QuillAudits

AUDIT REPORT

August 2025

For

alkimi

Table of Content

Executive Summary 03

Number of Security Issues per Severity 04

Summary of Issues 05

Checked Vulnerabilities 06

Techniques and Methods 07

Types of Severity 08

Types of Issues 09

Severity Matrix 10

Medium Severity Issues 11

1. Response Manipulation Enables Daily–Limit Bypass 11

2. Asymmetric Wallet-Linking Logic 13

3. Exposure of PII in Secondary JWT & Redundant Session Token 14

Closing Summary & Disclaimer 15

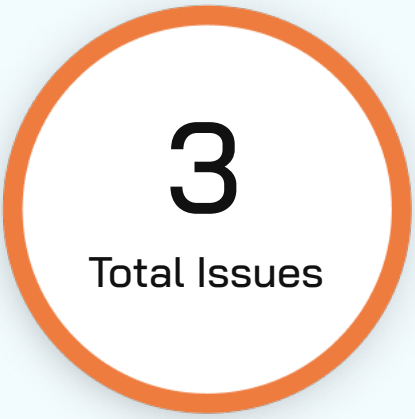
Executive Summary

Project Name	Alkimi
Protocol Type	Dapp Pentest
Project URL	https://penetration-testing.labs.alkimi.org/
Overview	Alkimi streamlines your operations, providing access to comprehensive data on ad tracking, placement, revenue, fees, and impressions, all within a unified platform. Alkimi aim to enhance the economic viability of quality advertising. The main scope of this pentest was the transfer and claim function of $ADS token from eth wallet to SUI wallet
Review 1	1st Aug 2025 - 7th Aug 2025
Updated Code Received	8th August 2025
Review 2	8th August - 14th August 2025

Verify the Authenticity of Report on QuillAudits Leaderboard:

https://www.quillaudits.com/leaderboard

Number of Issues per Severity



Critical	0 (0.0%)
High	0 (0.0%)
Medium	3 (100.0%)
Low	0 (0.0%)
Informational	0 (0.0%)

Severity

Issues	Critical	High	Medium	Low	Informational
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Partially Resolved	0	0	0	0	0
Resolved	0	0	**3**	0	0

Summary of Issues

Issue No.	Issue Title	Severity	Status
1	Response Manipulation Enables Daily–Limit Bypass	Medium	Resolved
2	Asymmetric Wallet-Linking Logic	Medium	Resolved
3	Exposure of PII in Secondary JWT & Redundant Session Token	Medium	Resolved

Checked Vulnerabilities

- 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 Scripting (XSS)
- Cross-Site Request Forgery
- Security Misconfiguration
- Broken Access Controls
- Insecure Cryptographic Storage
- Insufficient Cryptography
- Insufficient Session Expiration
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards
- Information Leakage
- Broken Authentication and Session Management
- Denial of Service (DoS) Attacks
- Malware
- Third-Party Components

And More..

Techniques and Methods

Throughout the pentest of application, 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	Acunetix	Nmap
DNSenum	Neucli	Metasploit
Dirbuster	Nabbu	Horusec
SQLMap	Turbo Intruder	Postman
Netcat	Nessus	And Many more..

Types of Severity

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

Critical: Immediate and Catastrophic Impact

Critical issues are the ones that an attacker could exploit with relative ease, potentially leading to an immediate and complete loss of user funds, a total takeover of the protocol's functionality, or other catastrophic failures. Critical vulnerabilities are non-negotiable; they absolutely must be fixed.

High (H): Significant Risk of Major Loss or Compromise

High-severity issues represent serious weaknesses that could result in significant financial losses for users, major malfunctions within the protocol, or substantial compromise of its intended operations. While exploiting these vulnerabilities might require specific conditions to be met or a moderate level of technical skill, the potential damage is considerable. These findings are critical and should be addressed and resolved thoroughly before the contract is put into the Mainnet.

