QuillAudits

Audit Report
December, 2023

For

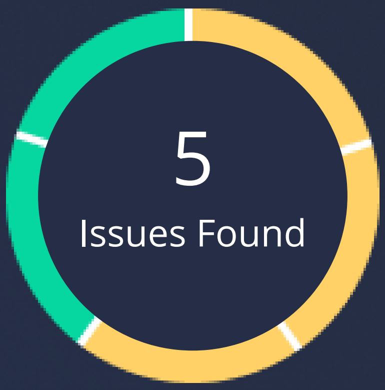
L

Table of Content

Executive Summary

Project Name	Locksonic
Overview	Locksonic is a platform to Connect with top NFT creators and trade unique X collections.
Timeline	1st December 2023 - 6th December 2023
Method	Manual Review, Automated Testing, Functional Testing, etc.
Audit Scope	The scope of this pentest was to analyze the **Web Application** for quality, security, and correctness. https://sec.golocksonic.xyz/
Review 2	7th December 2023 - 11th December 2023

Number of Issues per Severity

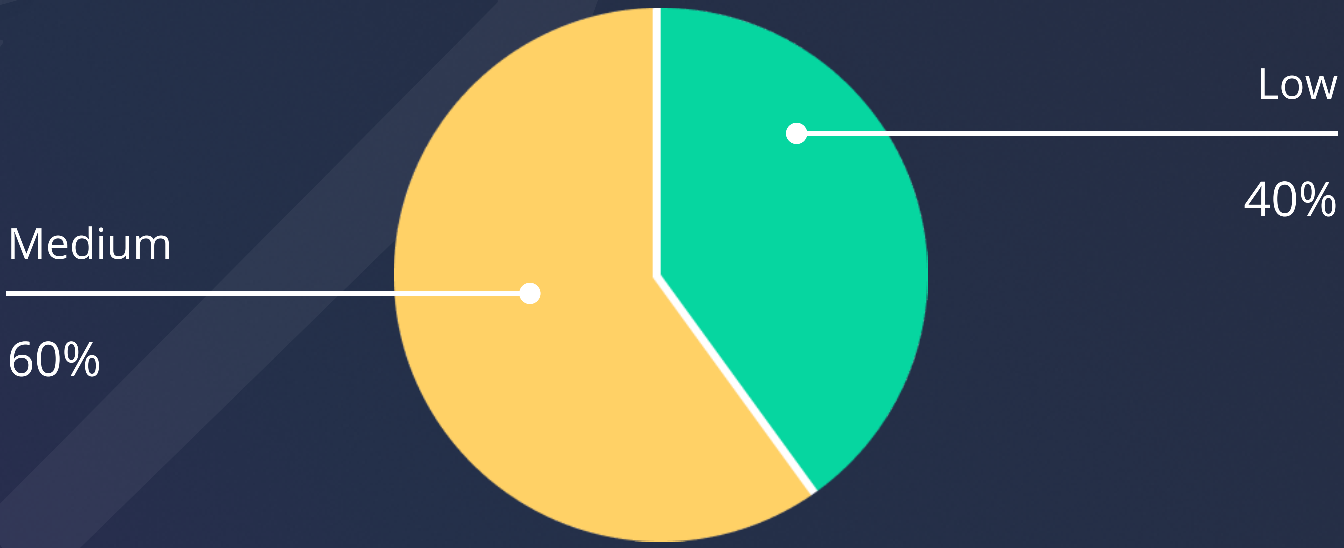


🟥 High 🟨 Medium

🟩 Low 🟪 Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	**2**	**1**	0
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	**1**	**1**	0

Security Chart



Checked Vulnerabilities

We scanned the application for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- ✓ Improper Authentication
- ✓ Improper Resource Usage
- ✓ Improper Authorization
- ✓ Insecure File Uploads
- ✓ Insecure Direct Object References
- ✓ Client-Side Validation Issues
- ✓ Rate Limit
- ✓ Input Validation
- ✓ Injection Attacks
- ✓ Cross-Site Request Forgery
- ✓ Broken Authentication and Session Management
- ✓ Insufficient Transport Layer Protection
- ✓ Broken Access Controls
- ✓ Insecure Cryptographic Storage
- ✓ Insufficient Cryptography
- ✓ Insufficient Session Expiration
- ✓ Information Leakage
- ✓ Third-Party Components
- ✓ Malware
- ✓ Denial of Service (DoS) Attacks
- ✓ Cross-Site Scripting (XSS)
- ✓ Security Misconfiguration
- ✓ Unvalidated Redirects and Forwards

And more...

Techniques and Methods

Throughout the pentest of Locksonic web applications, care was taken to ensure:

- Information gathering – Using OSINT tools information concerning the web architecture, information leakage, web service integration, and gathering other associated information related to web server & web services.
- Using Automated tools approach for Pentest like Nessus, Acunetix etc.
- Platform testing and configuration
- Error handling and data validation testing
- Encryption-related protection testing
- Client-side and business logic testing

Tools and Platforms used for Pentest:

- Burp Suite
- DNSenum
- Dirbuster
- SQLMap
- Acunetix
- Neucli
- Nabbu
- Turbo Intruder
- Nmap
- Metasploit
- Horusec
- Postman
- Netcat
- Nessus and many more

Issue Categories

Every issue in this report has been assigned with a severity level. There are four levels of severity, and each of them has been explained below.

High Severity Issues

A high severity issue or vulnerability means that your web app can be exploited. Issues on this level are critical to the web app's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the web app code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are four issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Issues Found

Medium Severity Issues

1. Image upload bypass

Description

The application has implemented a whitelist mechanism to restrict image uploads only from the domain media-uat.golocksonic.xyz. However, it was observed that the whitelist can be bypassed by manipulating the URL.

By changing the URL from media-uat.golocksonic.xyz to media-uat.golocksonic.xyz@shivang.github.io/asset/image/exp2.png, the application was found to fetch and display images from the external domain shivang.github.io. This indicates a failure in the whitelist implementation and poses a serious security risk as arbitrary images from unauthorized domains can be displayed.

