12.3.4 SQL Injection Attack Facts

In order to protect against SQL injection attacks, you need to understand the methodology for launching an SQL attack, the tools used in the attack, and how an IDS is evaded in an SQL attack.

This lesson covers the following topics:

- SQL injection methodology
- SQL injection tools
- IDS evasion

SQL Injection Methodology

The following table describes the four parts of the SQL Injection methodology.

Method	Description
Information gathering	During this step, the attacker collects information about the web application process and determines if the web application is connected to a database server. The attacker will then try injecting codes into fields to generate error messages. Whether intentionally or unintentionally generated, these error messages can provide information about the operating system, database type, database version, privilege level, and other useful information. An attacker can use the UNION operator to combine two or more statements into one. An attacker can also enter a string value where a number is requested in an input field.
Test for SQL injection vulnerabilities	Testing for SQL injection vulnerabilities includes function testing, which requires little knowledge of the inner design of the code or logic. Function testing can be as simple as adding a quotation mark or a quotation mark followed by another SQL command to the end of a URL. If an error is returned, the web application may be vulnerable to a SQL attack. Static/dynamic testing involves an analysis of the web application's source code in an attempt to find areas of the code that are vulnerable to SQL injection attacks. This can be done manually or with analysis tools. Static code analysis is completed without execution and is used to discover vulnerabilities present in the source code itself. Dynamic analysis is completed at runtime and is used to find vulnerabilities created by the interaction of the code with the SQL database and web services. During fuzz testing, a large amount of random data is input, and the output is observed to discover any coding errors.
Launch an SQL attack	In-band is the most common type of injection. It uses one communication channel to both attack and gather the results. Error-based injection is a technique that depends on error messages to obtain information about the database structure. Union-based injection uses the UNION operator to combine the results of multiple SELECT statements into one result. Blind SQL injection, also known as inferential SQL injection, is time-consuming because instead of receiving data, the results are true or false. If you've ever played 20 questions, you know that it can be difficult and tedious to investigate using true or false questions. You use the information gathered in previous questions to determine which questions to ask next, ultimately narrowing down the possibilities. Content-based blind SQL uses a query to alter the HTTP response differently for a true or false answer. Time-based blind SQL uses a query to alter the HTTP response time for a true or false answer.
Advanced SQL injection	Database, table, and column enumeration can assist in identifying user privilege level and database structure. Password grabbing involves grabbing a username and password from a user-defined table. An attacker can even transfer an entire database to his or her machine. The server is linked back to the attacker's database using OPENROWSET. The database structure is replicated and the data transferred via a remote connection. An attacker can interact with the base operating system by reading and writing system files from disk or by using direct command execution via the remote shell. Both of these options are limited by the privileges and permissions of the database. Network reconnaissance is also possible during these advanced attacks. It's possible to assess network connectivity, to gather IP information using reverse lookups, and to gather information through the use of commands such as ipconfig, tracert, and netstat.

SQL Injection Tools

The following table describes SQL injection tools.

Tool	Description
BSQL Hacker	BSQL Hacker is an injection framework that can exploit SQL injection vulnerabilities on most databases.
Havij	Havij is an SQL injection tool that an attacker can use to retrieve user and password hashes, conduct fingerprinting, access a file system, and execute commands.
Marathon Tool	An attacker can use Marathon Tool for heavy queries to complete time-based blind SQL injection attacks.
SQL	SQL Power Injector is used to find and exploit SQL injections on a web page.

Power Injector	
DroidSQLi	DroidSQLi is an injection tool for mobile devices running on an Android operating system. It allows an attacker to test MySQL web applications. It supports time-based injection, error-based injection, normal injection, and blind injection.

IDS Evasion

Given the popularity and potential damage of SQL injection attacks, an intrusion detection system (IDS) can be used to protect a network from this type of attack. The IDS will seek out known attacks. If a known pattern is detected, the administrator is alerted to the activity. However, there are ways to avoid these detection mechanisms. The most commonly used method of detection avoidance is to manipulate the input strings to avoid matching the known patterns. This can be done by converting queries into hexadecimal characters, using whitespace, using comments to break up code statements, or obscuring formatting and word use in SQL statements.

Attackers use various techniques to obscure the input strings in hopes of avoiding detection. Signature detection systems create a database of SQL injection attack strings, also known as signatures. These signatures are compared to the input strings to detect attacks. There are several types of signature evasion techniques. The following tables describes a few of these techniques.

Technique	Description
Char encoding	Char encoding uses the CHAR function to represent a character.
In-line comment	An in-line comment inserts comments between SQL keywords. As a result, the strings are obscured.
Obfuscated codes	Obfuscated codes are SQL statements that are hard to read and understand.
Manipulating white space	Input strings can be obscured by extra white space between keywords.
Hex coding	Hexadecimal code can be used to represent an SQL query.