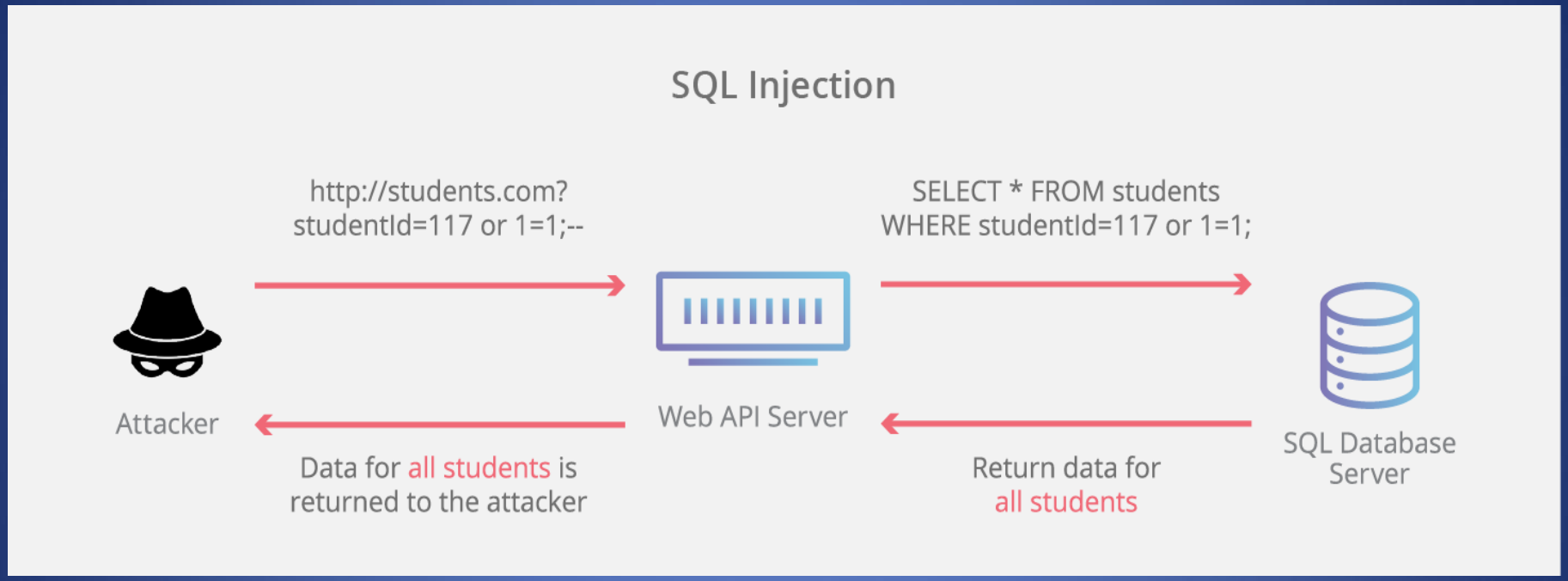
SQL Injection

Introduction

- SQL Injection is a type of an Injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application.
- Attackers can use SQL Injection vulnerabilities to bypass application security measures.
- They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database.
- They can also use SQL Injection to add, modify, and delete records in the database.

Example


SQL Injection
http://students.com?
studentId=117 or 1=1;--
SELECT * FROM students
WHERE studentId=117 or 1=1;
Attacker
Web API Server
SQL Database
Server
Data for all students is
returned to the attacker
Return data for
all students

Example

The following is the normal query to retrieve the record of a person whose mail ID is abc@xyz.com and password is 1234. If the below statement is executed against the database it provides the user information.

```
SELECT * FROM users WHERE email='abc@xyz.com' AND
  pswd=md5('1234');
```

The above code can be exploited by commenting out the password part and appending a condition that will always be true. Let's suppose an attacker provides the following input in the email address field.

xxx@xxx.xxx' OR 1 = 1 LIMIT 1 -- ']

The generated dynamic statement will be as follows.

```
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' OR 1
  = 1 LIMIT 1 -- ' ] AND pswd = md5('1234');
```

How to prevent SQL Injection Attacks

- Use Stored Procedures.
- Use prepared statements.
- Encrypt the sensitive/confidential data stored in database.
- Validate user input.
- Limit database permissions and privileges.
- Use Web Application Firewall (WAF) for web applications that access database.
- Avoid displaying database errors directly to users.