QuillAudits

Audit Report
September, 2023

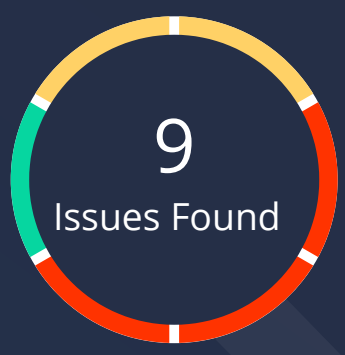
For

IBAX Crypto
Tokenizing Real Assets

Table of Content

Executive Summary

Project Name	IBAX
Overview	IBAX is an simple platform is simple and straightforward. You can acquire a shareholding interest in the Gas-LNG Project with a minimum contingent Gas Resource of 6 Trillion Cubic Feet (TCF). Your Token will increase in value as the project develops. Follow a few easy steps, create and fund your account.
Scope of Audit	The scope of this pentest was to analyze the Web App and API calls for quality, security, and correctness.
Timeline	8 August 2023 - 5 september 2023
Contracts in Scope	*https://stage-user.ibaxcrypto.io/* *https://stage-admin.ibaxcrypto.io/* *https://stage-api.ibaxcrypto.io*



High Medium Low Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	0
Partially Resolved Issues	0	0	0	0
Resolved Issues	4	3	2	0

Techniques and Methods

Throughout the pentest, 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, Wireshark, 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.

Types of Issues

Open
Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved
These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged
Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved
Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

Types of Severities

High

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

Medium

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

Low

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 severity 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.

Checked Vulnerabilities

- SQL/NoSQL Injection
- JNDI Injection
- RFD Attacks
- Insecure File Uploads
- Insecure Direct Object References
- WebSocket Hijacking
- H2 Hijacking
- HTTP Smuggling
- Request/Response Smuggling
- Cross-Site Scripting
- Cross-Site Request Forgery
- Cross-site WebSocket Hijacking
- ReDOS
- Dos/DDOS
- ClickJacking
- Tabnabbing
- Cookie Tossing
- XML external entity (XXE) injection
- Server-Side Request Forgery
- CRLF Injection
- Path Traversal
- Prototype Pollution
- Timing Attacks
- Dependency Confusion Attacks
- Third Party Issues
- Padding Oracle/ PRNG Attacks
- Subdomain Takeover
- Account Takeover
- Redirection Attacks
- HTTP Parameter Pollution
- Cache Poisoning
- And more.

Manual Testing

High Severity Issues

1. SQL Injection

Description

The stage-api.ibaxcrypto.io subdomain at maxprice parameter is vulnerable to SQL injection. This is because the parameter is not properly sanitized. This means that an attacker can inject malicious SQL code into the parameter, which will then be executed by the database.
The malicious SQL code could be used to steal sensitive information from the database, such as user passwords or credit card numbers.

Vulnerable Endpoint: https://stage-api.ibaxcrypto.io:443/api/v1/token/alltoken?limit=8&page=1&userId=112&paymentMode=&projectStatus=&maxPrice=413*&minPrice=0&salephaseType=&tokenCategory=

Steps to Reproduce

Use SQL Map to exploit using following command:

sqlmap.py -u "https://stage-api.ibaxcrypto.io:443/api/v1/token/alltoken?limit=8&page=1&userId=112&paymentMode=&projectStatus=&maxPrice=413&minPrice=0&salephaseType=&tokenCategory=" --user-agent="Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/115.0" --referer="https://stage-user.ibaxcrypto.io/" --headers="Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOjExMiwiZW1haWwiOiJDSVBIRVI5NDJAWU9QTUFJTC5DT00iLCJhY2NvdW50VHlwZSI6IlJldGFpbGVyIiwiaWF0IjoxNjkxNTYzMzI1LCJleHAiOjE2OTE1OTIxMjV9.qGk3wzjeX-KlCMe_mrXQJ-BZOvxo5dHjVy3htG6GKKM" --delay=0 --timeout=30 --retries=0 --level=4 --risk=3 --threads=8 --time-sec=5 --technique=BEUSTQ -b --current-db --hostname --is-dba --users --privileges --dbs --batch --answers="crack=N,dict=N,continue=Y,quit=N"*

POC

Recommendation

- To mitigate this vulnerability, the maxprice parameter should be properly sanitized. This can be done by using a library like db-sanitize.
- In addition, the database should be updated to use prepared statements. This will prevent attackers from injecting malicious SQL code into the database.

