

Project

**DENAIL OF SERVICE USING MYSQL RELATIONAL**

**DATABASE STRCTURE BASED ON NETWORK SECURITY**



**Prepared By: Guided By:**

**Sakshi Shaha Zakir Hussain**

## Table of Contents

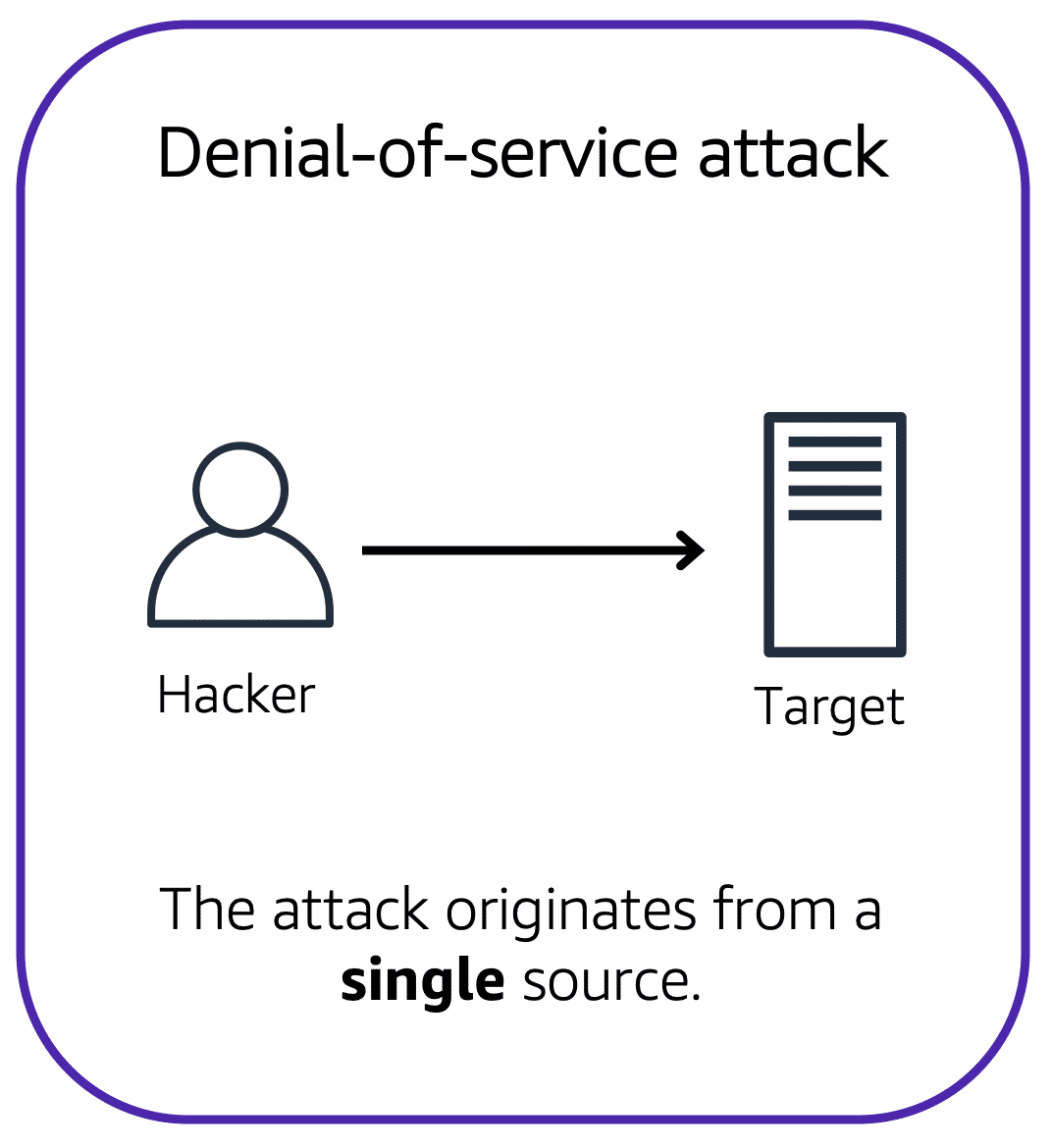
|  |  |
| --- | --- |
| **S No.** | **Content** |
| 1 | Denial-Of-Service Attack |
| 1.1 | Types of DOS |
| 1.2 | Common DOS Attack Techniques |
| 1.3 | DOS Attack Can Be Launched Using Various Tools And Techniques |
| 2 | Structured Query Language |
| 3 | Databases Used In This Project |
| 3.1 | Database 1: Attack\_Detection |
| 3.2 | Database 2: Network\_Traffic |
| 3.3 | Database 3: System\_Resources |
| 3.4 | Database 4: Incident\_Response |
| 3.5 | Database 5: Security\_Information |
| 4 | Tables Used In Each Databases |
| 4.1 | Database 1: attacks , attack types , sources , detection\_rules , alerts |
| 4.2 | Database 2: traffic , protocols , ip\_addresses , network\_devices , traffic\_stats |
| 4.3 | Database 3: resource\_usage , resources , system\_stats , process\_list , user\_sessions |
| 4.4 | Database 4: incidents , incident\_types , response\_plans ,response\_teams , incident\_reports |
| 4.5 | Database 5: vulnerabilities ,patches ,security\_advisories , threat\_intelligence , security\_incidents |
| 5 | Queries Identified By The Network Infra Security Team |
| 6 | Goal of The Project |

