REVATURE

Project

DENIAL OF SERVICE ATTACK USING MYSQL RELATIONAL DATABASE STRCTURE BASED ON NETWORK SECURITY

Submitted By
More Tushar

Guided By
Zakir Hussain

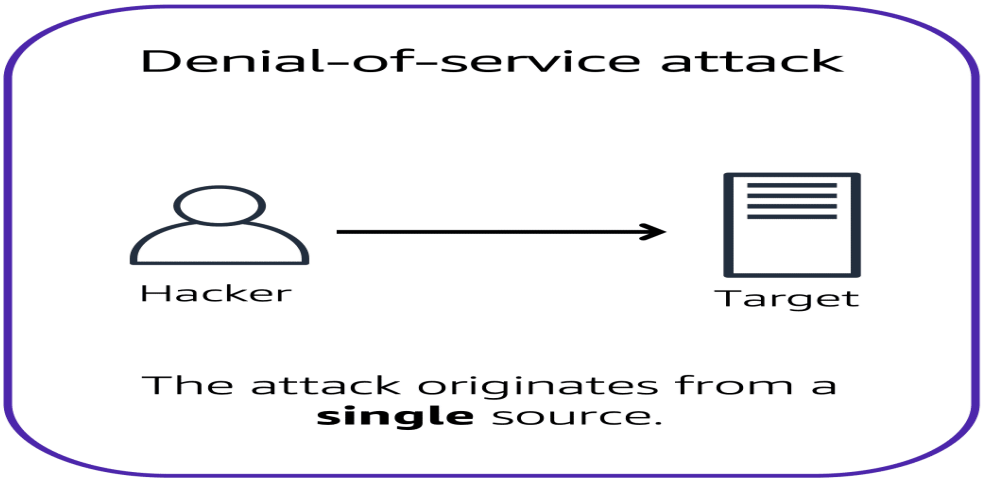
TABLE OF CONTENTS

Denial-of-service attacks

A **denial-of-service (DoS) attack** is a deliberate attempt to make a website or application unavailable to users.

Denial of service (DOS) is a network security attack, in which, the hacker makes the system or data unavailable to someone who needs it. Hacker tries to make a network, system, or machine unavailable by flooding it with fake requests or traffic. This prevents real users from accessing it, causing anything from slowdowns to complete shutdowns.

A Denial of Service (DoS) attack is a type of cyberattack where an attacker attempts to make a computer or network resource unavailable by overwhelming it with traffic or requests. The goal of a DoS attack is to exhaust the resource's capacity, making it unable to handle legitimate requests.



For example : An attacker might flood a website or application with excessive network traffic until the targeted website or application becomes overloaded and is no longer able to respond. If the website or application becomes unavailable, this denies service to users who are trying to make legitimate requests.

Types of DoS attacks:

1. **Volume-Based Attacks:** Volume-based attacks flood a network with too much data, overpowering its bandwidth and making the network unusable. Examples include **UDP floods** and **ICMP floods**. In a UDP flood, attackers send many UDP packets to random ports on a server, making the server busy trying to handle all these requests, which slows down or stops legitimate traffic.

2. **Protocol Attacks:** Protocol attacks exploit weaknesses in network protocols to use up server resources. Examples are **SYN floods** and the **Ping of Death**. In a SYN flood, attackers send many SYN requests to a server but don't complete the handshake, leaving the server stuck with half-open connections. The Ping of Death involves sending oversized packets to crash or disrupt the target server.

3. **Application Layer Attacks:** Application layer attacks target specific applications or services, causing them to crash or become very slow. Examples include **HTTP floods** and **Slowloris**. In an HTTP flood, attackers send many HTTP requests to a web server, consuming its resources. Slowloris keeps many connections to the server open by sending incomplete HTTP requests, preventing the server from handling new, legitimate requests.

Common DoS attack techniques:

1. **Flooding**: Flooding is a common DoS attack technique where a perpetrator sends a large number of requests or traffic to overwhelm the targeted resource, such as a website or server, making it unavailable to legitimate users.

2. **Buffer overflow**: Buffer overflow is another common DoS attack technique where an attacker sends more data than a buffer can handle, causing it to crash or become overwhelmed. This can lead to system instability or unexpected behavior, making the resource inaccessible to legitimate users.

3. **Malformed packets**: Malformed packets are a type of DoS attack technique where an attacker sends packets with incorrect or malicious data to cause errors in the targeted system. This can disrupt the communication between network devices or servers, leading to downtime or service interruption for legitimate users.

4. **SYN flooding**: SYN flooding is a type of DoS attack technique where an attacker sends a large number of SYN requests to the target, such as a server, in order to fill up the cache and prevent it from accepting new connections. This type of attack can lead to service unavailability for legitimate users.

DoS attacks can be launched using various tools and techniques, including:

1. **Botnets**: Botnets are networks of compromised devices that are controlled by a single entity to carry out coordinated attacks. These devices can include computers, servers, and IoT devices that have been infected with malware. Botnets are often used in DoS attacks to overwhelm a target with a large volume of traffic, causing it to become inaccessible to legitimate users.

2. **Malware:** Malware is malicious software that is designed to harm or exploit systems. It can be used to infect devices within a network and turn them into part of a botnet, or to launch other forms of cyber attacks.

3. **Scripting:** Scripting involves using scripts or automated tools to carry out attacks. This can include running scripts that flood a network with traffic or automate the process of sending malicious packets to a target.

To protect against DoS attacks, organizations can use:

1**. Firewalls:** Firewalls are a common defense mechanism used to block malicious traffic from reaching a network or resource. They can be configured to filter out unwanted traffic based on predefined rules or criteria, helping to prevent DoS attacks from overwhelming the system.