Status

Resolved

2. Account Login Bypass By Response Manipulation

Description

Account login bypass by response manipulation is a type of attack where an attacker can bypass the authentication process by manipulating the response from the server. This can be done by changing the response code, the response body, or the cookies that are sent back to the client. In the case of the ibax website, an attacker could exploit the account login bypass by response manipulation vulnerability to gain unauthorized access to any account. This is because the website does not properly validate the response from the server. The malicious request could be used to gain unauthorized access to any account

Vulnerable Component

1. https://stage-user.ibaxcrypto.io/login - User and Company Both
2. https://stage-admin.ibaxcrypto.io/

Steps to Reproduce

1. Capture the request while logging in Using Burp Suite
2. Click on View Response of the request option
3. Change the response from 400 to Following Response.

HTTP/2 200 OK
Date: Wed, 09 Aug 2023 13:15:18 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 955
*Access-Control-Allow-Origin: **
Content-Security-Policy: default-src 'self';base-uri 'self';font-src 'self' https: data:;form-action 'self';frame-ancestors 'self';img-src 'self' data:;object-src 'none';script-src 'self';script-src-attr 'none';style-src 'self' https: 'unsafe-inline';upgrade-insecure-requests
Cross-Origin-Embedder-Policy: require-corp
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
X-Dns-Prefetch-Control: off
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Download-Options: noopen
X-Content-Type-Options: nosniff
Origin-Agent-Cluster: ?1
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: no-referrer
X-Xss-Protection: 0
Ratelimit-Limit: 4000
Ratelimit-Remaining: 3981

Ratelimit-Reset: 50
Etag: W/"3bb-lg5tbJ1ruOKVP067HZfhqhPgIbY"

{"success":true,"data":
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOjExMiwiZW1haWwiOiJDSVBIRVI5NDJAWU9QTUFJTC5DT00iLCJhY2NvdW50VHlwZSI6IlJldGFpbGVyIiwiaWF0IjoxNjkxNTg2OTE4LCJleHAiOjE2OTE2MTU3MTh9.NWjVhAb3VdizJ_eTum_9a2QNc5YLB8JiZRO0q3iuCcc","user":
{"account_accountId":112,"account_email":"CIPHER942@YOPMAIL.COM","account_accountType":"Retailer","account_loginAttempt":0,"account_lockedTimeInMs":"0","account_isLocked":0,"account_isActive":1,"account_kycStatus":"Pending","account_kycMessage":null,"account_emailVerified":1,"account_isLoginFirstAttempt":1,"account_is2FAactive":0,"account_userProfile":"user/112-1691563368620-oZZLkqxYxrsm.png","account_createdAt":"2023-08-09T06:39:42.195Z","account_updateedAt":"2023-08-09T13:15:18.000Z","account_userName":"DEMOO1","retailer_retailerId":88,"retailer_firstName":"Quill","retailer_lastName":"demo","retailer_accountId":112},"emailVerification":false},"message":"Logged in successfully"}

4. Similar 200 Response of admin or Company can also be crafted
5. All you need is Email and a previous valid JWT of the user

Here this issue arises because JWT is not verified properly by the application even expired can be used it just verifies if a JWT is provided in response or not and not that it verifies it.

Recommendation

To mitigate the account login bypass by response manipulation vulnerability, the website should be updated to properly validate the response from the server. This can be done by using a library like jwt.io.

Status

Resolved

3. KYC Requirement Bypassed

Description

KYC Verification is needed to execute any type of action in the application. But all it verifies is the response and it's parameter and doesn't check for it int the backend in JWT or another method to cross verify and only relies on the response body for KYC verification status

Vulnerable Endpoint:
https://stage-api.ibaxcrypto.io/api/v1/user/auth/login
account_kycStatus - In response Body

Steps to Reproduce

1. Try to login into the application and capture the request in BurpSuite
2. Check for the account_kycStatus in the response body
3. It can be pending so change it to approved
4. Now when your login is successful you can easily buy, create project or perform any action with KYC requirements.