**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 cyber-attack 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:**

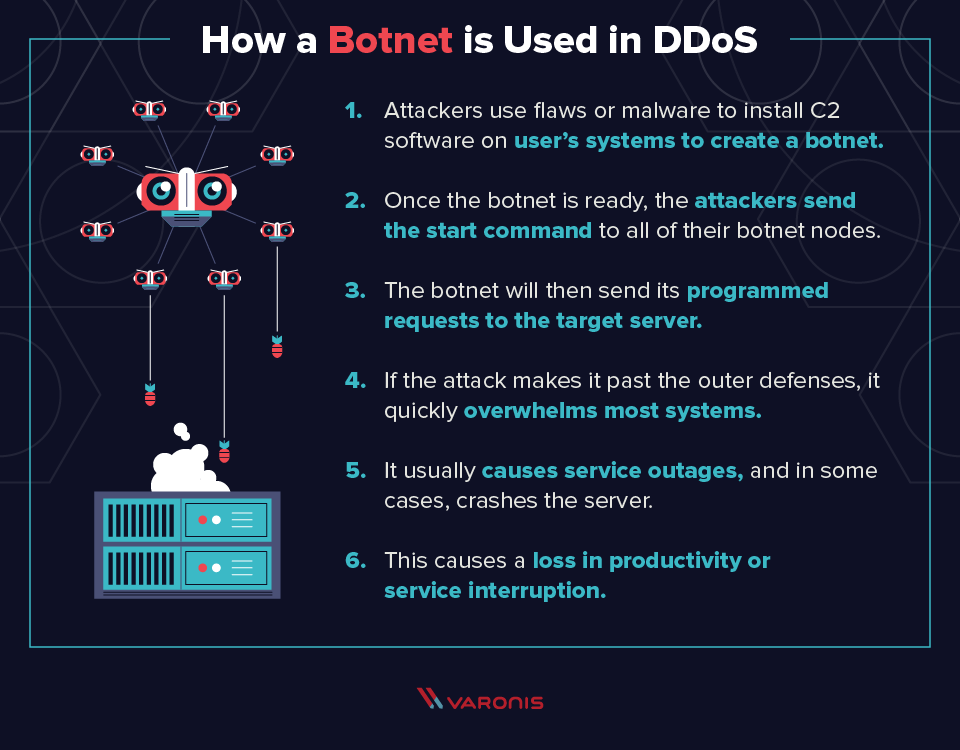
1. **Volume-Based Attacks:** Volume-based attacks flood a network with too much data, overpowering its bandwidth and making the network unusable. Examples include **UDP floods** and **ICMP floods**. In a UDP flood, attackers send many UDP packets to random ports on a server, making the server busy trying 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](https://www.geeksforgeeks.org/http-full-form/) 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.

