, DNS).

Signs of a DoS Attack

Unusually slow network performance: Slow loading of websites or applications.

Unavailability of services: Websites, applications, or services become unreachable.

Excessive traffic: Anomalous traffic spikes from suspicious sources.

High CPU/memory usage: Servers exhibit high resource usage due to processing large amounts of incoming requests.

Prevention and Mitigation of DoS Attacks

Rate Limiting: Controls the number of requests a server can process per second to prevent overwhelming traffic.

Firewalls: Configuring firewalls to block unwanted traffic can mitigate some types of DoS attacks.

Traffic Filtering: Implementing traffic filtering rules to block malicious IP addresses or filter out attack patterns.

Content Delivery Networks (CDN): Using CDNs to distribute the load across multiple servers and mitigate high-traffic attacks.

DDoS Protection Services: Services like AWS Shield, Cloudflare, or Akamai that specifically defend against large-scale DDoS attacks.

DATABASES AND TABLES USED

Denial of Service (DoS) attack on a MySQL database typically involves overwhelming the MySQL server's resources (CPU, memory, or disk I/O) to prevent it from handling legitimate queries and connections. This can make the database inaccessible to users and services. Below are various methods that attackers may use to perform a DoS attack on MySQL databases, along with prevention and mitigation techniques.

The system is organized into five distinct databases, each with specific tables that store essential information for detecting, monitoring, and responding to DoS attacks.

DATABASE STRUCTURE :

Database 1: Attack_Detection

Database 2: Network_Traffic

Database 3: System_Resources

Database 4: Incident_Response

Database 5: Security_Information

Database 1: Attack_Detection

This database focuses on detecting and documenting the details of DoS attacks, including attack types, sources, detection rules, and generated alerts.

Tables:

1. attacks: Stores information about detected attacks, such as the type of attack, date, and source IP address.
2. attack_types: Contains a list of different types of attacks, including descriptions (e.g., DDoS, SQL Injection).
3. sources: Stores information about the origin of attacks, including source IP addresses and country details.
4. detection_rules: Stores the rules used to detect various types of attacks.
5. alerts: Tracks alerts generated by the system when attacks are detected, including the alert level and time of detection.


Database 2: Network_Traffic

This database monitors network traffic, protocols used, and information related to the devices involved in the traffic flow.

Tables:

1. traffic: Logs network traffic details, including timestamps, source and destination IP addresses, and protocols used.
2. protocols: Contains information about network communication protocols (e.g., TCP, UDP).
3. ip_addresses: Stores IP addresses, categorized by type (public, private).
4. network_devices: Provides details about network devices such as routers, firewalls, and servers.
5. traffic_stats: Records statistical data about network traffic volumes over time.

Database 3: System_Resources



This database monitors the usage of system resources, such as CPU, memory, and storage, which can be critical in identifying potential DoS attacks that exhaust system capacities.

Tables:

1. `resource_usage`: Tracks system resource usage over time, including CPU, memory, and disk usage.
2. `resources`: Contains descriptions of system resources and their functions.
3. `system_stats`: Logs overall system performance data such as uptime and load.
4. `process_list`: Stores details of running processes, including CPU usage and process IDs.
5. `user_sessions`: Monitors user session data, including session start and end times.

Database 4: Incident_Response

This database focuses on documenting incidents and managing the response plans and teams that handle security events.

Tables:

1. `incidents`: Logs details of security incidents, including the type and description of the incident.
2. `incident_types`: Contains different types of incidents (e.g., system breach, DDoS).
3. `response_plans`: Stores predefined response strategies for handling different types of incidents.
4. `response_teams`: Contains information about teams responsible for incident management, including team leaders.
5. `incident_reports`: Documents reports generated after an incident, including timelines and actions taken.

Database 5: Security_Information

This database is used for storing and managing critical security data such as vulnerabilities, patches, advisories, and intelligence on threats.

Tables:

1. vulnerabilities: Stores information about known vulnerabilities, their severity, and possible mitigation.
2. patches: Tracks patches applied to the system, including release dates and descriptions.
3. security_advisories: Contains official advisories about security threats and solutions.
4. threat_intelligence: Provides intelligence on known or emerging security threats, including threat levels.
5. security_incidents: Logs past security incidents, focusing on lessons learned and actions taken.


