API SECURITY INTERVIEW QUESTIONS & ANSWERS

Prepared by HANIM EKEN

https://ie.linkedin.com/in/hanimeken

1. **What is API security, and why is it important?**

API security refers to the measures taken to protect Application Programming Interfaces (APIs) from potential threats and vulnerabilities. APIs act as intermediaries that allow different software applications to communicate and exchange data. Ensuring API security is crucial because APIs are often exposed to the internet, making them susceptible to attacks, data breaches, and unauthorized access. A breach in API security can lead to the exposure of sensitive information, financial loss, and reputational damage.

2. **What are the common security vulnerabilities associated with APIs?**

- Injection attacks (e.g., SQL injection, Command injection).
- Cross-Site Scripting (XSS) attacks.
- Cross-Site Request Forgery (CSRF) attacks.
- Broken authentication and session management.
- Insecure direct object references.
- Lack of proper access controls.
- Improper error handling and information disclosure.
- Insufficient encryption and data protection.

3. **How can you prevent SQL injection attacks on APIs?**

- Use parameterized queries: Instead of directly embedding user inputs into SQL queries, use parameterized queries with placeholders to separate data from the query itself.
- Input validation: Validate and sanitize all user inputs to ensure they meet the expected format and don't contain malicious code.
- Least privilege principle: Ensure that the API only has the necessary permissions to access the required database resources and nothing more.

4. **What is OAuth, and how does it improve API security?**

OAuth (Open Authorization) is an open-standard authorization framework that allows applications to securely access resources on behalf of users. It enables users to grant limited access to their resources (e.g., data, profile) to third-party applications without sharing their credentials.
OAuth improves API security by eliminating the need for applications to store user passwords. Instead, applications receive access tokens after user authorization, which they can use to access specific resources for a limited time. This way, even if the access token gets compromised, it has a short lifespan and limited permissions, reducing the potential damage.

5. **How can you secure APIs using HTTPS**?

- Enabling SSL/TLS: Use HTTPS (HTTP over SSL/TLS) to encrypt data transmitted between the client and the server. SSL/TLS ensures data integrity, confidentiality, and authenticity.
- Certificate management: Obtain an SSL certificate from a trusted Certificate Authority (CA) and keep it up to date. Regularly check for certificate expiration and renew as necessary.
- Disable insecure protocols: Disable older and insecure SSL/TLS versions (e.g., SSLv2, SSLv3, TLSv1.0, TLSv1.1) to prevent potential vulnerabilities.

6. How can you monitor and log API activities for security purposes?

Monitoring and logging API activities are crucial for security analysis and incident response. Some practices include:

- API logs: Implement logging mechanisms to capture all API requests and responses along with relevant metadata.
- Real-time monitoring: Use intrusion detection systems and security information and event management (SIEM) tools to monitor real-time API traffic for suspicious activities.
- Anomaly detection: Set up alerting systems to notify administrators of unusual API activities or potential security breaches.
- Regular audits: Conduct periodic security audits of API usage and access logs to identify patterns of misuse or unauthorized access.

7. How can you handle authentication and authorization in API security?

- Authentication: Use robust authentication mechanisms, such as OAuth, API keys, or JWT (JSON Web Tokens), to verify the identity of users and applications accessing the API.
- Authorization: Implement role-based access controls (RBAC) to define what actions each authenticated user or application is allowed to perform. Limit access to sensitive API endpoints based on user roles and permissions.

API security is an ongoing process, and staying up to date with the latest security best practices and regularly patching vulnerabilities is crucial to maintaining a secure API environment.

8. How can you handle API versioning to maintain security?

API versioning helps ensure backward compatibility and security. Implement versioning in the URI or request headers. This way, you can make updates to the API without affecting existing clients, giving them time to migrate to newer, more secure versions.

9. What are some common types of API attacks?

Several types of API attacks exist. Some of the common ones include:

- Injection attacks: Malicious code is injected into API requests to manipulate data or gain unauthorized access.
- Denial of Service (DoS) and Distributed Denial of Service (DDoS): Overwhelming the API with excessive requests, causing it to become unresponsive or unavailable.
- Man-in-the-Middle (MITM) attack: An attacker intercepts and potentially alters the data exchanged between the client and the API server.
- Broken Object-Level Authorization: Exploiting weaknesses in object-level authorization to access unauthorized resources.
- API Spoofing: Impersonating a legitimate client application to gain access to the API.

- Token-based attacks: Exploiting weaknesses in token-based authentication mechanisms, such as JSON Web Tokens (JWT).

10. How can you protect against API injection attacks?

a. Use parameterized queries and prepared statements to separate data from the API commands.
b. Employ input validation and data sanitization to block or neutralize malicious input. Implement strict type checking for incoming data to prevent unexpected behavior.
c. Regularly update and patch your API components to avoid known vulnerabilities.

Remember to provide detailed and well-explained answers during the interview to showcase your understanding of API security concepts and best practices. Good luck!

HANIM EKEN

https://ie.linkedin.com/in/hanimeken