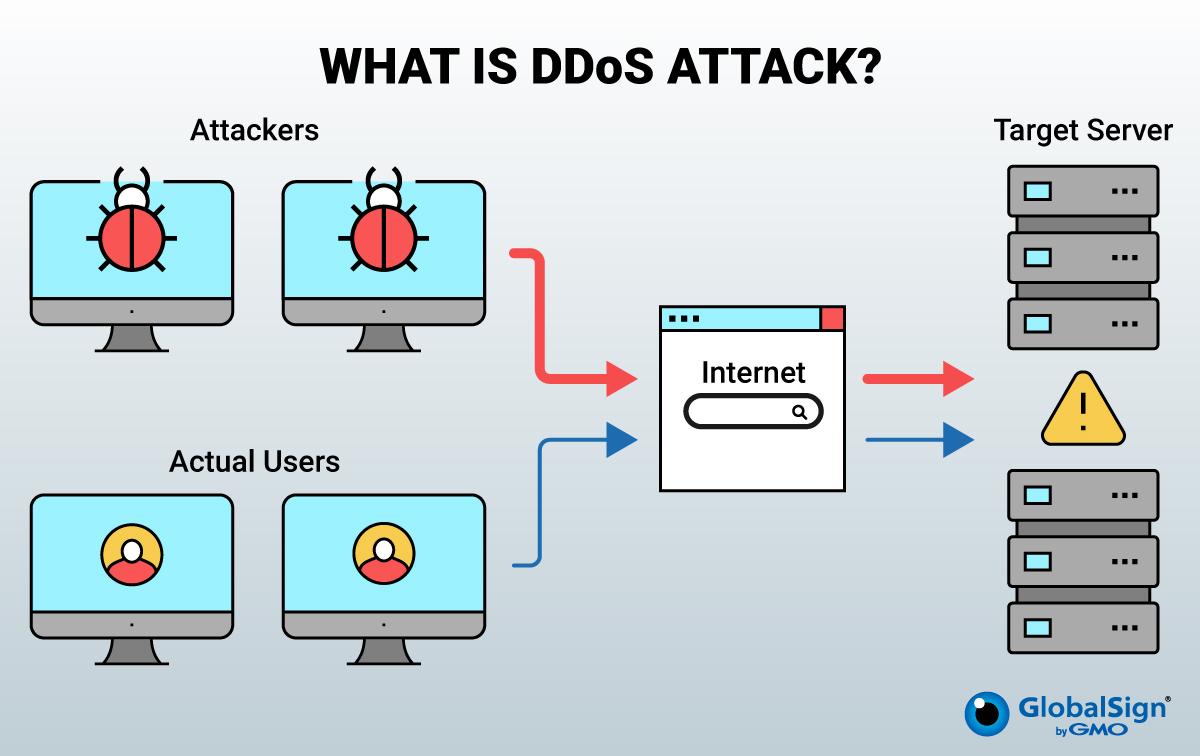


1. **Malware**: Malware is malicious software that is designed to harm or exploit systems. It can be used to infect devices within a network and turn them into part of a botnet, or to launch other forms of cyber-attacks.
2. **Scripting:** Scripting involves using scripts or automated tools to carry out attacks. This can include running scripts that flood a network with traffic or automate the process of sending malicious packets to a target.