Recommendation

- Rely on multiple requests and always believe on server side data and not on client-side response for such critical actions.
- Verify status via JWT or some cookie based method for each request to tampering can be avoided.

Status

Resolved

4. Secret Keys Leaked

Description

During the security assessment of the web application at https://stage-user.ibaxcrypto.io/, critical vulnerabilities were identified. The assessment revealed the leakage of sensitive information including Google Secret, Client ID, and JWT Secret in the source code of the application. This can potentially lead to unauthorized access, data breaches, and identity theft.

Vulnerable Endpoint:

- Link 1
- Link 2

Impact

The leakage of sensitive information exposes the application and its users to various security risks:

- Unauthorized Access: Attackers who gain access to the leaked secrets could impersonate legitimate users, perform unauthorized actions, or access sensitive user data.
- Data Breaches: The leaked secrets could be exploited to access the application's database and extract sensitive user information, which may include personally identifiable information (PII) and financial data.
- Identity Theft: Attackers could use the stolen secrets to generate valid tokens or sessions, allowing them to impersonate legitimate users and perform malicious actions on their behalf.
- Financial Loss: Unauthorized access to user accounts could result in financial losses, both for users and the application owner.

Recommendation

- Remove Hard-Coded Secrets: Remove all hard-coded secrets, such as Google Secret, Client ID, and JWT Secret, from the source code and configuration files. Store these secrets in a secure and centralized location, such as environment variables or a secure credentials management solution.
- Implement Secrets Management: Utilize a secure secrets management solution to handle sensitive information. This ensures that secrets are securely stored, managed, and accessed only by authorized components of the application.
- Implement Least Privilege: Apply the principle of least privilege to access tokens and permissions. Ensure that tokens and secrets are granted the minimum required permissions to perform their intended tasks.

Status

Resolved

Medium Severity Issues

5. KYC Verification Spoofing

Description

KYC spoofing is a type of attack where an attacker uses a fake identity to gain access to a system or service. This can be done by using a fake passport or another identification document to create a fake account.
The KYC verification requested by the application can be spoofed easily as it dosen't verify if the passport is currently of the user or not and not checks the face properly.
I was able to bypass the passport verification by an image from pintrest and cropping the photo from the same fake passport and using it for face verification.
This can cause issues in the long run in case of any incident and you may later realise that the KYC of user was fake.

POC

Account of DEMOO1

Recommendation

To mitigate the risk of KYC spoofing, the organization should implement additional security measures to verify the identity of users. This could include:

- Conducting liveness checks during face verification. This would involve asking the user to perform a series of actions, such as blinking or moving their head, to confirm that they are a real person.
- Using multi-factor authentication (MFA). This would involve requiring users to provide two or more pieces of identification, such as a password and a code from their phone, to authenticate themselves.
- Implementing a risk-based approach to KYC. This would involve assessing the risk of each user and implementing more stringent verification measures for users who are considered to be high-risk.
- Change the KYC Verification provider to a more reliant and secure KYC Verification service.

Status

Resolved

6. Temp Mail Usage allowed

Description

During our security assessment of the https://stage-user.ibaxcrypto.io/ platform, we identified a critical vulnerability in the registration process. The application currently allows users to register using disposable/temporary email services like yopmail. This poses a significant security risk as it enables potential malicious actors to create multiple accounts without a valid, traceable email address. Such accounts can be used to engage in fraudulent activities, potentially compromising the integrity of the platform and exposing legitimate users to financial risks.

Vulnerable Endpoint: *https://stage-user.ibaxcrypto.io/sign-up*

Impact

The registration vulnerability exposes several risks and potential consequences:

- Fraudulent Activities: Malicious users can create multiple fake accounts, potentially engaging in fraudulent investment activities and scams.
- Abuse of Platform: The ability to use disposable emails makes it difficult to track and block abusive users who may exploit the platform's services for unauthorized purposes.
- Financial Loss: Legitimate users could be exposed to financial loss if they unknowingly invest in schemes facilitated by malicious users.
- Reputation Damage: Successful exploitation of this vulnerability could lead to a loss of trust from users, damaging the platform's reputation and credibility.
- Regulatory Compliance: Depending on the nature of the platform, inadequate user verification could lead to non-compliance with financial regulations and industry standards.

Recommendation

- Block Disposable Email Services: Implement a mechanism to block disposable/temporary email domains, such as yopmail, during the registration process. Utilize an updated and comprehensive list of such services to prevent their usage.
- Strong Authentication: Implement multi-factor authentication (MFA) for user accounts. This adds an extra layer of security, making it significantly harder for malicious actors to gain unauthorized access.
- User Monitoring: Implement monitoring mechanisms to track user activities for suspicious behavior. Apply heuristics and anomaly detection to identify patterns that might indicate fraudulent activities.

Status

Resolved

7. Weak Account ID values

Description

During the security assessment of the web application at https://stage-user.ibaxcrypto.io/, it was observed that the application assigns three-digit account IDs to each user account. This is an insecure practice that can potentially lead to unauthorized access and enumeration attacks.

Vulnerable Endpoint: *https://stage-api.ibaxcrypto.io/api/v1/user/auth/getkycdata?type=Retailer&limit=10&page=1&s=*

Impact

The insecure assignment of three-digit account IDs exposes several security risks:

- Predictable Account Enumeration: Attackers can easily iterate through the range of three-digit numbers to enumerate valid user account IDs. This can facilitate brute-force attacks and unauthorized access to user accounts.
- Reduced Entropy: The limited range of three-digit numbers significantly reduces the entropy of account IDs, making them more susceptible to guessing or enumeration attacks.
- Information Leakage: The predictable account IDs can inadvertently leak information about the number of users registered on the platform, which could be exploited by attackers for social engineering or reconnaissance purposes.

Recommendation

- Implement Stronger Account ID Generation
- Implement Server-Side Authentication and Authorization
- User Enumeration Mitigation

Status

Resolved

Low Severity Issues

8. Multiple Deprecated Libraries in package-lock.json

Description

package-lock Stores files that can be useful for the dependency of the application. This is used for locking the dependency with the installed version. It will install the exact latest version of that package in your application and save it in package. This arises a problem if the dependency used has an exploit in the version mentioned. It can create a backdoor for an attacker.

Vulnerable Dependencies:
dottie
semver
word-wrap

Recommended Fix

1. Update all the above mentioned Dependencies
2. Remove any Library Not needed.

Impact

Multiple of these libraries have public exploits and CVE-registered issues that have been patched and can help your application stay more secure from any dependency-vulnerable issues.

Status

Resolved

9. Clickjacking

Description

A Clickjacking vulnerability was identified on the web application. Clickjacking, also known as a UI redress attack, is a technique that tricks users into clicking on malicious content or performing unintended actions without their knowledge or consent. In this case, the vulnerability allows an attacker to overlay or embed malicious content on top of the legitimate ibax website, potentially leading to various forms of abuse or exploitation.

Vulnerable Endpoint:

https://stage-user.ibaxcrypto.io

Steps to Reproduce

1. Create a malicious web page or use an existing website under your control.
2. Modify the malicious web page's HTML to include an iframe that loads

```
<html>
  <body>
    <h1>Malicious Website</h1>
    <iframe src="https://stage-user.ibaxcrypto.io"></iframe>
  </body>
</html>
```

3. Host the malicious web page on a web server.
4. Open the link where the malicious web page is hosted in your browser. You will find your website embedded in an iframe.

Recommendation

1. Set the X-Frame-Options HTTP response header to deny or sameorigin. This will prevent the website from being loaded inside an iframe on malicious websites.
2. Implement a strong Content Security Policy that includes the frame-ancestors directive with 'self' or specific trusted domains to restrict which websites can embed Google's content.

Status

Resolved

Closing Summary

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

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

Disclaimer

QuillAudits Dapp security audit provides services to help identify and mitigate potential security risks in iBax. 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 iBax. 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 iBax 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+
Audits Completed

$30B
Secured

800K
Lines of Code Audited

Follow Our Journey

Audit Report
September, 2023

For

IBAX Crypto
Tokenizing Real Assets

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com

audits@quillhash.com