IDENTIFIED QUERIES

attacks table

1. `SELECT * FROM attacks; // Retrieve all attacks`
2. `SELECT * FROM attacks WHERE attack_type = 1; // Retrieve attacks by type (e.g., DDoS)`
3. `SELECT * FROM attacks WHERE attack_date BETWEEN '2022-01-01' AND '2022-01-31'; // Retrieve attacks by date range`
4. `SELECT * FROM attacks WHERE source_ip = '192.168.1.100'; // Retrieve attacks by source IP`

attack_types table

1. `SELECT * FROM attack_types; // Retrieve all attack types`
2. `SELECT * FROM attack_types WHERE type_name = 'DDoS'; // Retrieve attack type by name`



3. SELECT * FROM attack_types WHERE description LIKE '%flooding%'; // Retrieve attack types by description

sources table

1. SELECT * FROM sources; // Retrieve all sources
2. SELECT * FROM sources WHERE source_ip = '192.168.1.100'; // Retrieve source by IP
3. SELECT * FROM sources WHERE source_country = 'USA'; // Retrieve sources by country

detection_rules table

1. SELECT * FROM detection_rules; // Retrieve all detection rules
2. SELECT * FROM detection_rules WHERE rule_name = 'Rule 1'; // Retrieve detection rule by name
3. SELECT * FROM detection_rules WHERE rule_description LIKE '%DDoS%'; // Retrieve detection rules by description

alerts table

1. SELECT * FROM alerts; // Retrieve all alerts
2. SELECT * FROM alerts WHERE alert_level = 'High'; // Retrieve alerts by level
3. SELECT * FROM alerts WHERE alert_date BETWEEN '2022-01-01' AND '2022-01-31'; // Retrieve alerts by date range

FINAL GOAL OF THIS PROJECT

In this project, we have constructed a framework for managing and analyzing Denial of Service (DoS) attacks through a multifaceted approach involving several specialized databases. Each database plays a critical role in capturing and processing different aspects of network security, from attack detection and network traffic analysis to system resource monitoring and incident response. The ATTACK_DETECTION database provides a foundational repository for logging attack details, classifying attack types, and generating alerts based on detection rules. This structured data allows for a nuanced understanding of attack patterns and source information, which is essential for effective mitigation strategies. The NETWORK_TRAFFIC database complements this by offering detailed insights into network activity, including traffic volume and protocol usage. This information is crucial for identifying unusual traffic patterns that may indicate ongoing or potential attacks. The SYSTEM_RESOURCES is pivotal for monitoring system performance metrics such as CPU, memory, and disk usage. By analyzing this data during attack events, we can assess the impact on system resources and optimize our response to prevent system overloads and outages. The INCIDENT_RESPONSE database captures all aspects of incident management, including detailed records of incidents, response plans, and team activities. This facilitates a structured approach to addressing security incidents and ensures that response efforts are documented and refined. Finally, the SECURITY_INFORMATION database consolidates data on vulnerabilities, patches, and threat intelligence. This comprehensive repository supports proactive security measures by tracking known vulnerabilities and ensuring timely application of patches and updates. By integrating these databases, we create a robust system capable of addressing various dimensions of network security. The provided SQL queries allow for deep analysis and extraction of critical information, enhancing our ability to detect, respond to, and mitigate DoS attacks effectively.

Conclusion

This project demonstrates the effectiveness of a multi-database approach in managing Denial of Service (DoS) attacks. By utilizing the ATTACK_DETECTION, NETWORK_TRAFFIC, SYSTEM_RESOURCES, INCIDENT_RESPONSE, and SECURITY_INFORMATION databases, we create a comprehensive framework for monitoring and responding to security threats.

- ATTACK_DETECTION captures attack details and alerts.
- NETWORK_TRAFFIC monitors traffic patterns.
- SYSTEM_RESOURCES tracks system performance.
- INCIDENT_RESPONSE manages incident records and response plans.
- SECURITY_INFORMATION provides data on vulnerabilities and patches. Together, these databases allow for detailed analysis and effective mitigation of DoS attacks, improving overall network security. This integrated approach ensures a well-rounded defense strategy and enhances our capability to address and prevent security threats.