Security analysis

Kwetter

Date	**:**	**29/05/2024**
Version	**:**	**1.0**
State	**:**	**Writing**
Author	**:**	**Saeed Ba Wazir**

Contents

OWASP top 10 2
Explanation and motivation 4
API1:2023 - Broken Object Level Authorization 4
API2:2023 - Broken Authentication 4
API3:2023 - Broken Object Property Level Authorization 4
API4:2023 - Unrestricted Resource Consumption 5
API5:2023 - Broken Function Level Authorization 5
API6:2023 - Unrestricted Access to Sensitive Business Flows 5
API7:2023 - Server Side Request Forgery 5
API8:2023 - Security Misconfiguration 6
API10:2023 - Unsafe Consumption of APIs 6
References 7

OWASP top 10

For this security analysis OWASP top 10 will be checked in Kwetter, OWASP Top 10 is a crucial resource for developers and security professionals to understand the most critical web application security risks.

	Likelihood	Impact	Risk	Actions possible	Planned
API1:2023 - Broken Object Level Authorization	Low	Severe	High	Fixed	Yes
API2:2023 - Broken Authentication	Low	Severe	Medium	1. Adding limit for login tries 2. Permitting strong passwords only	Yes
API3:2023 - Broken Object Property Level Authorization	Low	Severe	Medium	Fixed	Yes
API4:2023 - Unrestricted Resource Consumption	Low	Moderate	Medium	1. Adding limits for the properties inside the request	No, risk accepted due time limitation.
API5:2023 - Broken Function Level Authorization	Low	Severe	Medium	Fixed	Yes
API6:2023 - Unrestricted Access to Sensitive Business Flows	Low	Severe	Medium	1. Adding non-human patterns detection	No, risk accepted.
API7:2023 - Server Side Request Forgery	Medium	Moderate	Medium	1. Limit the types of image that user can upload when the user update their image and validate it.	No, risk accepted.

API8:2023 - Security Misconfiguration	High	Severe	High	1. Enable Https protocol 2. Implement CORS policy 3. Do not send exceptions trace back to the user 4. Ensure all of the servers in the traffic chain process incoming requests in a uniform manner	Yes, but now all of them, the first, the fourth will not be implement due to time constrains.
API9:2023 - Improper Inventory Management	Low	Low	Low	Fixed	Yes
API10:2023 - Unsafe Consumption of APIs	Low	Low	Low		

(OWASP Top 10 API Security Risks – 2023, n.d.)

Explanation and motivation

API1:2023 - Broken Object Level Authorization

BOLA vulnerabilities in APIs grant unauthorized data access through manipulated object IDs (numbers, fancy codes, text) in requests. Easy to spot in targets, headers, or payloads, these IDs become hackers' skeleton keys to unlock sensitive data.

The likelihood of security vulnerability is low since it is created due to the poor designing of the API endpoints, when the application does not check the user have access to the resources that is requesting.

The impact of this security vulnerability is severe, as it can result in unauthorized access to sensitive data, modification of system resources, and even complete system compromise. This can have significant consequences for an organization, including financial losses, damage to reputation, and legal liabilities. Therefore the Kwetter should always implement a strong object level authorization mechanisms.
(API1:2023 Broken Object Level Authorization, n.d.)

API2:2023 - Broken Authentication

Due to its inherent accessibility, the authentication mechanism presents a vulnerable entry point for attackers. While exploiting certain authentication weaknesses may necessitate advanced technical expertise, readily available tools can significantly lower the barrier to entry for malicious actors.

The likelihood for this low since Kwetter already validates the tokens and use a strong algorithms to generate the tokens, however the application does not provide brute force preventing mechanism, therefore the application is vulnerable to this kind of attacks and the risk is medium and the impact is severe, therefore there will be a mechanism to lock the account in case the user reach a number of wrong passwords attempts, and when creating a password to be a strong passwords only to decrease the likelihood of this type of attacks.
(API2:2023 Broken Authentication, n.d.)

API3:2023 - Broken Object Property Level Authorization

RESTful APIs often suffer from excessive data exposure, returning all object properties by default. While GraphQL requires more specific requests to expose unwanted data, these properties can still be identified. This process can be laborious, but some automated tools can help alleviate the burden.

Since Kwetter uses REPR or request endpoint response pattern to ensure that response doesn't expose sensitive data and when sending a request the attacker can not access properties that they do not have access to them, which make the likelihood of this kind of attacks low however the impact will be severe since this kind of attacks may cause significant consequences for an organization, including financial losses, damage to reputation, and legal liabilities, lastly the risk will be medium because in the process of Kwetter development, this kind of attacks is kept in mind and actions toward preventing this kind of attacks is already taken.
(API3:2023 Broken Object Property Level Authorization, n.d.)

API4:2023 - Unrestricted Resource Consumption

BOLA attacks are low-complexity as they leverage basic API requests. The grunt work can be automated, with a single machine or cloud resources firing off a barrage of requests, potentially overwhelming the API and causing a Denial-of-Service (DoS).