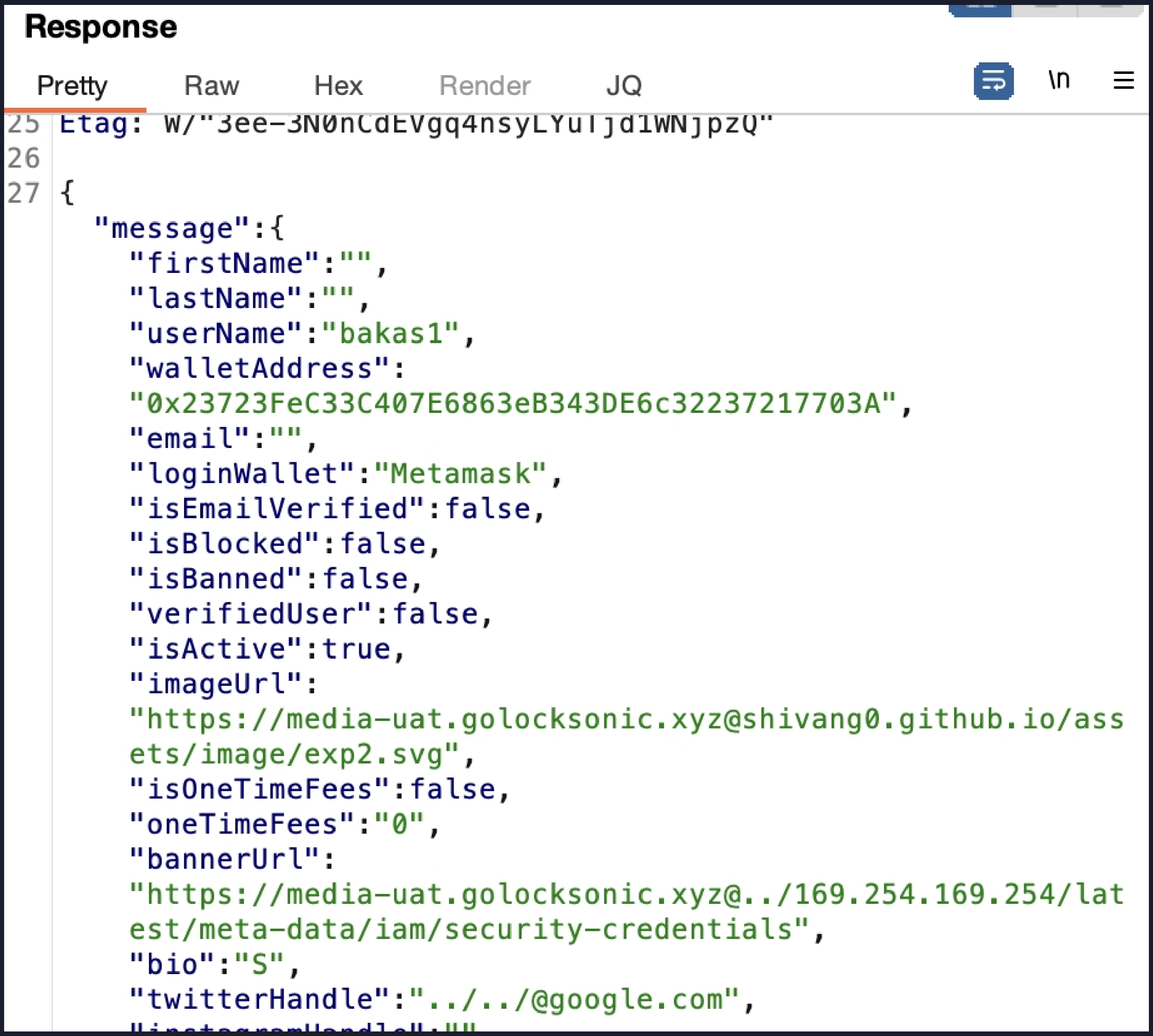
Steps to Reproduce

1. Upload an image in profile section and capture the request to https://api-uat.golocksonic.xyz/api/v1/user via PATCH method.
2. imageUrl and bannerUrl needs to be from https://media-uat.golocksonic.xyz or else the request fails as "message":"Malicious File".
3. You can bypass this by adding “@” after the domain. "imageUrl":"https://media-uat.golocksonic.xyz@shivang0.github.io/assets/image/exp2.svg”.
4. Similar can be also done for bannerUrl.

Recommendation

- **Strict Validation:** Implement strict validation mechanisms for image uploads to ensure that only files from trusted and authorized domains are accepted.
- **Server-Side Checking:** Conduct server-side validation to verify that the uploaded image URLs belong to the authorized domain (media-uat.golocksonic.xyz) rather than relying solely on client-side checks.

POC



```
Etag: W/"3ee-3N0nCdEVgq4nsyLYuIjd1WNjpzQ"

{
  "message":{
    "firstName":"",
    "lastName":"",
    "userName":"bakas1",
    "walletAddress":
    "0x23723FeC33C407E6863eB343DE6c32237217703A",
    "email":"",
    "loginWallet":"Metamask",
    "isEmailVerified":false,
    "isBlocked":false,
    "isBanned":false,
    "verifiedUser":false,
    "isActive":true,
    "imageUrl":
    "https://media-uat.golocksonic.xyz@shivang0.github.io/ass
    ets/image/exp2.svg",
    "isOneTimeFees":false,
    "oneTimeFees":"0",
    "bannerUrl":
    "https://media-uat.golocksonic.xyz@../169.254.169.254/lat
    est/meta-data/iam/security-credentials",
    "bio":"S",
    "twitterHandle":"../../@google.com",
```

Impact

- **Unauthorized Content Display:** An attacker can upload arbitrary images from external domains, leading to the unauthorized display of content on the application. This could result in the presentation of malicious, inappropriate, or misleading images to users.
- **Brand Reputation Damage:** Displaying content from unauthorized sources could harm the reputation of the application and the associated brand. Users may lose trust in the platform if they perceive it as vulnerable to manipulative practices.
- **Cross-Site Scripting (XSS) Attacks:** The ability to load content from external domains opens up the possibility of injecting malicious scripts into the application. This can lead to Cross-Site Scripting attacks, enabling the attacker to steal sensitive user information or perform actions on behalf of the user.

Status

Resolved

2. Host Header Injection

Description

Host Header Injection is a web vulnerability that occurs when an attacker can manipulate the Host header value in an HTTP request. In the context of the security assessment of sec.golocksonic.xyz, it implies that the application does not properly validate or sanitize the Host header, allowing an attacker to inject a malicious Host header.