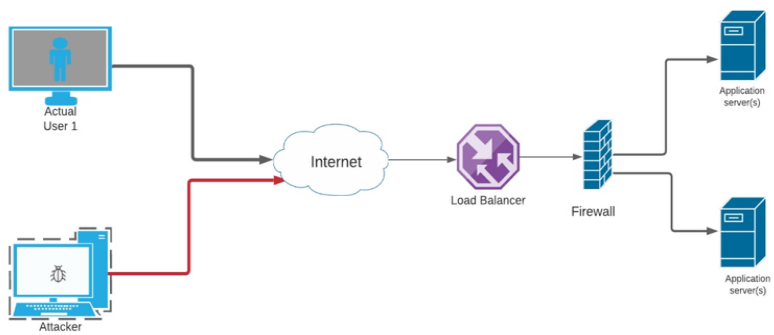
2. **Intrusion Detection/Prevention Systems (IDS/IPS):** Intrusion Detection/Prevention Systems (IDS/IPS) are security tools that monitor network traffic for suspicious activity and take action to prevent potential attacks. They can help detect and prevent DoS attacks by analyzing traffic patterns and identifying abnormal behavior that may indicate an ongoing attack.

3. **Load balancing:** Load balancing is a technique used to distribute incoming network traffic across multiple servers or resources. By spreading the workload evenly, load balancing can help prevent any single resource from becoming overwhelmed during a DoS attack, ensuring that traffic is handled efficiently and effectively.

4. **Content Delivery Networks (CDNs)** Content Delivery Networks (CDNs) are distributed servers that cache content and deliver it to users based on their geographical location.

5. **DDoS mitigation services:** DDoS mitigation services are specialized services designed to detect and mitigate DoS attacks in real time.

To protect against DoS attacks



SQL stands for Structured Query Language, and it is a standard programming language used to manage and manipulate relational databases. SQL allows users to query, insert, update, and delete data from databases, as well as create and manage database schemas, tables, and indexes. It is widely used in database management systems such as MySQL, PostgreSQL, Oracle, and Microsoft SQL Server for storing, retrieving, and managing data efficiently.
Collapse

MySQL is an open-source relational database management system that is based on SQL. It allows users to create, manage, and manipulate databases, tables, and data efficiently. MySQL is commonly used in web applications and is known for its high performance, scalability, and reliability.

Database :

A database is a structured collection of data stored in tables and managed through a database management system like MySQL.

DATBASE STRCTURE
Database 1: Attack_Detection
Database 2: Network_Traffic
Database 3: System_Resources
Database 4: Incident_Response
Database 5: Security_Information

The command to create a database in SQL :

CREATE DATABASE <database_name> ;

This command is used to create a new database with the specified name

Database 1: Attack_Detection

```
mysql> CREATE DATABASE Attack_Detection ;
Query OK, 1 row affected (0.03 sec)
```

Database 2: Network_Traffic

```
mysql> CREATE DATABASE Network_Traffic ;
Query OK, 1 row affected (0.01 sec)
```

Database 3: System_Resources

```
mysql> CREATE DATABASE System_Resources ;
Query OK, 1 row affected (0.02 sec)
```

Database 4: Incident_Response

```
mysql> CREATE DATABASE Incident_Response ;
Query OK, 1 row affected (0.02 sec)
```

Database 5: Security_Information

```
mysql> CREATE DATABASE Security_Information ;
Query OK, 1 row affected (0.01 sec)
```

The command to show a database in SQL :

SHOW DATABASES ;

This command is used to display a list of all databases .

The command to USE a database in SQL :

USE <database_name> ;

```
mysql> USE  Attack_Detection ;
Database changed
```

TABLE :

Table are used to store data in a structured format. Each table consists of rows and columns, with each row representing a record or entry in the database, and each column representing a specific attribute or piece of information related to that record.

Database 1: Attack_Detection

Attack detection databases are used to identify and report potential security threats or attacks on a network or system.

Tables:

1) attacks
2) attack_types
3) sources
4) detection_rules
5) alerts

The command to create a TABLE in SQL :

CREATE TABLE <table_name> (<column_name> <data_type>);

This command is used to create a new table with the specified name and define the columns with their respective data types .

1 Create attacks table

```
mysql> create table attacks(id int(5),attack_type int(5),attack_date DATETIME, source_ip varchar(50));
Query OK, 0 rows affected, 2 warnings (0.03 sec)
```

2 Create attack_types table

```
mysql> create table attacks_types(id int(5),type_name varchar(50), description varchar(200));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

3 Create sources table

```
mysql> create table sources(id int(5), source_ip varchar(50), source_country varchar(50));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

4 Create detection_rules table

```
mysql> create table detection_rules(id int(5), rule_name varchar(50), rule_description varchar(100));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

5 Create alerts table

```
mysql> create table alerts(id int(5),attack_id int(5),alert_date datetime,alert_level varchar(50));
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

Database 2: Network_Traffic :

The Network_Traffic database is a collection of tables that store information related to network traffic, such as data transmitted, source and destination IP addresses, protocols used, and timestamps. This database is crucial for analyzing network usage, identifying issues, and improving network performance.

