QuillAudits

Audit Report
February, 2023

For

DiveWallet

Table of Content

Executive Summary

Project Name The DiveWallet

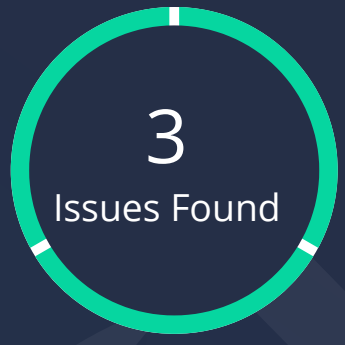
Overview The DiveWallet is a blockchain project that is making a blockchain wallet for users.

Timeline 21 December,2023 - 10 February,2023

Scope of Audit The scope of this pentest was to analyze the source code AND corresponding wallet for quality, security, and correctness.

In Scope:
Source-Code
https://bitbucket.org/flutterdevcodiste/divewalletapp/src/milestone-3/
Branch: milestone-3

Fixed In: *https://bitbucket.org/flutterdevcodiste/divewalletapp/src/new-design/*
Branch: new-design



High Medium

Low Informational

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

Techniques and Methods

Throughout the pentest of DiveWallet, 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
- Frida
- 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

- Improper Platform Usage
- Insecure Data Storage
- Insecure Communication
- Insecure Authentication
- Insufficient Cryptography
- Insecure Authorization
- Client Code Quality
- Code Tampering
- Reverse Engineering
- Extraneous Functionality
- Indirect Object ReferenceFunctionality Abuse
- Business Logic
- Wallet Seed Phrase Protection
- Previous attack exploits
- Transfer of tokens from and in application
- Rate Limit Issues and Bruteforcing

Manual Testing

High Severity Issues

No issues found

Medium Severity Issues

No issues found

Low Severity Issues

1. Infura RPC Key Leaked

Infura provides access to the Ethereum JSON-RPC API method library that interacts with the Ethereum blockchain. Methods include functionality for reading and writing data to the network and executing smart contracts. The RPC has a limit for free usage and the key leaked can be used by any unauthorized attacker to use the company's RPC link for personal usage.

Vulnerable File Location: /lib/src/core/utils/encryptions.dart

Impact

You would have to upgrade to Premium plans and that will increase the company's cost substantially per month.

Recommendation

Authenticate all requests to Infura with a project secret (API key secret).
Authenticate all requests to Infura with a JSON web token.
Set rate limiting [https://docs.infura.io/infura/networks/ethereum/how-to/secure-a-project/set-rate-limits].
Use allowlists [https://docs.infura.io/infura/networks/ethereum/how-to/secure-a-project/use-an-allowlist].

Status

Resolved

2. Multiple API-Keys Disclosed

These keys are from EVM-Chain explorers to receive data from the blockchain. Such keys are usually of free tier and have limit. After surpassing the limit the keys would generate rate limit error and would be useless for some time and this could be harmful for the application.

Vulnerable File Location: /lib/src/core/utils/app_strings.dart

Impact

You would have to upgrade to Premium plans and that will increase the company's cost substantially per month.

Recommendation

Encrypt such keys or store it on the server side and not on the client-side to avoid such issues

Status

Resolved

3. Using Math.Random

The JavaScript Math.random() function is designed to return a floating point value between 0 and 1. It is widely known (or at least should be) that the output is not cryptographically secure. When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information.
As the Math.random() function relies on a weak pseudorandom number generator, this function should not be used for security-critical applications or for protecting sensitive data. In such a context, a cryptographically strong pseudorandom number generator (CSPRNG) should be used instead.

Vulnerable File Location: assets/ethers.min.js, assets/trustwallet.min.js

Recommendation

Use strong generation methods as Crypto.getRandomValues() when required.

Reference

1. Stakoverflow

Status

Acknowledged

Informational Issues

No issues found

Closing Summary

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

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

Disclaimer

QuillAudits Dapp audit is not a security warranty, investment advice, or an endorsement of the DiveWallet Platform. 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 DiveWallet Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

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.

700+
Audits Completed

$16B
Secured

700K
Lines of Code Audited

Follow Our Journey

Audit Report
February, 2023

For

DiveWallet

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com