Introduction to Authentication Mechanisms

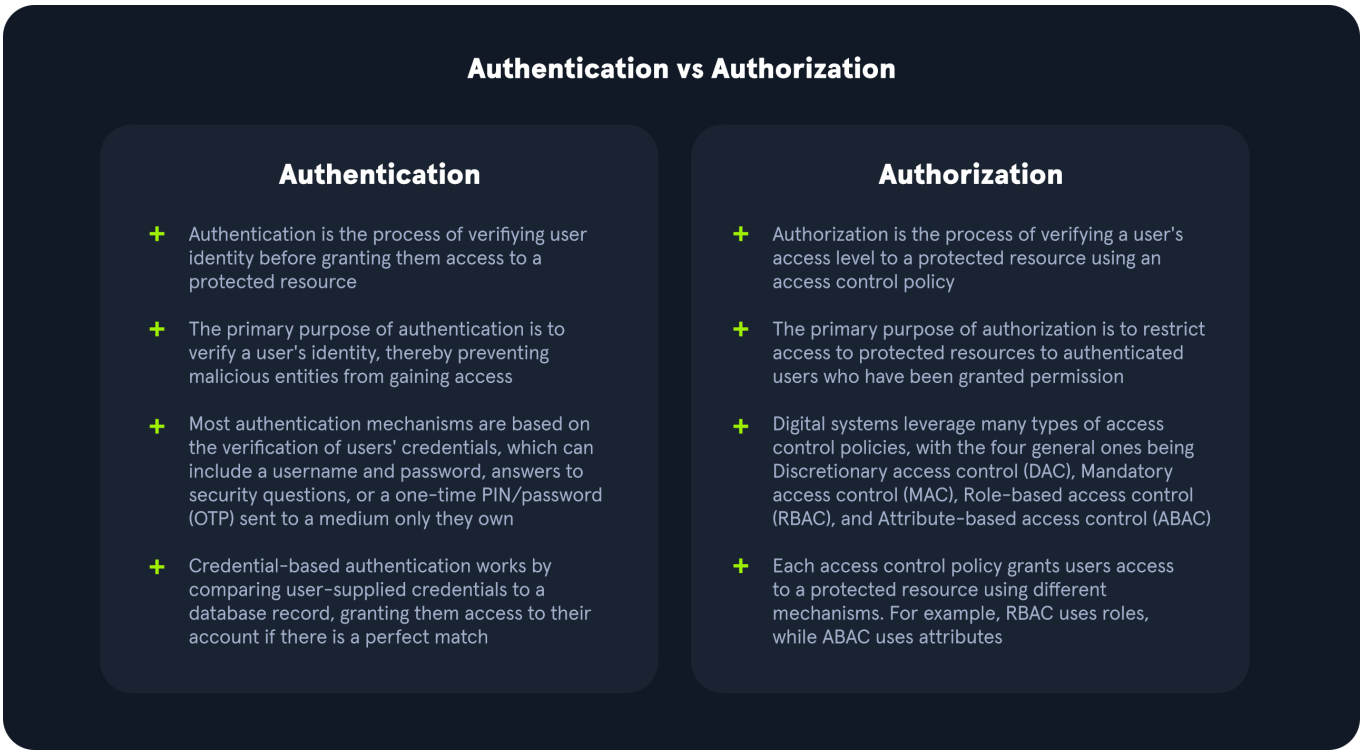
Introduction to Authentication Mechanisms

Organizations aim to streamline the user experience, allowing users to access multiple applications and websites by logging in only once. They may also want to reduce the number of disparate authentication and authorization silos for ease of management and to enforce standard policies. Frameworks such as OAuth, OpenID Connect, and SAML can help organizations build secure and standard authentication and authorization flows.

Authentication vs. Authorization

Authentication is the process of `confirming a user's identity`. The most common form of authentication is checking a user's username and password. For instance, the user confirms their identity to the website by providing a username and password.

On the other hand, `Authorization` relates to `a user's permissions` or their access level. Authorization is typically governed by an access control policy, with the four general ones being Discretionary access control (DAC), Mandatory access control (MAC), Role-based access control (RBAC), and Attribute-based access control (ABAC). RBAC, an access control policy used for web applications, relies on roles to grant users different permissions. For instance, an admin user might have the "writer" role which allows changing content on a website (write permission), while a regular user might have the "reader" role, which allows only reading the content (read permission). Proper authorization checks ensure that a regular user cannot obtain write permissions to the site's content.



Broken Authentication

It's not uncommon to find incorrectly implemented access control mechanisms. The impact ranges from the disclosure of sensitive information to the compromise of the underlying system. For example, if we compromise an application's ability to identify the requesting user via its API, this compromises the overall web application security.

Authentication mechanisms can be compromised in many ways, including:

- Brute-forcing the login page with a list of usernames and passwords
- Manipulating unsigned or weakly signed session tokens such as JWT
- Exploiting weak passwords and encryption keys
- Obtaining authentication tokens and passwords from a URL

The Broken Authentication module covered basic techniques to attack authentication mechanisms, and this module will focus on some of the more advanced authentication attacks that rely on common standards or frameworks.

JWT

JSON Web Token (JWT) is a format for transmitting cryptographically secure data. While JWTs are not directly tied to authentication in web applications, many web applications use JWTs as a stateless session token. These tokens, encoded as JSON objects, are a secure and efficient

way to transmit information between a client and a server. JWTs consist of three main parts: a header, a payload, and a signature, enabling authentication, authorization, and stateless information exchange. They have become popular for implementing token-based authentication and authorization mechanisms due to their simplicity, flexibility, and widespread support across different programming languages and platforms.

OAuth

OAuth is an open standard protocol that allows secure authorization and delegation of access between different web services without sharing user credentials. It enables users to grant third-party applications limited access to resources on other web services, such as social media accounts or online banking, without exposing their passwords. OAuth operates through token-based authentication processes, facilitating seamless interactions between service providers and consumers while maintaining user privacy and security. Widely adopted across various industries, OAuth has become the de facto standard for enabling secure API access and authorization in modern web and mobile applications.

SAML

Secure Assertion Markup Language (SAML) is an XML-based open standard for exchanging authentication and authorization data between identity providers (IdPs) and service providers (SPs). SAML enables single sign-on (SSO), allowing users to access multiple applications and services with a single set of credentials. In the SAML workflow, the user's identity is authenticated by the IdP, which then generates a digitally signed assertion containing user attributes and permissions. This assertion is sent to the SP, which validates it and grants access accordingly. SAML is widely used in enterprise environments and web-based applications to streamline authentication processes and enhance security through standardized protocols and assertions.