Tables:

1 traffic
2 protocols
3 ip_addresses
4 network_devices
5 traffic_stats

1 Create traffic table

```
mysql> create table traffic(id int(5),timestamp datetime, source_ip varchar(
50), destination_ip varchar(50),protocol varchar(50));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

2 protocols

```
mysql> create table protocols(id int(5),protocol_name varchar(50),protocol_description varchar(100));
y OK, 0 rows affected, 1 warning (0.01 sec)
```

3 ip_addresses

```
mysql> create table ip_addresses(id int(5),ip_address varchar(50),ip_type varchar(50));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

4 network_devices

```
mysql> create table network_devices(id int(5), device_name varchar(50), device_type varchar(50));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

5 traffic_stats

```
mysql> create table traffic_stats(id int(5),timestamp datetime,traffic_volum
e int(50));
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

Database 3: System_Resources

System resources refer to the hardware and software components of a computer system that are used to perform tasks and operations. This includes but is not limited to, CPU (Central Processing Unit), memory (RAM), storage (hard drive or SSD), network resources, and peripherals such as printers or monitors. Monitoring and managing system resources is important to ensure that the system operates efficiently and effectively.

Tables:

1 resource_usage

```
mysql> create table resource_usage(id int(5), timestamp datetime, cpu_usage decimal(5,2),memory_usage decimal(5,2),disk_usage decimal(5,2));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

2 resources

```
mysql> create table resources(id int(5),resource_name varchar(50), resource_description varchar(100));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

3 system_stats

```
mysql> CREATE TABLE system_stats (id INT(5),  timestamp DATETIME,system_load DECIMAL(5, 2),system_uptime INT(5));
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

4 process_list

```
mysql> CREATE TABLE process_list (id INT(5),process_name VARCHAR(50),process_pid INT(10), process_cpu_usage DECIMAL(5, 2));
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

5 user_sessions

```
mysql> CREATE TABLE user_sessions ( id INT(5), user_id INT(5),session_start DATETIME,session_end DATETIME);
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

Database 4: Incident_Response :

Incident response refers to the process of reacting to and managing security incidents within an organization. It involves detecting, responding to, and resolving incidents to minimize damage and prevent future incidents.

Tables:

1 incidents

```
mysql> CREATE TABLE incidents ( id INT(5), incident_date DATE, incident_type VARCHAR(50), incident_description VARCHAR(255));
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

2 incident_types

```
mysql> CREATE TABLE incident_types (  id INT(5),type_name VARCHAR(50),  type_description VARCHAR(255));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

3 response_plans

```
mysql> CREATE TABLE response_plans (id INT(5), plan_name VARCHAR(50), plan_description VARCHAR(255));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

4 response_teams

```
mysql> CREATE TABLE response_teams (  id INT(5), team_name VARCHAR(50), team_lead VARCHAR(50));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

5 incident_reports

```
mysql> CREATE TABLE incident_reports (id INT(5),incident_id INT(5),report_date DATE,  report_description VARCHAR(255));
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

Database 5: Security_Information

The Security_Information database is a collection of tables that store information related to security incidents, threats, vulnerabilities, and security measures within an organization. This database is crucial for monitoring and managing security risks, analyzing security events, and implementing security protocols to protect sensitive information and prevent unauthorized access.

Tables:

1 vulnerabilities

```
mysql> CREATE TABLE vulnerabilities (id INT(5),vuln_name VARCHAR(50), vuln_description VARCHAR(255), vuln_severity VARCHAR(20));
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

2 patches

```
mysql> CREATE TABLE patches (id INT(5), patch_name VARCHAR(50), patch_description VARCHAR(255), patch_release_date DATE);
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

3 security_advisories

```
mysql>  CREATE TABLE security_advisories ( id INT(5), advisory_name VARCHAR(50),advisory_description VARCHAR(255));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

4 threat_intelligence

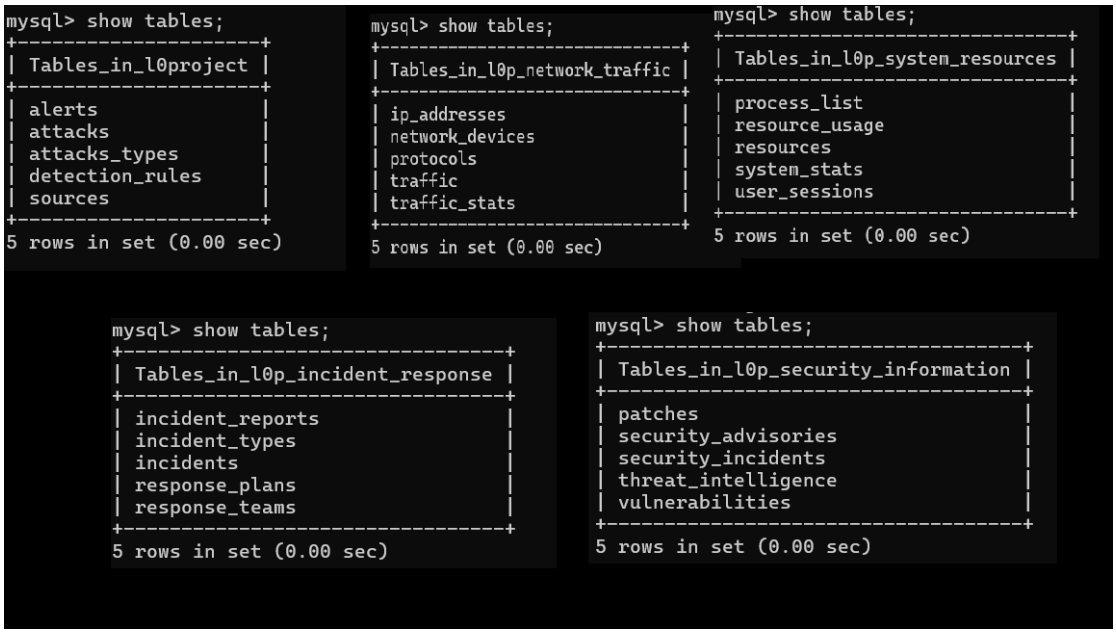
```
mysql> CREATE TABLE threat_intelligence (  id INT(5),  threat_name VARCHAR(50),threat_description VARCHAR(200),  threat_level VARCHAR(20));
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