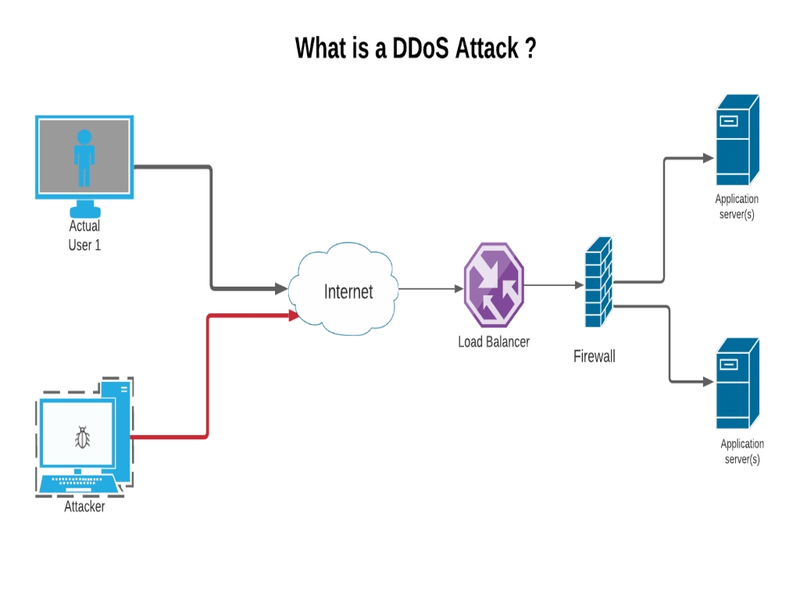
**To protect against DoS attacks, organizations can use:**

**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.

1. **Intrusion Detection/Prevention Systems (IDS/IPS):** It is a 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.
2. **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.
3. **Content Delivery Networks (CDNs):** Content Delivery Networks (CDNs) are distributed servers that cache content and deliver it to users based on their geographical location.
4. **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.

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



Database 2: Network\_Traffic



Database 3: System\_Resources



Database 4: Incident\_Response



Database 5: Security\_Information



The comand to USE a database in SQL:

USE <database\_name>;



**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** :

create table Attacks(id integer, attack\_type integer, attack\_date datetime, source\_ip varchar(30) );



2 **Create attack\_types table:**

**create table Attack\_types (id integer, type\_name varchar(100), descriptions varchar(100));**

****

**3 Create sources table:**

**create table Sources (id integer, source\_ip varchar(50), source\_country varchar(50));**



**4 Create detection\_rules table:**

create table Detection\_rules (id integer, rule\_name varchar(100), rule\_description varchar(100));



**5 Create alerts table**

**create table Alerts (id integer, attack\_id integer, alert\_date datetime, alert\_level varchar(50));**



**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**

use Network\_Traffic;



**2 protocols :**

**create table protocols (id integer, protocol\_name varchar(50), protocol\_description varchar(50));**



**3 ip\_addresses :**

**create table ip\_addresses (id integer, ip\_address varchar(50), ip\_type varchar(50));**



**4 network\_devices :**

**create table network\_devices (id integer, device\_name varchar(50), device\_type varchar(50));**



**5 traffic\_stats**

create table traffic\_status (id integer, overall\_timestamp datetime, traffic\_volume integer);



**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:



2 resources:



3 system\_stats:



4 process\_list:



5 user\_sessions:



**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

2 incident\_types

3 response\_plans

4 response\_teams

5 incident\_reports

**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

2 patches

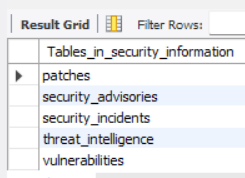
3 security\_advisories

4 threat\_intelligence

5 security\_incidents

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.

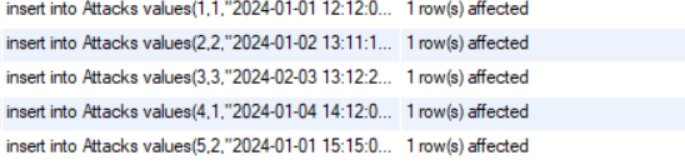
**Database 1: Attack\_Detection**

**1 Attacks Table:**

Insert data on detected attacks, including type, date, and source IP.

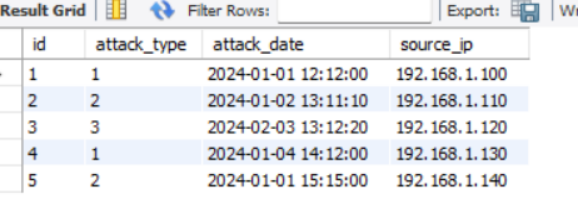
The Attacks table typically includes columns such as id (a unique identifier for each attack entry), attack\_type (the type of attack that occurred), attack\_date (the date and time when the attack occurred), and source\_ip (the IP address of the attacker). This information helps security teams track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

Inserting data into attacks table:

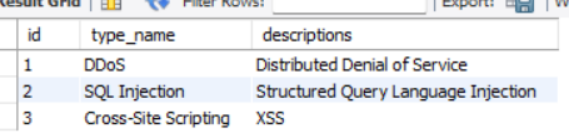


The command to DISPLAY ALL DATA of a table in SQL:

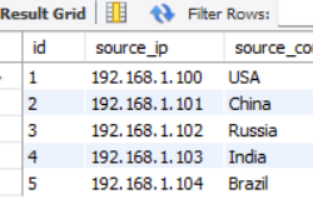
SELECT \* FROM table\_name;



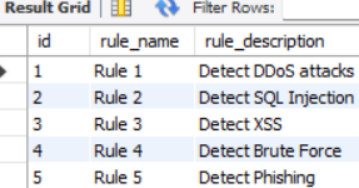
attack\_types table: Inserting data into attack\_types table



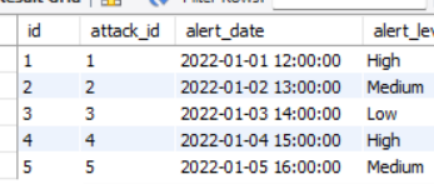
3 sources table: Inserting data into sources table



4 detection\_rules: Inserting data into detection\_rules table



5 alerts table: Inserting data into alerts table



**Database 2: Network\_Traffic**

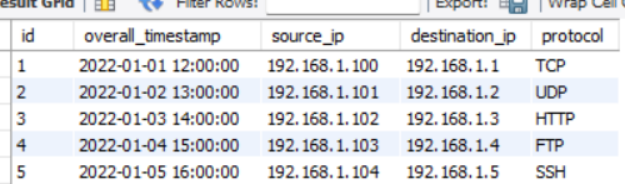
1 traffic table: Inserting data into traffic table.

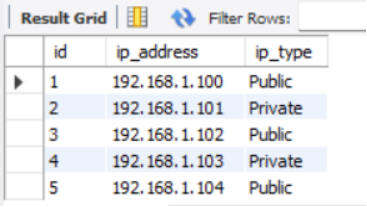
2 protocols table: Inserting data into protocols table.

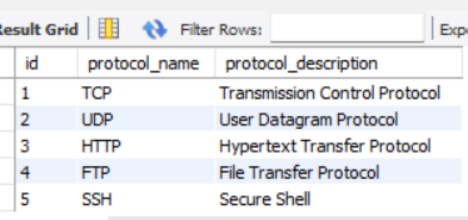
3 ip\_addresses table: Inserting data into ip\_addresses table.

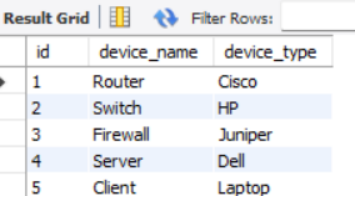
4 network\_devices table: Inserting data into network\_devices table.

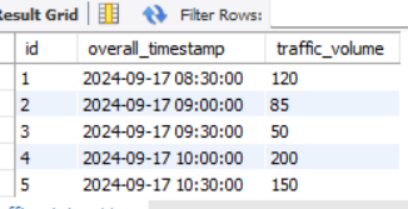
5 traffic\_stats table: Inserting data into traffic\_stats table.



****

****

****

****

**Database 3: System\_Resources**

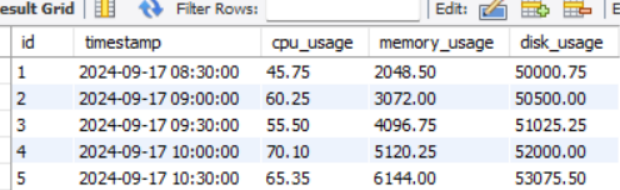
1 resource\_usage table: Inserting data into resource\_usage table

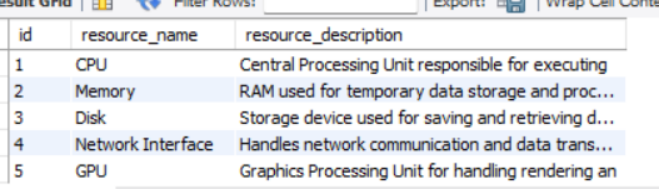
2 resources table: Inserting data into resources table

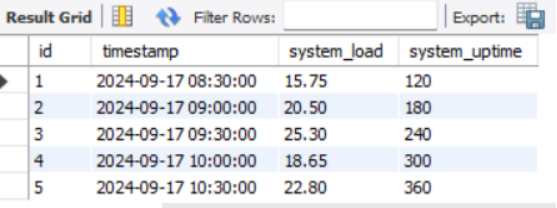
3 system\_stats table: Inserting data into system\_stats table

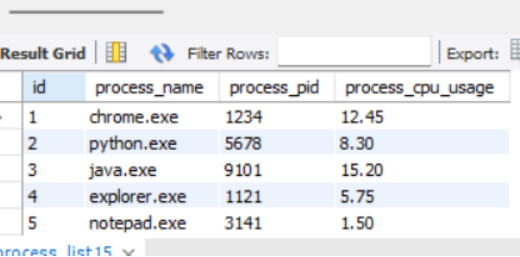
4 process\_list table: Inserting data into process\_list table

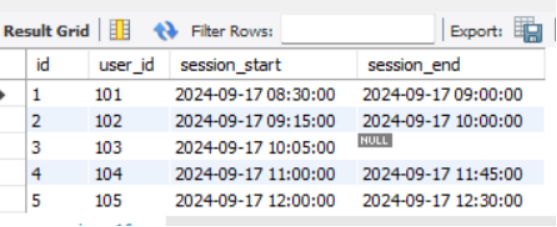
5 user\_sessions table: Inserting data into user\_sessions table



****

****

****

****

**Database 4: Incident\_Response**

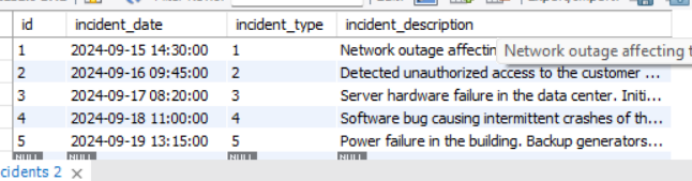
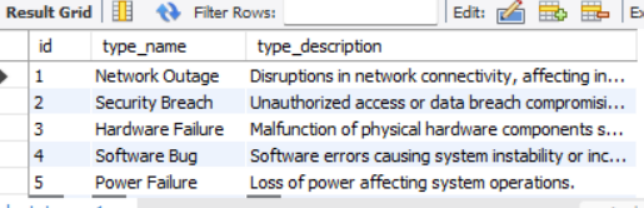
1 incidents table: Inserting data into incidents table

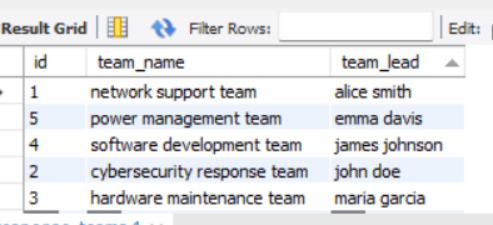
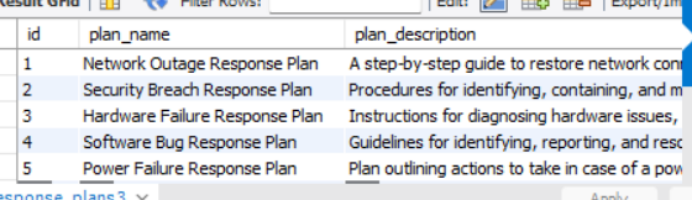
2 incident\_types table: Inserting data into incident\_types table

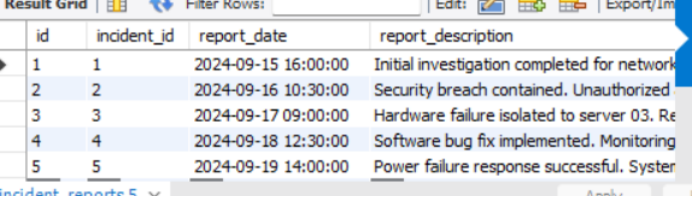
3 response\_plans table: Inserting data into response\_plans table

4 response\_teams table: Inserting data into response\_teams table

5 incident\_reports table: Inserting data into incident\_reports table



****

****

**Database 5: Security\_Information**

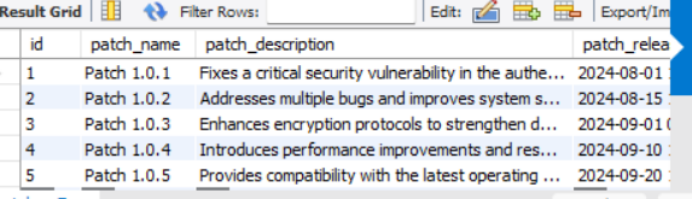
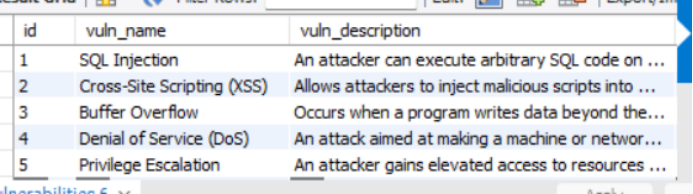
1 vulnerabilities table: Inserting data into vulnerabilities table

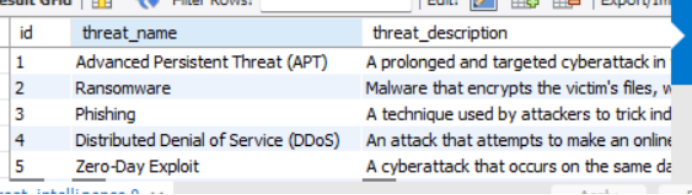
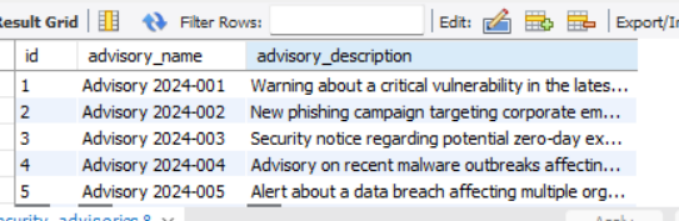
2 patches table: Inserting data into patches table

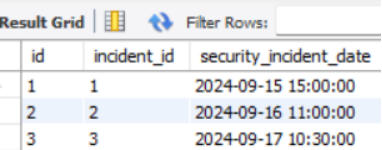
3 security\_advisories: Inserting data into security\_advisories table

4 threat\_intelligence: Inserting data into threat\_intelligence table

5 security\_incidents: Inserting data into security\_incidents table

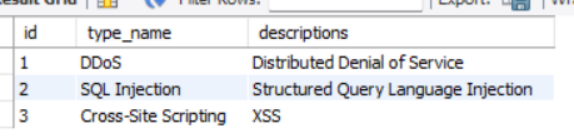


****

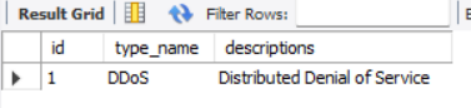
****

**Attack\_types table**

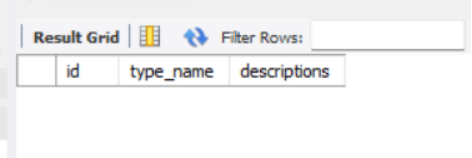
1. SELECT \* FROM attacks;