For this security vulnerability the likelihood is low since the application have rate limits for each endpoint to ensure that DOS attacks can be prevented, but the application still does not limit the properties inside the request themselves, which make the impact of this vulnerability moderate and the risk medium, lastly we can add limits for the properties but due to the time limitation we will keep it in the plan till there is time for it and because the application does not use any third party resource beside the infrastructure for the deployment.
(API4:2023 Unrestricted Resource Consumption, n.d.)

API5:2023 - Broken Function Level Authorization

Unauthorized access masquerading as legitimacy. Attackers leverage standard API calls, but target endpoints beyond their permitted access level. These exposed endpoints become easy pickings for unauthorized data retrieval.

Kwetter's risk is considered medium for this kind of attacks. While authentication and authorization for admin endpoints along with development focus on secure endpoints minimize the likelihood of attacks, an undetected vulnerability could be severely exploited. In other words, Kwetter's defenses make these kind of attacks unlikely but potentially impactful.

(API5:2023 Broken Function Level Authorization, n.d.)

API6:2023 - Unrestricted Access to Sensitive Business Flows

By understanding the app's core functionalities (business model), attackers can identify critical data flows and automate unauthorized access to them. This disrupts these sensitive operations, potentially causing significant harm to the business.

The likelihood is moderate for this type of attacks, due to the possibility of using more than machine at the same time to spam the application to create tweets without validating whether this is bot or human, however the application will limit the requests but this is not enough for this kind of attacks and this will lead to high usage of the resource of the application, lastly for this type of attacks there will be no actions taken in this version, however using non-human detections tools can be a good solution.
(API6:2023 Unrestricted Access to Sensitive Business Flows, n.d.)

API7:2023 - Server Side Request Forgery

Insecure API calls can backfire, SSRF vulnerabilities arise when an API blindly fetches resources from URLs provided by users. This lets attackers trick the application into sending requests to unintended destinations, bypassing firewalls and VPNs.

The likelihood of this attacks is medium since the application does not validate the image of the user when fetching the user image from the source or the uploaded image which make the risk level as moderate and the impact medium, however this can be solve with validating and allowing the user to provide links from certain origins like Google drive, but for this version we will not fix it due to time limitations.
(API7:2023 Server Side Request Forgery, n.d.)

API8:2023 - Security Misconfiguration

Hackers go for the easy wins. They target unpatched holes, common attack points, weak default settings, and exposed files. Since this info is often public with ready-made tools to exploit them, it makes hacking these systems way easier.

The likelihood of this type of cyber attacks is high, due to the current configurations, which will lead to severe impact and this is risky for the application therefore the configuration of CORs policy will be updated and the exceptions traces will not be return and when the application return an exception the application will return only that there is an exception without the trace. Additionally if there is more time the security of the traffic in the servers will be looked into and the https enable for all of the traffics.
(API8:2023 Security Misconfiguration, n.d.)

API9:2023 - Improper Inventory Management

Unauthorized access often stems from outdated APIs or unpatched endpoints. These vulnerabilities, coupled with readily available exploits or lax security practices, create easy entry points for threat actors. In some instances, data breaches can occur through unintended third-party access.

For the likelihood of this cyber attacks is low due to the fact this application already have different environments for the development as locally and staging using GitHub actions, and lastly production which is deployed on Azure and using GitHub secrets repository to pass the secrets. However this risk and the impact will be low, due to the fact that the application does not have multiple versions running in the same time.
(API9:2023 Improper Inventory Management, n.d.)

API10:2023 - Unsafe Consumption of APIs

Exploiting this vulnerability necessitates identifying and potentially compromising dependent APIs. However, this crucial information is often obscure, and these dependent APIs themselves might be inherently secure, significantly hindering attacker success.

The likelihood of this will be low due to the fact Kwetter does not use an external services, therefore the risk and the impact will be low as well, however acknowledging the fact that the developers tend to trust external service which needs to be considered in future integrations with external services in the future or the impact and the risk will be high.

(API10:2023 Unsafe Consumption of APIs, n.d.)

References

OWASP Top 10 API Security Risks – 2023. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0x11-t10/

API1:2023 Broken Object Level Authorization. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/

API2:2023 Broken Authentication. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/

API3:2023 Broken Object Property Level Authorization. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa3-broken-object-property-level-authorization/

API4:2023 Unrestricted Resource Consumption . Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa4-unrestricted-resource-consumption/

API5:2023 Broken Function Level Authorization. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/

API6:2023 Unrestricted Access to Sensitive Business Flows. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa6-unrestricted-access-to-sensitive-business-flows/

API7:2023 Server Side Request Forgery. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa7-server-side-request-forgery/

API8:2023 Security Misconfiguration. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/

API9:2023 Improper Inventory Management. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xa9-improper-inventory-management/

API10:2023 Unsafe Consumption of APIs. Retrieved from: https://owasp.org/API-Security/editions/2023/en/0xaa-unsafe-consumption-of-apis/