5 security_incidents

```
mysql> CREATE TABLE security_incidents (id INT(5),incident_id INT(5),security_incident_date DATETIME);
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

The command to show tables in SQL :

SHOW TABLES ;



The command to insert data into a table in SQL is the INSERT INTO statement :

INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);

This command is used to add new records or data entries to a table in a SQL database.

Database 1: Attack_Detection

1 Attacks Table : Insert data on detected attacks, including type, date, and source IP.

The Attacks table typically includes columns such as id (a unique identifier for each attack entry), attack_type (the type of attack that occurred), attack_date (the date and time when the attack occurred), and source_ip (the IP address of the attacker). This information helps security teams track and analyze different types of attacks on a network or system.

Inserting data into attacks table

```
mysql> insert into attacks values (1, 1, "2022-01-01 12:00:00", "192.168.1.100"),(2, 2, "2022-01-02 13:00:00", "192.168.1.101"),(3, 3, "2022-01-03 14:00:00"
, "192.168.1.102"),(4, 1, "2022-01-04 15:00:00", "192.168.1.103"),(5, 2, "2022-01-05 16:00:00", "192.168.1.104");
Query OK, 5 rows affected (0.01 sec)
```

The command to DISPLAY ALL DATA of a table in SQL :

SELECT * FROM table_name;

```
mysql> select * from attacks;
+------+-------------+---------------------+---------------+
| id   | attack_type | attack_date         | source_ip     |
+------+-------------+---------------------+---------------+
|    1 |           1 | 2022-01-01 12:00:00 | 192.168.1.100 |
|    2 |           2 | 2022-01-02 13:00:00 | 192.168.1.101 |
|    3 |           3 | 2022-01-03 14:00:00 | 192.168.1.102 |
|    4 |           1 | 2022-01-04 15:00:00 | 192.168.1.103 |
|    5 |           2 | 2022-01-05 16:00:00 | 192.168.1.104 |
+------+-------------+---------------------+---------------+
5 rows in set (0.00 sec)
```

2 attack_types table : Inserting data into attack_types table

```
mysql> select * from attacks_types;
+------+---------------------+-----------------------------------+
| id   | type_name           | description                       |
+------+---------------------+-----------------------------------+
|    1 | DDoS                | Distributed Denial of Service     |
|    2 | SQL Injection       | Structured Query Language Injection |
|    3 | Cross-Site Scripting | XSS                              |
|    4 | Brute Force         | Password Guessing                 |
|    5 | Phishing            | Social Engineering                |
+------+---------------------+-----------------------------------+
5 rows in set (0.00 sec)
```

3 sources table : Inserting data into sources table

```
mysql> select * from sources;
+------+---------------+----------------+
| id   | source_ip     | source_country |
+------+---------------+----------------+
|    1 | 192.168.1.100 | USA            |
|    2 | 192.168.1.101 | China          |
|    3 | 192.168.1.102 | Russia         |
|    4 | 192.168.1.103 | India          |
|    5 | 192.168.1.104 | Brazil         |
+------+---------------+----------------+
5 rows in set (0.00 sec)
```

4 detection_rules : Inserting data into detection_rules table

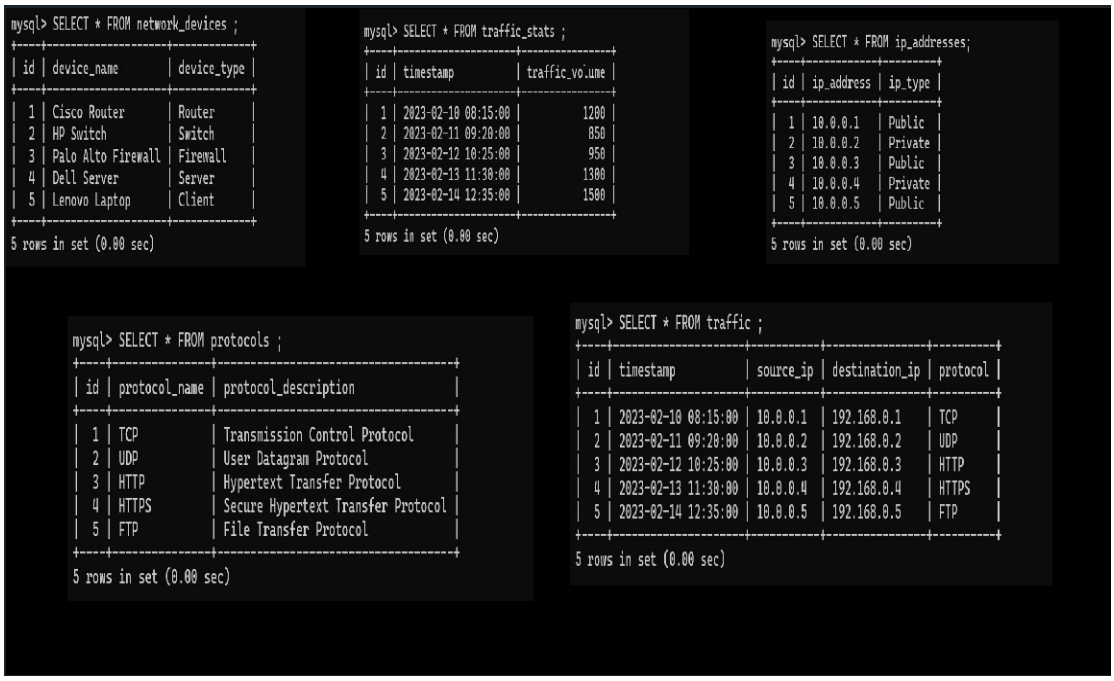
```
mysql> select * from detection_rules;
+------+-----------+----------------------+
| id   | rule_name | rule_description     |
+------+-----------+----------------------+
|    1 | Rule 1    | Detect DDoS attacks  |
|    2 | Rule 2    | Detect SQL Injection |
|    3 | Rule 3    | Detect XSS           |
|    4 | Rule 4    | Detect Brute Force   |
|    5 | Rule 5    | Detect Phishing      |
+------+-----------+----------------------+
5 rows in set (0.00 sec)
```

5 alerts table : Inserting data into alerts table

```
mysql> select * from alerts;
+------+-----------+---------------------+-------------+
| id   | attack_id | alert_date          | alert_level |
+------+-----------+---------------------+-------------+
|    1 |         1 | 2022-01-01 12:00:00 | High        |
|    2 |         2 | 2022-01-02 13:00:00 | Medium      |
|    3 |         3 | 2022-01-03 14:00:00 | Low         |
|    4 |         4 | 2022-01-04 15:00:00 | High        |
|    5 |         5 | 2022-01-05 16:00:00 | Medium      |
+------+-----------+---------------------+-------------+
5 rows in set (0.00 sec)
```

Database 2: Network_Traffic

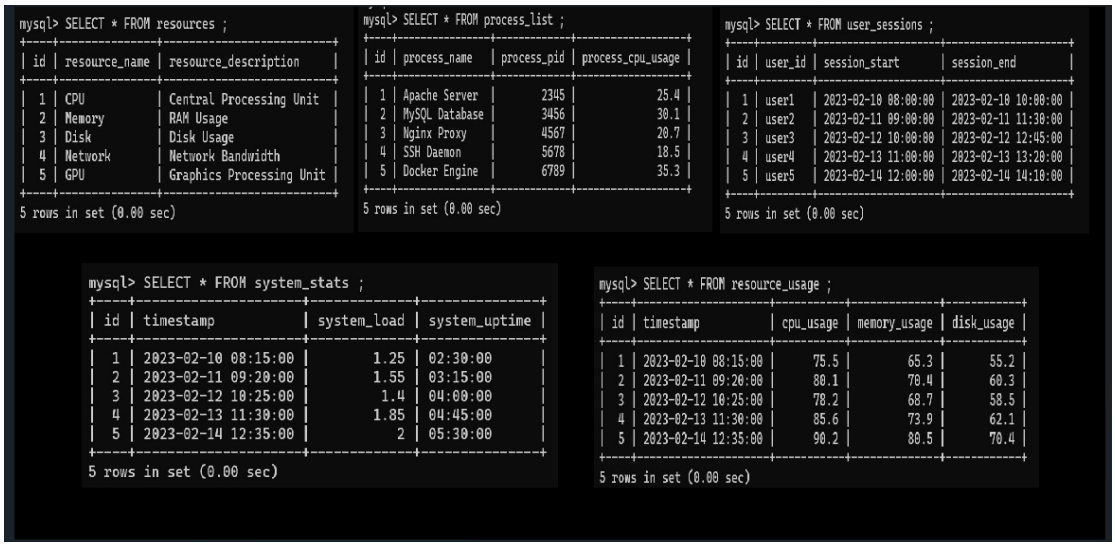
1 traffic table: Inserting data into traffic table .
2 protocols table : Inserting data into protocols table .
3 ip_addresses table : Inserting data into ip_addresses table .
4 network_devices table :Inserting data into network_devices table.
5 traffic_stats table :Inserting data into traffic_stats table.



```
mysql> SELECT * FROM network_devices ;
+----+------------------+-------------+
| id | device_name      | device_type |
+----+------------------+-------------+
|  1 | Cisco Router     | Router      |
|  2 | HP Switch        | Switch      |
|  3 | Palo Alto Firewall | Firewall  |
|  4 | Dell Server      | Server      |
|  5 | Lenovo Laptop    | Client      |
+----+------------------+-------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM traffic_stats ;
+----+---------------------+--------------+
| id | timestamp           | traffic_vo.ume |
+----+---------------------+--------------+
|  1 | 2023-02-10 08:15:00 |         1200 |
|  2 | 2023-02-11 09:20:00 |          850 |
|  3 | 2023-02-12 10:25:00 |          950 |
|  4 | 2023-02-13 11:30:00 |         1300 |
|  5 | 2023-02-14 12:35:00 |         1500 |
+----+---------------------+--------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM ip_addresses;
+----+------------+---------+
| id | ip_address | ip_type |
+----+------------+---------+
|  1 | 10.0.0.1   | Public  |
|  2 | 10.0.0.2   | Private |
|  3 | 10.0.0.3   | Public  |
|  4 | 10.0.0.4   | Private |
|  5 | 10.0.0.5   | Public  |
+----+------------+---------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM protocols ;
+----+---------------+-----------------------------------+
| id | protocol_name | protocol_description              |
+----+---------------+-----------------------------------+
|  1 | TCP           | Transmission Control Protocol     |
|  2 | UDP           | User Datagram Protocol            |
|  3 | HTTP          | Hypertext Transfer Protocol       |
|  4 | HTTPS         | Secure Hypertext Transfer Protocol |
|  5 | FTP           | File Transfer Protocol            |
+----+---------------+-----------------------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM traffic ;
+----+---------------------+-----------+----------------+----------+
| id | timestamp           | source_ip | destination_ip | protocol |
+----+---------------------+-----------+----------------+----------+
|  1 | 2023-02-10 08:15:00 | 10.0.0.1  | 192.168.0.1    | TCP      |
|  2 | 2023-02-11 09:20:00 | 10.0.0.2  | 192.168.0.2    | UDP      |
|  3 | 2023-02-12 10:25:00 | 10.0.0.3  | 192.168.0.3    | HTTP     |
|  4 | 2023-02-13 11:30:00 | 10.0.0.4  | 192.168.0.4    | HTTPS    |
|  5 | 2023-02-14 12:35:00 | 10.0.0.5  | 192.168.0.5    | FTP      |
+----+---------------------+-----------+----------------+----------+
5 rows in set (0.00 sec)
```

Database 3: System_Resources

1 resource_usage table: Inserting data into resource_usage table
2 resources table: Inserting data into resources table
3 system_stats table: Inserting data into system_stats table
4 process_list table: Inserting data into process_list table
5 user_sessions table:Inserting data into user_sessions table



```
mysql> SELECT * FROM resources ;
+----+---------------+---------------------------+
| id | resource_name | resource_description      |
+----+---------------+---------------------------+
|  1 | CPU           | Central Processing Unit   |
|  2 | Memory        | RAM Usage                 |
|  3 | Disk          | Disk Usage                |
|  4 | Network       | Network Bandwidth         |
|  5 | GPU           | Graphics Processing Unit  |
+----+---------------+---------------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM process_list ;
+----+--------------+-------------+-------------------+
| id | process_name | process_pid | process_cpu_usage |
+----+--------------+-------------+-------------------+
|  1 | Apache Server |        2345 |              25.4 |
|  2 | MySQL Database |       3456 |              30.1 |
|  3 | Nginx Proxy   |        4567 |              20.7 |
|  4 | SSH Daemon    |        5678 |              18.5 |
|  5 | Docker Engine |        6789 |              35.3 |
+----+--------------+-------------+-------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM user_sessions ;
+----+---------+---------------------+---------------------+
| id | user_id | session_start       | session_end         |
+----+---------+---------------------+---------------------+
|  1 | user1   | 2023-02-10 08:00:00 | 2023-02-10 10:00:00 |
|  2 | user2   | 2023-02-11 09:00:00 | 2023-02-11 11:30:00 |
|  3 | user3   | 2023-02-12 10:00:00 | 2023-02-12 12:45:00 |
|  4 | user4   | 2023-02-13 11:00:00 | 2023-02-13 13:20:00 |
|  5 | user5   | 2023-02-14 12:00:00 | 2023-02-14 14:10:00 |
+----+---------+---------------------+---------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM system_stats ;
+----+---------------------+-------------+---------------+
| id | timestamp           | system_load | system_uptime |
+----+---------------------+-------------+---------------+
|  1 | 2023-02-10 08:15:00 |        1.25 | 02:30:00      |
|  2 | 2023-02-11 09:20:00 |        1.55 | 03:15:00      |
|  3 | 2023-02-12 10:25:00 |         1.4 | 04:00:00      |
|  4 | 2023-02-13 11:30:00 |        1.85 | 04:45:00      |
|  5 | 2023-02-14 12:35:00 |           2 | 05:30:00      |
+----+---------------------+-------------+---------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM resource_usage ;
+----+---------------------+-----------+--------------+------------+
| id | timestamp           | cpu_usage | memory_usage | disk_usage |
+----+---------------------+-----------+--------------+------------+
|  1 | 2023-02-10 08:15:00 |      75.5 |         65.3 |       55.2 |
|  2 | 2023-02-11 09:20:00 |      80.1 |         70.4 |       60.3 |
|  3 | 2023-02-12 10:25:00 |      78.2 |         68.7 |       58.5 |
|  4 | 2023-02-13 11:30:00 |      85.6 |         73.9 |       62.1 |
|  5 | 2023-02-14 12:35:00 |      90.2 |         80.5 |       70.4 |
+----+---------------------+-----------+--------------+------------+
5 rows in set (0.00 sec)
```

Database 4: Incident_Response

1 incidents table: Inserting data into incidents table
2 incident_types table:Inserting data into incident_types table
3 response_plans table: Inserting data into response_plans table
4 response_teams table: Inserting data into response_teams table
5 incident_reports table: Inserting data into incident_reports table

mysql> select * from incident_types;

id	type_name	type_description
1	Volume-based attacks	Overwhelming the network with traffic to consume bandwidth.
2	Protocol attacks	Exploiting weaknesses in network protocols to consume resources
3	Application attacks	Targeting specific applications or services to consume resources

3 rows in set (0.00 sec)

mysql> select * from response_teams;

id	team_name	team_lead
1	Network Security Team	Alice Johnson
2	Incident Response Team	Bob Smith
3	Application Security Team	Carol Williams
4	Infrastructure Support Team	David Brown
5	Cloud Operations Team	Eve Davis

5 rows in set (0.00 sec)

mysql> select * from incidents;

id	incident_date	incident_type	incident_description
1	2024-09-01	DoS Attack	A flood of traffic overwhelmed the web server, causing downtime for 2 hours.
3	2024-09-10	Application Layer DoS	An HTTP flood attack targeted our application, leading to slow performance and eventual crash.
1	2024-09-01	Network Outage	A major network outage affected the entire office for 3 hours.
2	2024-09-05	Data Breach	Unauthorized access to customer data was detected and resolved.

4 rows in set (0.00 sec)

mysql> select * from response_plans;

id	plan_name	plan_description
1	Basic DoS Mitigation	Initial response to DoS attacks, including traffic filtering and rate limiting.
3	Application Layer Defense	Specific measures for defending against application layer attacks, such as deploying WAFs (Web Application Firewalls) and monitoring for unusual patterns.
3	Traffic Filtering	Deploy filtering solutions to detect and block excessive traffic that overwhelms network bandwidth.
4	Web Application Firewall	Deploy a WAF to protect web applications by filtering and monitoring HTTP traffic to prevent attacks targeting application services.

4 rows in set (0.00 sec)

Database 5: Security_Information

1 vulnerabilities table: Inserting data into vulnerabilities table
2 patches table: Inserting data into patches table
3 security_advisories: Inserting data into security_advisories table
4 threat_intelligence: Inserting data into threat_intelligence table
5 security_incidents :Inserting data into security_incidents table

mysql> select * from patches;

id	patch_name	patch_release_date	patch_description
1	SYN Cookie Implementation Patch	2024-03-15	Patch to enable SYN cookies to mitigate SYN flood attacks by protecting the server?s connection backlog.
2	DNS Server Hardening Update	2024-05-20	Update to harden DNS servers against amplification attacks by disabling open recursion and rate limiting queries.
3	WAF Rule Update for HTTP Floods	2024-07-10	Update to Web Application Firewall (WAF) rules to detect and block excessive HTTP requests and mitigate HTTP flood attacks.
4	Network Bandwidth Management Patch	2024-08-25	Patch to improve network bandwidth management and introduce traffic shaping to handle high volume traffic efficiently.
5	Application Resource Optimization Patch	2024-09-05	Patch to optimize resource allocation and limit excessive resource consumption in applications to prevent performance degradation.

5 rows in set (0.00 sec)

mysql> select * from security_advisories;

id	advisory_name	advisory_description
1	Mitigating SYN Flood Attacks	Advisory on implementing SYN cookies and increasing backlog queue size to defend against SYN flood attacks.
2	Protecting Against DNS Amplification	Guidance on securing DNS servers to prevent amplification attacks, including disabling open recursion and rate limiting.
3	Defending Against HTTP Flood Attacks	Recommendations for using Web Application Firewalls (WAFs) and rate limiting to mitigate HTTP flood attacks targeting web applications.
4	Handling Network Bandwidth Saturation	Advice on managing network bandwidth effectively to avoid saturation from high-volume DoS attacks, including implementing traffic shaping.
5	Preventing Application Layer Resource Exhaustion	Strategies for optimizing application performance and preventing resource exhaustion during application-layer DoS attacks, including resource throttling and efficient coding practices.

5 rows in set (0.00 sec)

mysql> select * from security_incidents;

id	incident_id	security_incident_date
1	101	2024-01-15 09:30:00
2	102	2024-02-20 14:45:00
3	103	2024-03-10 17:00:00
4	104	2024-04-05 11:15:00
5	105	2024-05-25 20:30:00

5 rows in set (0.00 sec)

mysql> select * from threat_intelligence;

id	threat_name	threat_description	threat_level
1	SYN Flood	A type of DoS attack that exploits TCP handshake by sending many SYN requests without completing the connection.	High
2	HTTP Flood	An attack where the attacker sends a high number of HTTP requests to overwhelm the web server.	Medium
3	Ping of Death	Sending malformed or oversized pings to crash the target system.	Critical
4	UDP Flood	An attack that overwhelms the target with a high volume of UDP packets, causing the system to become unresponsive.	High
5	DNS Amplification	Exploiting DNS servers to flood the target system with amplified DNS response traffic.	Critical

5 rows in set (0.00 sec)

attacks table

1. SELECT * FROM attacks;

mysql> select * from attacks;

id	attack_type	attack_date	source_ip
1	1	2022-01-01 12:00:00	192.168.1.100
2	2	2022-01-02 13:00:00	192.168.1.101
3	3	2022-01-03 14:00:00	192.168.1.102
4	1	2022-01-04 15:00:00	192.168.1.103
5	2	2022-01-05 16:00:00	192.168.1.104

5 rows in set (0.00 sec)

2. SELECT * FROM attacks WHERE attack_type = 1;