1. SELECT \* FROM ATTCK\_TYPES WHERE TYPE\_NAME = 'DDoS';

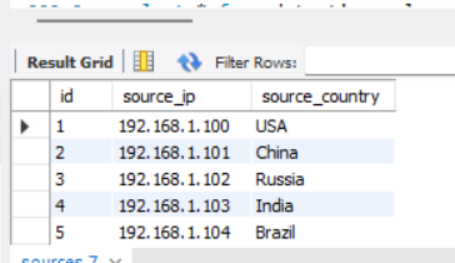


1. SELECT \* FROM ATTACK\_TYPES WHERE DESCRIPTION LIKE '%flooding%';

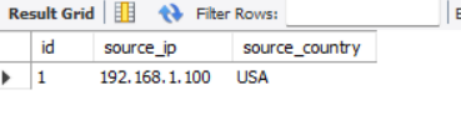


**Sources table**

1. SELECT \* FROM sources;



1. SELECT \* FROM sources WHERE source\_ip = '192.168.1.100';

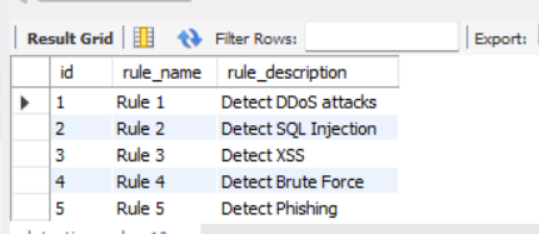


1. SELECT \* FROM sources WHERE source\_country = ‘usa’;

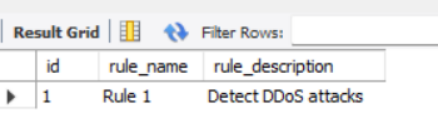
Screenshot 2024-09-17 171601

**detection\_rules table**

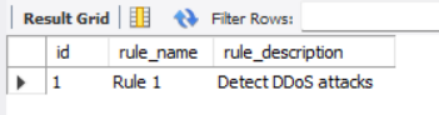
1. SELECT \* FROM detection\_rules;



1. SELECT \* FROM detection\_rules WHERE rule\_name = 'rule 1';



1. SELECT \* FROM detection\_rules WHERE rule\_description LIKE '%DDoS%';



**Alerts table**

1. SELECT \* FROM alerts;

Screenshot 2024-09-17 172700

1. SELECT \* FROM alerts WHERE alert\_level = 'High';

Screenshot 2024-09-17 172839

1. SELECT \* FROM alerts WHERE alert\_date BETWEEN '2022-01-01' AND '2022-01-31';

Screenshot 2024-09-17 172943

**Goal of This Project:**

Based on the schema and tables created, is to build a comprehensive **Cyber security Attack Detection and Response System** that integrates various components to effectively detect, analyze, and respond to security threats. Here's a breakdown of the core objectives of the project:

### ****1. Attack Detection****

* **Goal:** Detect various types of cyber-attacks in real-time, such as Distributed Denial of Service (DDoS), SQL Injection, Cross-Site Scripting (XSS), Brute Force attacks, and more.

### ****2. Incident Management****

* **Goal:** Respond to detected threats with a defined incident response plan that mitigates the impact of security breaches.

### ****3. Resource Monitoring and Optimization****

* **Goal:** Continuously monitor system resource usage (CPU, memory, disk) to ensure optimal performance and detect abnormal usage patterns that may indicate an attack.

### ****4. Threat Intelligence****

* **Goal:** Leverage external threat intelligence data to stay ahead of known and emerging threats.

### ****5. Security Automation****

* **Goal:** Automate the detection, response, and recovery from cyber-attacks to reduce human error and reaction time

### ****6. Reporting and Auditing****

* **Goal:** Generate detailed security reports and maintain logs for auditing and compliance purposes.