**OWASP API Security Top 10**

| **STRIDE** | **OWASP** | **Potential Threat** | **Threat Examples** | **Mitigation** |
| --- | --- | --- | --- | --- |
| **Spoofing** | [**API2:2023 - Broken Authentication**](https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/) | **No or weak authentication**  Exposed session ID in the URL  Store passwords in plain text | An attacker intercepts an e-commerce App authenticated user's session, steals the session ID, and misuses the session ID pretending to be the actual user | Implement API Key based authentication and token generation at the centralized Identity provider. Prefer JWT for tokens |
| Use a standard Vault for API Key management (revoke, rotate, generate keys) |
| **Elevation of Privilege** | [**API1:2023 - Broken Object Level Authorization**](https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/) | **No or weak Access Control**  Poorly implemented role checks  A user gains access to an object (file or an account) that he is not supposed (authorized) to access (BOLA)  Exposed Admin-end point (BFLA) | A user gains access to a premium version of a music album without any upgrade (BOLA)  A non-Admin user of a music application can invoke a sensitive function (delete music album) (BFLA) | Implement function level access control check to verify whether the request is allowed e.g. for removing data, validate if the user has permission to remove that data. |
| **Tampering** | [**API7:2023 - Server Side Request Forgery**](https://owasp.org/API-Security/editions/2023/en/0xa7-server-side-request-forgery/)[**API8:2023 - Security Misconfiguration**](https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/) | **Input / Output Attacks**  The back-end server’s restricted URL (endpoint) is left accessible through the middle tier that is detected & misused by the attacker | An attacker sends a request from a front-end App e.g. (Fetch(<http://localhost/admin-config>) and middle-tier performs the Fetch operation without validating the input of Fetch | Validate input: length/range/format and type/Content type/File type and size etc |
| If a request from a client is supposed to send a URL then use a whitelist of allowed URLs to validate the URL |
| **Tampering,**  **Information Disclosure** | [**API8:2023 - Security Misconfiguration**](https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/) | **No or Weak Cryptography or**  **Insecure Communication** | Default username/password for an Admin account were not removed from an E-Commerce application that allowed an attacker to abuse and login as Admin | Must only provide HTTPS endpoints |
| Must only provide HTTPS for call-back or webhook end-points |
| **Repudiation** |  | **Error Logging and Auditing** | No error logs were found after a sensitive operation (e.g. money withdrawal) was performed from a stock purchase App and no record of who performed the operation | Respond with generic error messages - avoid revealing details of the failure unnecessarily |
| Write audit logs before and after security-related events |
| Consider logging token validation errors in order to detect attacks |
| **Information Disclosure** | [**API3:2023 - Broken Object Property Level Authorization**](https://owasp.org/API-Security/editions/2023/en/0xa3-broken-object-property-level-authorization/)[**API8:2023 - Security Misconfiguration**](https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/) | **Data breach**  Access or modify a property that should be private for a specific user-(BPLA | My social media account has a property (email) that should be visible only to “Close” friends but it is visible to “Acquaintances” friends  The default username/password for an Admin account was not removed from an E-Commerce application that allowed an attacker to abuse and log in as Admin | Passwords, security tokens, and API keys should not appear in the URL. |
| In POST/PUT requests sensitive data should be transferred to the request body or request headers |
| In GET requests sensitive data should be transferred in an HTTP Header |
| **Tampering,**  **Denial of Service,**  **Repudiation** | [**API8:2023 - Security Misconfiguration**](https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/) | **Misconfigura- tion**  Verbose Error message | The default username/password for an Admin account was not removed from an E-Commerce application that allowed an attacker to abuse and log in as Admin | Make sure management Endpoints are not left exposed (public) |
| Enable CORS headers and specify the allowed origins (domains/sub-domains) to restrict  unsolicited HTTP requests to the API |
| **References:** [**https://cheatsheetseries.owasp.org/cheatsheets/REST\_Security\_Cheat\_Sheet.html**](https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html)  [**https://owasp.org/www-project-api-security/**](https://owasp.org/www-project-api-security/) | | | | |