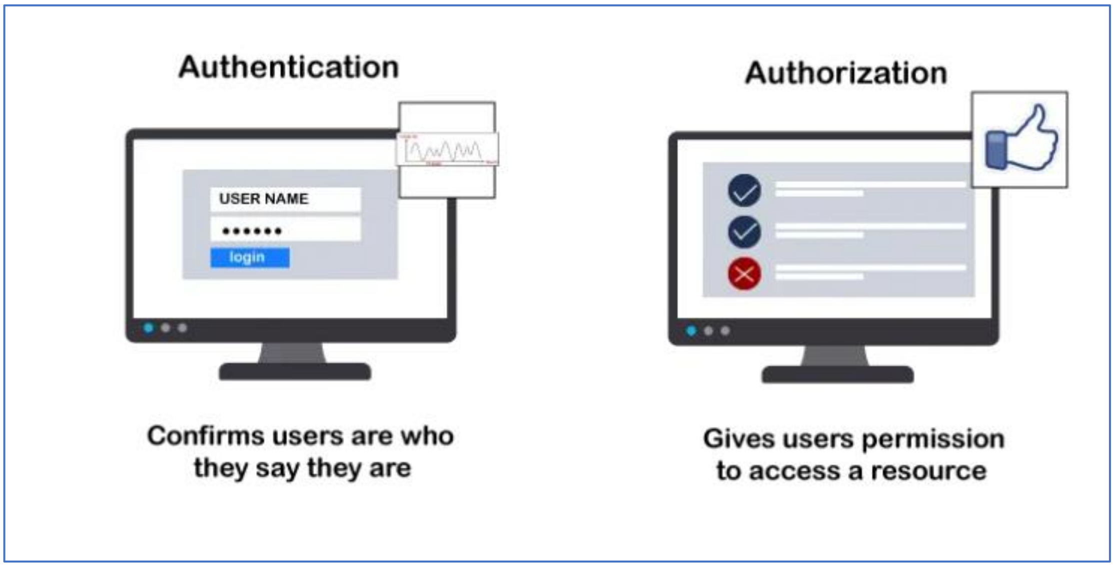
# Secure Coding Terminology

Here's a list of key secure coding terminology:

- **Authentication:** The process of verifying the identity of a user, device, or entity in a computer system.

- **Authorization:** The process of determining what an authenticated user or entity is allowed to do within a system.


Authentication
USER NAME
••••••
login
Confirms users are who they say they are
Authorization
Gives users permission to access a resource

- **Input Validation:** The practice of ensuring that user input is properly checked and sanitized before it is processed by the application to prevent malicious data from causing harm.

- **SQL Injection (SQLi):** A code injection technique that exploits vulnerabilities in an application's software by inserting malicious SQL code into an input field, allowing attackers to manipulate the database.

- **Cross-Site Scripting (XSS):** A vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users, potentially leading to data theft, session hijacking, or defacement.

- **Cross-Site Request Forgery (CSRF):** An attack that tricks a user into executing unwanted actions on a web application in which they are authenticated.

- **Buffer Overflow:** A condition where a program writes more data to a buffer than it can hold, potentially allowing attackers to execute arbitrary code or crash the system.

- **Encryption:** The process of converting data into a coded form to prevent unauthorized access, ensuring confidentiality and integrity of the data.

- **Hashing:** The process of converting data into a fixed-size string of characters, which is typically a digest that represents the data. It is commonly used for securely storing passwords.

- **Secure Sockets Layer (SSL)/Transport Layer Security (TLS):** Protocols designed to provide secure communication over a computer network by encrypting the data transmitted.

- **Least Privilege:** The principle of granting users and systems the minimum level of access necessary to perform their functions, reducing the risk of misuse or compromise.

- **Code Injection:** A general term for any attack in which an attacker supplies code to be executed by the application, exploiting vulnerabilities in the software.

- **Security by Design:** An approach to software development that integrates security considerations into the design and development process from the beginning.

- **Threat Modeling:** The process of identifying, assessing, and prioritizing potential threats to a system and defining measures to mitigate them.

- **Patch Management:** The process of managing updates and patches for software applications and systems to fix vulnerabilities and improve security.

- **Secure Coding Standards:** Guidelines and best practices that developers follow to write code that is secure and free from vulnerabilities. Examples include OWASP Secure Coding Practices and CERT Secure Coding Standards.

- **Static Code Analysis:** The examination of source code without executing it to find vulnerabilities, bugs, and compliance issues early in the development process.

- **Dynamic Code Analysis:** The analysis of running applications to identify security vulnerabilities, performance issues, and other bugs.

- **Penetration Testing (Pen Testing):** A simulated attack on a system performed by security experts to identify and fix security vulnerabilities before they can be exploited by attackers.

- **Security Incident Response:** The process of detecting, analyzing, and responding to security breaches or attacks to minimize damage and recover from the incident.

Understanding and implementing these secure coding terminologies and practices are crucial for developing robust and secure applications.