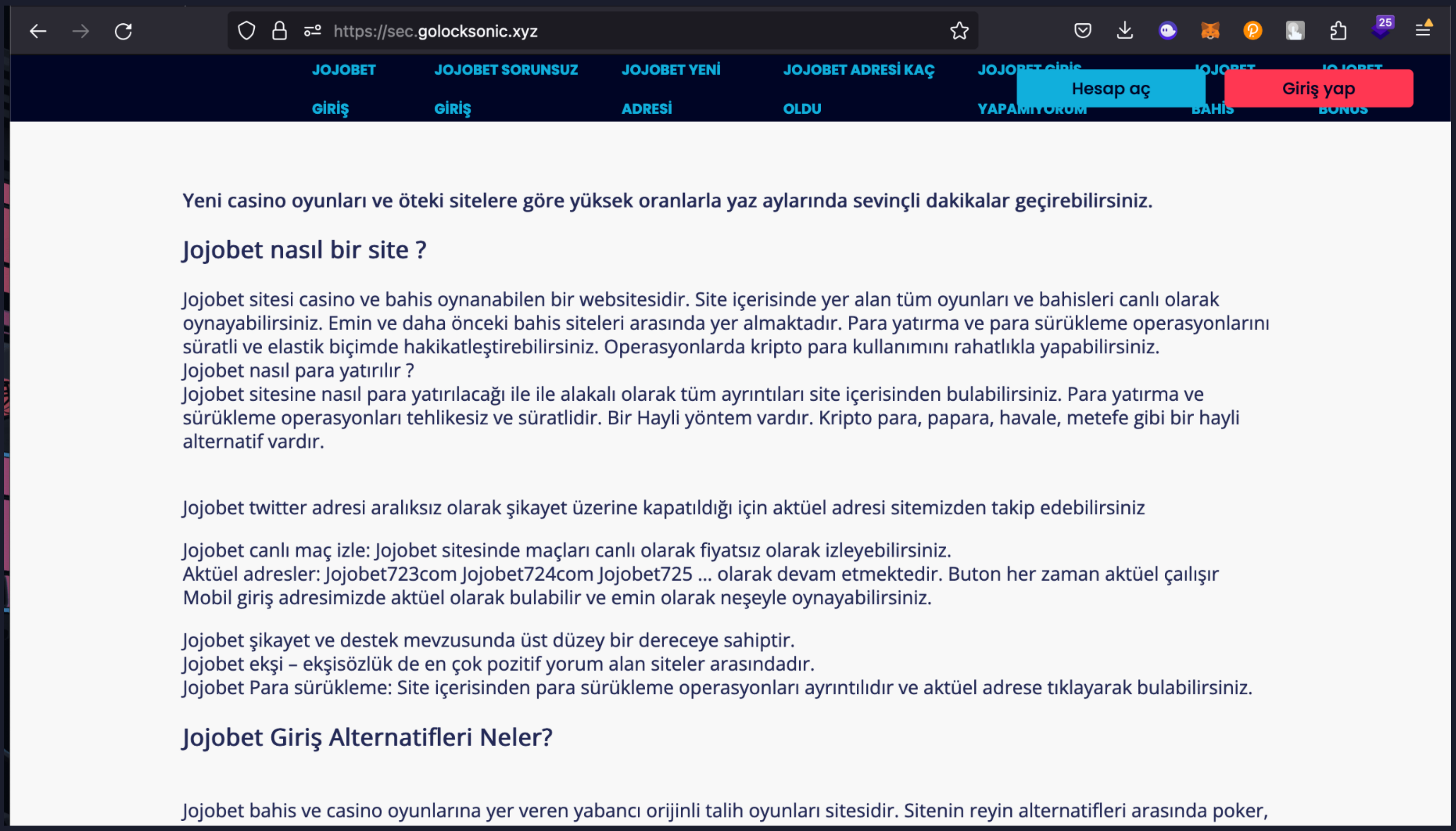
Steps to Reproduce

1. **Intercept the Request:** Use Burp Suite to intercept the HTTP request to sec.golocksonic.xyz through the Burp Proxy.
2. **Modify Host Header:** In Burp Repeater, manipulate the Host header value to a controlled or malicious domain. For example:
   Host: attacker-controlled-domain.com
3. **Send the Modified Request:** Forward the modified request with the tampered Host header to the server.
4. **Observe Application Behavior:** Analyze how the application processes the request with the manipulated Host header. Look for any unexpected behavior or responses.