Medium (M): Potential for Moderate Harm Under Specific Circumstances

Medium-severity bugs are loopholes in the protocol that could lead to moderate financial losses or partial disruptions of the protocol's intended behavior. However, exploiting these vulnerabilities typically requires more specific and less common conditions to occur, and the overall impact is generally lower compared to high or critical issues. While not as immediately threatening, it's still highly recommended to address these findings to enhance the contract's robustness and prevent potential problems down the line.

Low (L): Minor Imperfections with Limited Repercussions

Low-severity issues are essentially minor imperfections in the smart contract that have a limited impact on user funds or the core functionality of the protocol. Exploiting these would usually require very specific and unlikely scenarios and would yield minimal gain for an attacker. While these findings don't pose an immediate threat, addressing them when feasible can contribute to a more polished and well-maintained codebase.

Informational (I): Opportunities for Improvement, Not Immediate Risks

Informational findings aren't security vulnerabilities in the traditional sense. Instead, they highlight areas related to the clarity and efficiency of the code, gas optimization, the quality of documentation, or adherence to best development practices. These findings don't represent any immediate risk to the security or functionality of the contract but offer valuable insights for improving its overall quality and maintainability. Addressing these is optional but often beneficial for long-term health and clarity.

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.

Severity Matrix

Impact

Likelihood	High	Medium	Low
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

Impact

- **High** - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- **Medium** - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- **Low** - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

Asymmetric Wallet-Linking Logic

Resolved

Description

The application enforces an asymmetric rule: it allows multiple Ethereum addresses to be bound to a single Sui address, but prevents multiple Sui addresses from binding to the same Ethereum address. This inconsistency can be abused to escalate privileges or confuse reconciliation logic—an attacker who controls one Sui wallet can link dozens of ETH addresses under it, but a legitimate user holding multiple Sui wallets cannot consolidate them under their single ETH account.

Impact

Denial of Service for Legitimate Users: An attacker who controls a SUI wallet can pre-emptively "reserve" an ETH wallet, preventing its rightful owner from ever linking it.

Account Hijacking Risk: By linking a shared or public ETH address first, an attacker blocks others from accessing cross-chain features they legitimately deserve.

Poor UX & Support Burden: Innocent users get confusing "already linked" errors with no recourse.

Recommendation

Enforce Symmetric Policy: Decide on a clear linking model—either one-to-one (each wallet only links to a single cross-chain partner) or many-to-many—and implement it consistently in both directions.

Allow Release & Re-Linking: If unlinking is permitted, ensure the server fully clears prior associations so a fresh link can succeed.

POC

User A (SUI=S1) logs in and issues a link-wallet request binding ETH=E1 → SUI=S1. User B (SUI=S2) attempts the same binding (ETH=E1 → SUI=S2) and receives an error. Even after User A unlinks E1 from S1 via the client, the server continues to reject any subsequent E1→S2 link attempts—demonstrating that the block is permanent.

Medium Severity Issues

Response Manipulation Enables Daily—Limit Bypass

Resolved

Description

The /transfer-&-claim endpoint relies on client-side logic to enforce a per-day transfer cap. An attacker who intercepts the HTTP flow (e.g. in Burp Suite) can modify both the outgoing JSON payload and the incoming response so that the client believes its "daily total" is still under the limit allowing arbitrarily large transfers to be submitted and accepted.

Vulnerable URL

POST https://penetration-testing.labs.alkimi.org/transfer-&-claim

Impact

Unauthenticated Over-Limit Transfers: Attackers can send transfers well beyond the intended daily quota, draining funds or credit from user accounts.

Financial Loss & Fraud: Bypassed limits allow rapid unauthorized fund extraction.

Trust Erosion: Users and partners lose confidence when limits can be subverted.

Recommendation

Enforce Limits Server-Side: All daily-limit checks must occur on the server using immutable, authoritative counters (e.g. in the database) rather than relying on client-supplied or echoed values.

Remove Sensitive Logic from Client: Do not include any "remaining daily limit" in responses that the client uses to gate further actions. If client-side hints are desired, fetch them from a server-validated source on each action.

Replay & Tampering Protections: Sign or HMAC the response payloads including limit-related fields so that any tampering invalidates the signature and the server rejects the request.

