Name: Jomalyn N. Enriquez Integration System

Year & Section: 3B

1. Describe the difference between authentication and authorization, and explain why they are both important in web application security.

* Authentication is a process that verifies that someone or something is who they say they are. while Authorization is the security process that determines a user or service’s level of access.
* They are both important to secure sensitive data, stop unauthorized operations, assure regulatory compliance, limit threats, and establish responsibility. And in order to protect web applications from illegal access and uphold data's confidentiality, integrity, and availability authentication and authorization must be implemented.

1. What are some common techniques used for authentication and authorization in Node.js applications? Provide examples of each.

* Some common techniques used for authentication and authorization in Node.js applications are Username and Password Authentication - Users use this Authentication in facebook accounts, gmail accounts, etc. Token-based Authentication - A lot uses Token-based Authentication in fingerprint to unlock a phone.

1. Explain what JSON Web tokens (JWTs) are and how they are used in Node.js application. How do they improve security compared to other authentication methods?

* JWTs provide stateless, decentralized, tamper-proof, and flexible nature that is why it is popular to use as authorization and authentication. It also provides improved security compared to other authentication methods by reducing server-side storage requirements, enabling scalability, ensuring integrity and tamper-proofing, and offering flexibility in handling authentication and authorization requirements.

1. What is OAuth 2.0 and how is it used for authentication and authorization in Node.js application? Describe the different OAuth 2.0 flows and how they are used in practice.

* Open Authorization is designed to permit a website or application to access resources stored on other web apps on the user's behalf.
* It is used for authentication in a Node.js application by making it possible for users to log in using a third-party identity provider (like Google, Facebook, or GitHub), after which they can get an access token that can be used to access secured services or APIs on their behalf. It is used for authorization in a Node.js After the user has authenticated with a third-party identity provider, the application can grant the user access to their protected resources or APIs.

1. What is role-based access control (RBAC), and how is it used for authorization in Node.js application? Provide an example of how you might use RBAC to control access to a specific resource or functionality.

* It is a used to manage permissions and access control based on predefined roles. As an illustration, the resource "create Post" in our Node.js web application enables users to write new blog posts. We may use RBAC to define roles, assign roles, define permissions, check permissions, and manage authorization issues in addition to controlling access to these particular resources. By implementing RBAC in this way, the Node.js web application may improve security and prevent unauthorized access to sensitive functionality, making sure that only users with the appropriate roles and permissions are allowed to create, modify, or delete blog entries.

1. Describe the different techniques used for secure password storage in Node.js applications, and explain why secure password storage is important.

* Hashing - Hashing is the process of transforming any given key or a string of characters into another value because passwords are frequently the first line of defense against unauthorized access to user accounts, secure password storage is crucial. In the event of a data breach, passwords that are not stored securely can be easily compromised, allowing for unauthorized access to user accounts, identity theft, and other security risks.

1. What is two-factor authentication (2FA), and how can it be implemented in Node.js applications? Explain the benefits and potential drawbacks of 2FA.

* It is two-step verification or dual-factor authentication that is a security process in which users provide two different authentication factors to verify themselves. It can be implemented in SMS-based 2FA, time-based One-Time Password (TOTP), Authenticator Apps, Hardware Tokens. The benefits are it can Enhanced security and Protection against password-related risks. Usability and user experience: Implementing 2FA can add an extra step to the login process, which may inconvenience some users and potentially result in lower user adoption.

1. How can session management be used to improve authentication and authorization in Node.js applications? Describe some common techniques used for session management, and explain why they are important for security.

* Session management can be used to improve authentication and authorization in Node.js applications by monitoring a user's session state and managing their authentication and authorization status throughout their session. Session management can be used to improve authentication and authorization in Node.js applications by monitoring a user's session state and managing their authentication and authorization status throughout their session.

1. Describe a specific security vulnerability that could be exploited in a Node.js application that does not properly handle authentication or authorization. How might an attacker exploit this vulnerability, and what steps can be taken to prevent it?

* "Insecure Direct Object References" (IDOR) vulnerability occurs when an application allows direct access or manipulation of objects or resources without proper authentication or authorization checks. This could allow an attacker to access or manipulate sensitive data or perform actions on behalf of another user without proper authorization.
* An attacker could exploit the Insecure Direct Object References (IDOR) vulnerability in a Node.js application by manipulating the URLs or input parameters to access or manipulate unauthorized resources or actions. To prevent from attackers, it must be Implement proper authentication and authorization checks, Use RBAC or ABAC, Validate and sanitize user input, Use unique, unpredictable object references, Use secure session management, Follow the principle of least privilege, Regular security testing and monitoring.

1. Explain why it is important to regularly review and update the authentication and authorization mechanisms used in a Node.js application. What are some common issues that can arise if these mechanisms are not kept up-to-date?

* It is important to maintain to review and update the authentication and authorization for the

is crucial for maintaining the security and integrity of the application, addressing vulnerabilities, aligning with changing business requirements, managing user access effectively, complying with regulatory requirements, and incorporating new technologies and best practices. It is a proactive approach towards ensuring the security of the application and mitigating potential risks.

* Not keeping authentication and authorization mechanisms up-to-date can lead to security vulnerabilities, non-compliance, increased risk of unauthorized access, inadequate user management, reduced flexibility, and missed security enhancements. It is crucial to regularly review and update these mechanisms to ensure the security and integrity of a Node.js application.