Recommendation

- **Host Validation:** Implement strict validation and sanitation of the Host header on the server side. Only allow legitimate and expected hostnames.
- **Canonicalization:** Canonicalize the Host header to ensure consistency and prevent variations that might be exploited.
- **Security Headers:** Utilize security headers like HSTS (HTTP Strict Transport Security) to enhance the overall security posture of the application.
- **Logging and Monitoring:** Implement comprehensive logging of incoming requests, including Host headers, and establish monitoring mechanisms to detect and respond to any unusual or malicious activities.

POC



Yeni casino oyunları ve öteki sitelere göre yüksek oranlarla yaz aylarında sevinçli dakikalar geçirebilirsiniz.

Jojobet nasıl bir site ?

Jojobet sitesi casino ve bahis oynanabilen bir websitesidir. Site içerisinde yer alan tüm oyunları ve bahisleri canlı olarak oynayabilirsiniz. Emin ve daha önceki bahis siteleri arasında yer almaktadır. Para yatırma ve para sürükleme operasyonlarını süratli ve elastik biçimde hakikatleştirebilirsiniz. Operasyonlarda kripto para kullanımını rahatlıkla yapabilirsiniz.
Jojobet nasıl para yatırılır ?
Jojobet sitesine nasıl para yatırılacağı ile ile alakalı olarak tüm ayrıntıları site içerisinden bulabilirsiniz. Para yatırma ve sürükleme operasyonları tehlikesiz ve süratlidir. Bir Hayli yöntem vardır. Kripto para, papara, havale, metefe gibi bir hayli alternatif vardır.

Jojobet twitter adresi aralıksız olarak şikayet üzerine kapatıldığı için aktüel adresi sitemizden takip edebilirsiniz

Jojobet canlı maç izle: Jojobet sitesinde maçları canlı olarak fiyatsız olarak izleyebilirsiniz.
Aktüel adresler: Jojobet723com Jojobet724com Jojobet725 ... olarak devam etmektedir. Buton her zaman aktüel çalışır
Mobil giriş adresimizde aktüel olarak bulabilir ve emin olarak neşeyle oynayabilirsiniz.

Jojobet şikayet ve destek mevzusunda üst düzey bir dereceye sahiptir.
Jojobet ekşi – ekşisözlük de en çok pozitif yorum alan siteler arasındadır.
Jojobet Para sürükleme: Site içerisinden para sürükleme operasyonları ayrıntılıdır ve aktüel adrese tıklayarak bulabilirsiniz.

Jojobet Giriş Alternatifleri Neler?

Jojobet bahis ve casino oyunlarına yer veren yabancı orijinli talih oyunları sitesidir. Sitenin reyin alternatifleri arasında poker,

Impact

- **Domain Spoofing:** An attacker can make the application believe that the request is coming from a different domain. This can lead to domain spoofing attacks, causing confusion or misdirection of users.
- **Session Hijacking:** If the application relies on the Host header for session management, a manipulated Host header might lead to the session being associated with an attacker-controlled domain, facilitating session hijacking.
- **Cache Poisoning:** Content delivery networks (CDNs) and caching mechanisms often rely on the Host header for caching. Manipulating this header could result in cache poisoning, serving malicious content to users.

Status

Acknowledged

3. X-secret-token Leaked in JS File

Description

The X-secret-token leakage in the JavaScript file _app-f4c9dd6de52ee1bc.js on https://sec.golocksonic.xyz indicates that a sensitive token, presumably used for authentication or authorization, is exposed in the client-side code. The presence of such information in a JavaScript file could lead to potential security risks.

POC Link

https://sec.golocksonic.xyz/_next/static/chunks/pages/_app-f4c9dd6de52ee1bc.js

Impact

- **Unauthorized Access:** An attacker who gains access to the leaked X-secret-token could potentially impersonate a legitimate user or perform actions on behalf of the user, leading to unauthorized access.
- **Data Exposure:** If the token is used for API calls or other requests, an attacker could use the token to access and retrieve sensitive data from the server.
- **Security Misconfiguration:** The exposure of a secret token in client-side code suggests a security misconfiguration, highlighting potential weaknesses in the application's security implementation.

Remediation

- **Remove Tokens from Client-Side Code:** Sensitive tokens, especially those related to authentication and authorization, should never be present in client-side JavaScript files. Move such tokens to server-side code or implement more secure mechanisms.
- **Rotate Tokens:** If the leaked token is a long-lived token, consider rotating it to mitigate the risk of unauthorized access.