POC

```
POST /transfer-&-claim HTTP/2
Host: penetration-testing.labs.alkimi.org
Cookie: _vercel_jwt=…; labs-primary-wallet-session=…; labs-wallet=0x13E8…; …
Content-Type: text/plain;charset=UTF-8
Next-Action: 9769f2b24d02524c714d1760bc6935c1975e73c7
…other headers…

[{"amount":"187000000000000000000",
  "ethWallet":"0x13E889cB6472C1FFE1782A7B0CbeF81DfF839828",
  "suiWallet":"
0xbe8cd9fefb5c91f2f234ab4e6175e7c14be00d780886d1cd8fb47a4373aac483",
  "nonce":4}]

Response:
     HTTP/2 200 OK
…standard security headers…

0:["$@1",["nGpCcxLkq6ziK_JgrHxmq",null]]
1:{"success":true,"error":null,"data":{
     "amountInWEI":"187000000000000000000",
     "ethWallet":"0x13e889cb6472c1ffe1782a7b0cbef81dff839828",
     "suiWallet":"
0xbe8cd9fefb5c91f2f234ab4e6175e7c14be00d780886d1cd8fb47a4373aac483",
     "nonce":4,
     "deadline":1754473176,
     "signature":"0xd24b…5a95d1b"
}}
```

Exposure of PII in Secondary JWT & Redundant Session Token

Resolved

Description

The labs-primary-wallet-session cookie carries a JSON Web Token (JWT) that embeds sensitive user data email address, full name, user ID and roles on every request. Moreover, this secondary session cookie is not required for authorization: even if it's removed, the request still succeeds (200 OK) as long as the primary _vercel_jwt is present. Embedding PII directly in a long-lived, client-stored JWT increases risk of leakage, and maintaining two overlapping session tokens is needless complexity.

Impact

User Privacy Violation: Exposes user's email and name on every request, which can be intercepted on insecure networks or in browser logs.

Data Leakage Risk: Any third-party script or extension with access to cookies can harvest PII.

Attack Surface Bloat: Maintaining two session tokens increases chance of misconfiguration or token-replay attacks.

Recommendation

Strip PII from JWT Payload: Issue opaque session identifiers (random reference tokens) or minimal claims in the client-side JWT. Store all user attributes server-side and look them up by reference.

Consolidate to Single Token: Use only the primary authorization token (_vercel_jwt). Eliminate labs-primary-wallet-session entirely or convert it to a server-signed, HttpOnly, Secure cookie with no PII.

POC

Decode the PII-JWT in any JWT debugger. Observe clear-text "email":"shivang.tester@gmail.com", "fullName":"Shivang", etc.
Remove the entire labs-primary-wallet-session cookie from the browser's request.
Replay the request; the server still returns HTTP 200 OK and processes the action successfully under the same user context (due to _vercel_jwt alone).

Closing Summary

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

Some issues of medium severity were found. Alkimi team resolved all the issues mentioned.

Disclaimer

QuillAudits Dapp audit is not a security warranty, investment advice, or an endorsement of the Alkimi. This audit does not provide a security or correctness guarantee of the audited smart contracts.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multi-step process. One audit cannot be considered enough. We recommend that the Alkimi Team put in place a bug bounty program to encourage further analysis of the source code by other third parties.

About QuillAudits

QuillAudits is a leading name in Web3 security, offering top-notch solutions to safeguard projects across DeFi, GameFi, NFT gaming, and all blockchain layers. With seven years of expertise, we've secured over 1400 projects globally, averting over $3 billion in losses. Our specialists rigorously audit smart contracts and ensure DApp safety on major platforms like Ethereum, BSC, Arbitrum, Algorand, Tron, Polygon, Polkadot, Fantom, NEAR, Solana, and others, guaranteeing your project's security with cutting-edge practices.

7+ Years of Expertise	**1M+** Lines of Code Audited
50+ Chains Supported	**1400+** Projects Secured

Follow Our Journey

AUDIT REPORT

August 2025

For

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillaudits.com