12.3.2 SQL Injection Facts

SQL injection attacks are some of the most powerful and complicated types of attacks. They are the mechanisms behind many high-profile internet attacks. Sensitive information compromised in attacks include contact information, social security numbers, and even bank account numbers. In SQL injection attacks, the vulnerability is not in the SQL software, but in the way the websites are implemented. Many attacks could be prevented with careful configuration and penetration testing on the front end.

This lesson covers the following topics:

- Web application
- Database
- Structured Query Language
- SQL injection

Web Application

Because a web application is the target of the SQL injection attack, it's important to review how web applications work. To connect to a web application, a user makes a request from the browser through the internet to the web server. The web server accepts the request and sends it to the corresponding web application. The web application accesses the database, completes the requested task, and then responds to the web server. Once the transaction has been completed, the web server sends the requested information to the user's browser.

Database

Databases store all types of data, including application data, configuration data, customer data, and login information. This is the information that attackers seek. A database is typically described by the way it stores data. A relational database can be organized in different ways depending on need. For example, customer orders can be organized in a way that would allow them to be sorted based on customer number, zip code, or product number. A distributed database is made to be easily replicated to various locations across a network. An object-oriented database is designed around object classes and sub-classes.

Inside these databases, various methods are used to organize, manage, filter, and retrieve data. Records and rows are used to represent a collection of relative data such as information about a product, a user, or a customer.

Structured Query Language

Structured Query Language (SQL) is a language that was designed specifically to request data from a database. These requests take the form of a query, basically a question, asking the database to provide information specific to a request. It's important to note that SQL injections are a result of flaws in web applications, not in the database or the web server.

SQL injection attacks target vulnerabilities, such as non-validated input, and uses them to send SQL commands through the web application to the database. This is done by injecting a code into an existing line of code before sending it onto the database for execution. Assuming the injection is successful, the malicious code could be run on the database and return the requested information.

SQL Injection

Web applications send user credentials to a web server for authentication. The web server sends the credentials within a command that tells the database to validate the username and password before granting access. If validated, the user is directed to the requested page. The data that has been entered by the user is put into the same query as the commands. As a result, this code is susceptible to SQL injection attacks.

If the login fields have not been restricted, an attacker can add anything they want to into the fields. If the attacker knows the username but not the password, the attacker may be able to enter the username, quotation mark, and double dashes in the username field. The quotation mark indicates that data has ended and a command is beginning. The double dashes indicate that code is ending and a comment is being entered. Comments are code that a program does not execute and are usually used for explanations or reminders for the coder. Because of the instructions to treat everything after the dashes as comments, the command to verify the username with a given password is no longer visible, and the attacker is granted access to the user account.

Although it's fairly easy to detect initial vulnerabilities to such an attack, SQL injections are extremely complex. They require a lot of patience and a high level of knowledge of databases and coding. However, the benefits of a successful SQL attack can be substantial. The following table lists some of the benefits of an SQL injection attack.

Benefit	Description
Remote code execution	SQL injection can help an attacker gain access and execute code on a remote operating system.
Compromise data integrity	SQL injection is used to deface a web page by inserting malicious content on it or altering the contents of the database.
Bypass authentication	An attacker uses SQL injection to log into an application with administrative level privileges without the required user name and password.

Information disclosure	SQL injection is used to access sensitive information from a database.
Compromise data availability	An attacker uses SQL injection to remove information from a database, delete logs, or alter information that has been stored in a database.