```
mysql> SELECT * FROM attacks WHERE attack_type = 1;
+------+-------------+---------------------+---------------+
| id   | attack_type | attack_date         | source_ip     |
+------+-------------+---------------------+---------------+
|    1 |           1 | 2022-01-01 12:00:00 | 192.168.1.100 |
|    4 |           1 | 2022-01-04 15:00:00 | 192.168.1.103 |
+------+-------------+---------------------+---------------+
2 rows in set (0.02 sec)
```

3. SELECT * FROM attacks WHERE attack_date BETWEEN '2023-02-10 08:15:00' AND ' 2023-02-13 11:30:00' ;

```
mysql> SELECT * FROM attacks WHERE attack_date BETWEEN '2022-01-01 12:00:00' AND '  2022-01-04 15:00:00' ;
+------+-------------+---------------------+---------------+
| id   | attack_type | attack_date         | source_ip     |
+------+-------------+---------------------+---------------+
|    1 |           1 | 2022-01-01 12:00:00 | 192.168.1.100 |
|    2 |           2 | 2022-01-02 13:00:00 | 192.168.1.101 |
|    3 |           3 | 2022-01-03 14:00:00 | 192.168.1.102 |
|    4 |           1 | 2022-01-04 15:00:00 | 192.168.1.103 |
+------+-------------+---------------------+---------------+
4 rows in set, 5 warnings (0.00 sec)
```

4. SELECT * FROM attacks WHERE source_ip = '10.0.0.2';

```
mysql> SELECT * FROM attacks WHERE source_ip = '192.168.1.102';
+------+-------------+---------------------+---------------+
| id   | attack_type | attack_date         | source_ip     |
+------+-------------+---------------------+---------------+
|    3 |           3 | 2022-01-03 14:00:00 | 192.168.1.102 |
+------+-------------+---------------------+---------------+
1 row in set (0.00 sec)
```

attack_types table

1. SELECT * FROM attack_types;

```
mysql> select * from attacks_types;
+------+----------------------+-------------------------------------+
| id   | type_name            | description                         |
+------+----------------------+-------------------------------------+
|    1 | DDoS                 | Distributed Denial of Service       |
|    2 | SQL Injection        | Structured Query Language Injection |
|    3 | Cross-Site Scripting | XSS                                 |
|    4 | Brute Force          | Password Guessing                   |
|    5 | Phishing             | Social Engineering                  |
+------+----------------------+-------------------------------------+
5 rows in set (0.00 sec)
```

2. SELECT * FROM attack_types WHERE type_name = ' Brute Force ';

```
mysql> select * from attacks_types  WHERE type_name = 'Brute Force';
+------+------------+------------------+
| id   | type_name  | description      |
+------+------------+------------------+
|    4 | Brute Force | Password Guessing |
+------+------------+------------------+
1 row in set (0.00 sec)
```

3. SELECT * FROM attack_types WHERE description LIKE '%Deniel%';

```
mysql> SELECT * FROM attacks_types WHERE description LIKE '%Denial%';
+------+-----------+-------------------------------+
| id   | type_name | description                   |
+------+-----------+-------------------------------+
|    1 | DDoS      | Distributed Denial of Service |
+------+-----------+-------------------------------+
1 row in set (0.01 sec)
```

sources table

1. SELECT * FROM sources;

```
mysql> select * from sources
    -> ;
+------+---------------+----------------+
| id   | source_ip     | source_country |
+------+---------------+----------------+
|    1 | 192.168.1.100 | USA            |
|    2 | 192.168.1.101 | China          |
|    3 | 192.168.1.102 | Russia         |
|    4 | 192.168.1.103 | India          |
|    5 | 192.168.1.104 | Brazil         |
+------+---------------+----------------+
5 rows in set (0.00 sec)
```

2. SELECT * FROM sources WHERE source_ip = '10.0.0.1';

```
mysql> SELECT * FROM sources WHERE source_ip = '192.168.1.101';
+------+---------------+----------------+
| id   | source_ip     | source_country |
+------+---------------+----------------+
|    2 | 192.168.1.101 | China          |
+------+---------------+----------------+
1 row in set (0.00 sec)
```

3. SELECT * FROM sources WHERE source_country = India;

```
mysql> SELECT * FROM sources WHERE source_country = 'India';
+------+---------------+----------------+
| id   | source_ip     | source_country |
+------+---------------+----------------+
|    4 | 192.168.1.103 | India          |
+------+---------------+----------------+
1 row in set (0.00 sec)
```

detection_rules table

1. SELECT * FROM detection_rules;

```
mysql> select * from detection_rules;
+------+-----------+----------------------+
| id   | rule_name | rule_description     |
+------+-----------+----------------------+
|    1 | Rule 1    | Detect DDoS attacks  |
|    2 | Rule 2    | Detect SQL Injection |
|    3 | Rule 3    | Detect XSS           |
|    4 | Rule 4    | Detect Brute Force   |
|    5 | Rule 5    | Detect Phishing      |
+------+-----------+----------------------+
5 rows in set (0.00 sec)
```

2. SELECT * FROM detection_rules WHERE rule_name = 'Rule XSS';

```
mysql> SELECT * FROM detection_rules WHERE rule_name = 'Rule 3';
+------+-----------+------------------+
| id   | rule_name | rule_description |
+------+-----------+------------------+
|    3 | Rule 3    | Detect XSS       |
+------+-----------+------------------+
1 row in set (0.00 sec)
```

3. SELECT * FROM detection_rules WHERE rule_description LIKE '%DDoS%';

```
mysql> SELECT * FROM detection_rules WHERE rule_description LIKE '%DDoS%';
+------+-----------+--------------------+
| id   | rule_name | rule_description   |
+------+-----------+--------------------+
|    1 | Rule 1    | Detect DDoS attacks |
+------+-----------+--------------------+
1 row in set (0.00 sec)
```

alerts table

1. SELECT * FROM alerts;

```
mysql> select * from alerts;
+------+-----------+---------------------+-------------+
| id   | attack_id | alert_date          | alert_level |
+------+-----------+---------------------+-------------+
|    1 |         1 | 2022-01-01 12:00:00 | High        |
|    2 |         2 | 2022-01-02 13:00:00 | Medium      |
|    3 |         3 | 2022-01-03 14:00:00 | Low         |
|    4 |         4 | 2022-01-04 15:00:00 | High        |
|    5 |         5 | 2022-01-05 16:00:00 | Medium      |
+------+-----------+---------------------+-------------+
5 rows in set (0.00 sec)
```

2. SELECT * FROM alerts WHERE alert_level = 'High';

```
mysql> SELECT * FROM alerts WHERE alert_level = 'High';
+------+-----------+---------------------+-------------+
| id   | attack_id | alert_date          | alert_level |
+------+-----------+---------------------+-------------+
|    1 |         1 | 2022-01-01 12:00:00 | High        |
|    4 |         4 | 2022-01-04 15:00:00 | High        |
+------+-----------+---------------------+-------------+
2 rows in set (0.00 sec)
```

3. SELECT * FROM alerts WHERE alert_date BETWEEN '2023-02-10' AND '2023-02-12';

```
mysql> SELECT * FROM alerts WHERE alert_date BETWEEN '2022-01-02 13:00:00' AND '2022-01-04 15:00:00';
+------+-----------+---------------------+-------------+
| id   | attack_id | alert_date          | alert_level |
+------+-----------+---------------------+-------------+
|    2 |         2 | 2022-01-02 13:00:00 | Medium      |
|    3 |         3 | 2022-01-03 14:00:00 | Low         |
|    4 |         4 | 2022-01-04 15:00:00 | High        |
+------+-----------+---------------------+-------------+
3 rows in set (0.00 sec)
```