Security in WEB

Authentication

Authentication is the process of verifying the identity of a user or system to ensure that they are who they claim to be. It involves confirming the validity of a user's credentials, such as their username and password. Authentication is important in computer security to prevent unauthorised access to systems, applications, or data.

Authorisation

Authorisation is the process of granting or denying access to a resource or system based on the verified identity of the user or system. It is the second step in the security process, following authentication, and involves determining what level of access a user or system is permitted to have based on their authenticated identity and assigned permissions. Authorisation ensures that users or systems can only access the resources that they are allowed to, and helps to prevent unauthorised access and data breaches.

Authentication Types

Basic Auth

The most common form of authentication, Single-Factor Authentication, is also the least secure, as it only requires one factor to gain full system access. It could be a username and password, pin-number or another simple code

2FA

By adding a second factor for verification, two-factor authentication added extra security layer. It is an added layer that essentially double-checks that a user is, in reality, the user they're attempting to log in as—making it much harder to break.

Hawk Auth

Hawk authentication is a protocol for securing HTTP-based communications between clients (such as web browsers or mobile apps) and servers. It is designed to be simple, flexible, and secure, and is commonly used in modern web applications.

Hawk authentication works by using a shared secret key, which is known only to the client and the server, to create a message authentication code (MAC) for each request. The MAC is then included in the HTTP Authorization header, along with other information about the request, such as the timestamp and a nonce (a one-time-use value).

The server can then use the shared secret key to verify the authenticity of the MAC, ensuring that the request was not tampered with or intercepted by an attacker. This helps to prevent a range of attacks, including replay attacks, man-in-the-middle attacks, and cross-site request forgery (CSRF) attacks.

Hawk authentication is often used in conjunction with other security protocols, such as TLS/SSL (Transport Layer Security/Secure Sockets Layer), to provide end-to-end encryption and further enhance the security of web communications.

https://www.youtube.com/watch?v=H4mfCzLKy5c&ab_channel=AFUPPHP

Bearer Auth

Bearer token authentication is a method of authenticating clients in a networked environment, such as a web API or a mobile app. It involves the use of a bearer token, which is a security token that is issued by the server and presented by the client to gain access to protected resources or services.

Digest Auth

In digest authentication, the server sends a challenge to the client, which the client then encrypts using a one-way hash function and sends back to the server. The server then compares the encrypted response with its own version of the hash and if they match, the user is authenticated and granted access.

Digest authentication is often used in web applications to protect sensitive information such as passwords and credit card numbers. It is considered to be more secure than basic authentication, which sends passwords in plain text, making them vulnerable to interception by hackers.

The challenge is a randomly generated value that is sent to the client along with a nonce (number used once) and other parameters. The client then uses this information, along with the user's credentials, to create a response that is sent back to the server for verification. The challenge is designed to be unpredictable and unique, which helps to prevent replay attacks by attackers who may intercept the authentication request and try to use it later to gain unauthorized access.

SASL Auth

SASL (Simple Authentication and Security Layer) auth is a method of authenticating users in a networked environment, such as email or instant messaging systems.

With SASL, a client application (like an email program) communicates with a server (like an email server) to authenticate a user's identity. This can involve exchanging credentials, such as a username and password, or other forms of authentication, such as a security certificate.

SASL auth is a secure way of verifying that a user is who they say they are before granting them access to resources or services on a network. It helps to prevent unauthorised access and protect sensitive information.

How SASL is different from Basic Auth?

SASL (Simple Authentication and Security Layer) and Basic Auth are both methods for authenticating users, but they differ in how they transmit user credentials over a network.

With Basic Auth, the user's credentials (such as a username and password) are sent in plain text over the network, which can be intercepted and read by anyone who has access to the network traffic. This makes Basic Auth less secure and vulnerable to attacks like eavesdropping.

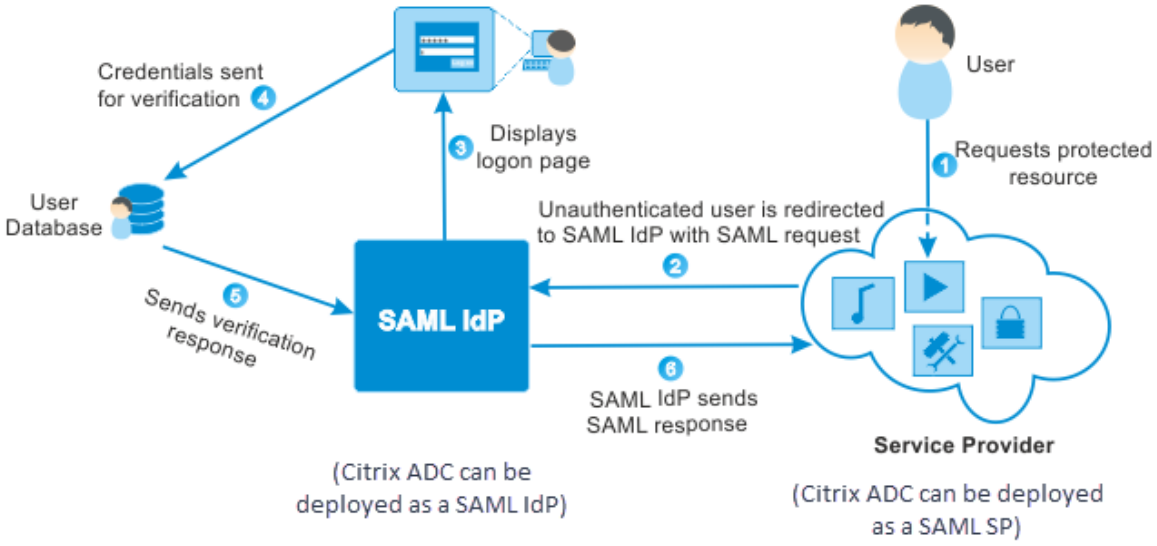
In contrast, SASL Auth uses a variety of encryption and security protocols to protect the user's credentials as they are transmitted over the network. This makes SASL Auth more secure and less vulnerable to attacks like eavesdropping.

SAML Auth

Security Assertion Markup Language (SAML) is an XML-based authentication mechanism that provides single sign-on capability and is defined by the OASIS Security Services Technical Committee.

Identity Provider — Performs authentication and passes the user's identity and authorization level to the service provider.

Service Provider — Trusts the identity provider and authorizes the given user to access the requested resource.



Open Authorization (OAuth): It is an open-standard authorization protocol that transfers identification information between apps and encrypts it into machine code. This enables users to grant an application access to their data in another application without them having to manually validate their identity—which is particularly helpful for native apps.