Status

Acknowledged

Low Severity Issues

1. Special Character Check

Description

The issue of not checking special characters in the username on the server-side, despite frontend validation, poses a security vulnerability. While the frontend validation provides a positive user experience by alerting the user about invalid characters, the lack of server-side validation allows an attacker to bypass this restriction.

Steps to Reproduce

1. Change username is Settings on https://sec.golocksonic.xyz/settings
2. Change username to badad'/f and it will show error that special characters are not allowed now change it to adminquill and pass it as save and intercept the request in Burp
3. Now Change the userName parameter in burp from adminquill to admin/@quill'1^& and forward the request
4. It will be changed to admin/@quill'1^

POC

https://sec.golocksonic.xyz/profile/0x23723FeC33C407E6863eB343DE6c32237217703A

Recommended Fix

- **Implement Strict Server-Side Validation:** Enforce strict server-side validation for usernames to ensure that only allowed characters are accepted. This should be independent of frontend validation.
- **Use Regular Expressions:** Employ regular expressions or similar mechanisms to define a whitelist of acceptable characters for usernames. This helps prevent injection attacks and ensures that only valid characters are processed.
- **Encode Special Characters:** If usernames are used in contexts where special characters may have a different meaning (e.g., in a URL or query string), ensure proper encoding to prevent injection vulnerabilities.
- **Input Sanitization:** Implement input sanitization routines to clean and filter user inputs, removing any potentially harmful characters.

Impact

- **Security Bypass:** Attackers can exploit this vulnerability to create usernames with special characters that may have been restricted on the front end. This could lead to security bypass scenarios.
- **Injection Attacks:** The absence of proper server-side validation may expose the system to injection attacks if special characters are used to manipulate queries or commands.
- **Username Spoofing:** Attackers might create usernames with special characters that resemble legitimate usernames, leading to confusion and potentially enabling social engineering attacks.

Status

Resolved

2. Subscribe Email List No Rate Limit

Description

The "Subscribe Email Newsletter no rate limit" vulnerability refers to a situation where the email newsletter subscription functionality lacks proper rate-limiting controls. Rate limiting is a crucial security mechanism that helps prevent abuse, such as spamming or brute-force attacks, by restricting the number of requests a user or an automated script can make within a specific time frame.

Steps to Reproduce

1. Visit https://sec.golocksonic.xyz/ and go to bottom of the page
2. Subscribe using any email address eg: johnhopper0123@yopmail.com
3. Intercept the request and send it to intruder and make it add emails from johnhopper0123+1@yopmail.com to johnhopper0123+500@yopmail.com
4. Start the attack and you will get 200 for every request with response of "message":"This user added on SendGrid."

Recommendation

1. **Implement Rate Limiting:** Introduce rate-limiting mechanisms to control the number of newsletter subscription requests from a single user or IP address within a specified time frame.
2. **Use CAPTCHA or Similar Challenges:** Incorporate CAPTCHA or other challenges as an additional layer of protection to distinguish between legitimate users and automated scripts.
3. **Monitoring and Logging:** Implement monitoring and logging functionalities to track suspicious patterns of subscription requests. Unusual activity should trigger alerts for further investigation.

Impact

1. **Spam Attacks:** Without rate limiting, attackers or automated scripts can abuse the newsletter subscription feature by making a large number of requests in a short period. This could result in spam flooding the system.
2. **Resource Exhaustion:** A high volume of newsletter subscription requests, especially if they are automated, can lead to resource exhaustion, affecting the availability and performance of the system.
3. **Abuse of Resources:** Attackers can exploit the lack of rate limiting to abuse system resources, potentially impacting the delivery of legitimate services.

Status

Acknowledged

Closing Summary

In this report, we have considered the security of the LockSonic Marketplace. We performed our audit according to the procedure described above.

Some issues of medium, low, and informational severity were found. Some suggestions and best practices were also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits Dapp Pentest security audit provides services to help identify and mitigate potential security risks in the LockSonic Platform. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of the LockSonic Platform. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of the LockSonic Team to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

850+	**$30B**	**$30B**
Audits Completed	Secured	Lines of Code Audited

Follow Our Journey

Audit Report
December, 2023

For

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com

audits@quillhash.com