Principles of Computer Security

Alvin Lin

January 2018 - May 2018

Data and Database Security

Databases are structured collections of data for use by one or more applications. They contain relationships between data items and groups of data items. They can contain sensitive data to be secured and use a query language to provide a uniform interface to the database. A database management system (DBMS) is a suite of programs for constructing and maintaining the database.

Relational Databases

Relational databases consist of rows and columns that hold a particular type of data. Each row has one column where all values are unique, forming a key for that row. This enables the creation of multiple tables linked together by a unique identifier that is present in all tables. A relational query language allows users to request data in this database to fit a given criteria.

Relational Database Elements

Terminology:

- A database is composed of relations/tables/files, which holds tuples/rows/records that have attributes/columns/fields.
- A primary key uniquely identifies a row and consists of one or more column names.
- A foreign key links one table to attributes in another.
- A view or virtual table shows the result of a query that returns selected rows and columns from one or more tables.

Structured Query Language (SQL)

SQL is a language for data manipulation. At its core, it supports a standard known as CRUD:

- Create data
- Read data
- Update data
- Delete data

It also defines the database structure as a data definition language, allowing for a database structure with tables, views, indexes, unique keys. Database structures can be created, modified, and deleted. The database schema generates tables stored in a data dictionary which contains metadata about how the table should store data.

Security in SQL Databases

Security with data is not merely securing the database. The data, database, DBMS must be secure along with applications, operation systems, web servers, and network environments that can access the data. Continuous patching of DBMS is needed in order to fix new vulnerabilities that come to light. The interaction of the DBMS with the OS must be secure, with secure administrative accounts, policies, and permissions. The interaction of the DBMS with the OS must be secure. Connections between the clients and the server must be secure, though this may have the side effect of limiting possible connections. The network that the DBMS is on must enforce authentication, integrity, and encryption. Generally, the database server lies behind a firewall and is separated from the web server.

Secure Application Development

Applications that access DBMS services can be subject to SQL injections. If sensitive data is stored in the database, it can be leaked if the user input is not sanitized, allowing for arbitrary queries to be executed on the DBMS.

Data Privacy

Data privacy is a field that overlaps with data security, especially confidentiality. In today’s world, there has been a dramatic increase in the amount of data aggregated and stored, often motivated by law enforcement, national security, and economic incentives. People have become increasingly aware of the access and use of their personal information. Concerns about the extent of privacy compromise have led to a variety of legal and technical approaches to protect privacy.

Security Specifications with SQL Databases

- The grant statement is used to confer authorization. It does not grant privilege to access any underlying views, and the grantor must already have permission to access the specified item before granting permission.
- Permission can be granted to select, insert, update, or delete data.
- Roles are often assigned to users to specify common classes of privileges.
- The revoke statement revokes permissions on a user or group. Revoking a user privilege may cascade to others, and this can cause interesting security issues with both timing and doubly granted permissions.
- SQL does not support tuple level authorization. Authorization to access individual data must be done by the web application on top of the database. This is often difficult because authorization loopholes can occur in multiple parts of the web application.

SQL Injection Attacks

SQL injection is the most common network based security threat to SQL databases. It exploits the basic web application design by sending malicious SQL queries to the database server. The most common use of this is bulk data extraction, but this can be used to modify data, delete data, execute arbitrary OS commands, or even launch DDOS attacks. This is often done by prematurely terminating a text string and appending a command to the end of it. Attack avenues:

- User input attack: inject SQL commands by crafting suitable user inputs to send
- Server variables attack: forge HTTP values and network headers to exploit vulnerabilities by placing data directly into the headers
- Second order injection attack: rely on existing system data to trigger an SQL injection attack, modifying the system data from within

- Cookies attack: modify an SQL query using malicious cookie data
- Physical user input attack: apply user input to construct an attack outside the realm of web requests

Inband Attacks

These attacks often use the same channel for injecting SQL code and retrieving the result. Attackers can facilitate these attacks through different methods:

- Tautology attacks inject code into conditional statements so that they always return true.
- End-of-line comments injected into an input field can nullify the legitimate code that follows.
- Piggybacked queries allow attackers to put additional queries on top of a legitimate query.

Inferential Attacks

Inferential attacks do not involve a transfer of data, but an attacker can reconstruct information by sending particular requests and observing the resulting behavior of the website or database server. Illegal or logically incorrect queries allow an attacker to gather important information about the type and structure of the backend database of a Web application. This is often done as a preliminary and information gathering step for other attacks. Blind SQL injection can also be done to infer the data present in a database even when the system is sufficiently secure to not display any erroneous information back to the attacker.

Out-of-band Attacks

Data is retrieved using a different channel than the attacking channel. This can be used when there are limitations on information retrieval, but outbound connectivity from the database server is lax.

Countermeasures

SQL injection can be countered by defensive coding, parameterized query injection, and input sanitizing. Detection of an attack can be signature based, anomaly based, or through code analysis. During run-time, queries can be checked to see if they conform to a model of expected queries.

Database Encryption

The database is typically the most valuable information resource and is protected by multiple layers of security such as firewalls, authentication, access control systems, and encryption. Encryption becomes the last line of database security defense and can be applied to the entire database, the record level, the attribute level, or to the invididual fields in the database. Encryption has the disadvantage of key management and is often inflexible due to the difficulty in record searching.

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech