Audit Report

Dock Wallet Application

December 1, 2021

Version 1.0

Table of Contents

License

Disclaimer

This audit has been performed by

Oak Security

https://oaksecurity.io/
info@oaksecurity.io

Introduction

Purpose of this Report

Oak Security has been engaged by the Dock Association to perform a security audit of the Dock mobile wallet application.

The objectives of the audit are as follows:

1. Determine the correct functioning of the protocol, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine smart contract bugs, which might lead to unexpected behaviour.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebase Submitted for the Audit

The audit has been performed on the following GitHub repository:

https://github.com/docknetwork/dock-app

Commit hash: `74afaf5f240f4d8302d1a3e0f3e01c243933e9f3`

The codebase relies heavily on the Dock React Native SDK (https://github.com/docknetwork/react-native-sdk). **This additional codebase has been analyzed for functionality but has NOT been fully reviewed for security as part of the scope for this audit.**

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the code base's intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under-/overflow issues
    c. Key management vulnerabilities
4. Report preparation

Functionality Overview

The codebase implements a React Native application that acts as a wallet for the Dock blockchain.

How to read this Report

This report classifies the issues found into the following severity categories:

Severity	Description
Critical	A serious and exploitable vulnerability that can lead to loss of funds, unrecoverable locked funds, or catastrophic denial of service.
Major	A vulnerability or bug that can affect the correct functioning of the system, lead to incorrect states or denial of service.
Minor	A violation of common best practices or incorrect usage of primitives, which may not currently have a major impact on security, but may do so in the future or introduce inefficiencies.
Informational	Comments and recommendations of design decisions or potential optimizations, that are not relevant to security. Their application may improve aspects, such as user experience or readability, but is not strictly necessary. This category may also include opinionated recommendations that the project team might not share.

The status of an issue can be one of the following: **Pending,** **Acknowledged** or **Resolved**. Informational notes do not have a status, since we consider them optional recommendations.

Note, that audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of the system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**. We include a table with these criteria below.

Note, that high complexity or low test coverage does not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than a security audit and vice versa.

Summary of Findings

No	Description	Severity	Status
1	Use of dependencies with know security vulnerabilities	Major	Resolved
2	Logging of secrets	Major	Resolved
3	Passwords are not required to include any special characters	Minor	Resolved
4	Secrets could be leaked by bugtracking tools	Minor	Resolved
5	Cleartext rpc communication could be intercepted	Minor	Resolved
6	Short delay for passcode entry could enable brute force attack	Minor	Resolved
7	Missing error check-in account deletion	Minor	Resolved
8	Use of cryptographically unsafe random number generator	Informational	-
9	Hardcoded test account address	Informational	-
10	Unused imports	Informational	-
11	Leftover TODO comments	Informational	-
12	Unimplemented Feature	Informational	-
13	Leftover debug functionality	Informational	-

Code Quality Criteria

Criteria	Status	Comment
Code complexity	Medium	-
Code readability and clarity	High	-
Level of Documentation	Medium-low	Architectural documentation is sparse and GitHub README files are kept to a minimum. However, the code is well-commented.
Test Coverage	Medium	-

Architectural Observations

The Dock wallet application is implemented as a hierarchical deterministic wallet with the following features:

- Compliance with BIP-32 standard (https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki)
- Seeds are generated from 12 mnemonic words
- Account management for Dock users with the possibility to interact with the Dock token
- Biometric authentication is offered as an alternative to passcode authentication to the user depending on device support
- The user backup wallet is encrypted with an 8-12 character alphanumeric password

The Polkadot.js library is used to perform account related cryptographic operations. This is encapsulated in the Dock SDK. Hence, the wallet follows industry-standard account management.

However, since Polkadot.js only runs in a browser right now, a browser is simulated through a WebView environment. This is interconnected with the rest of the system through an RPC link.

Data is stored in three different data stores:

- Local device key store for user passcode
- Local storage for non-sensitive config data and encrypted wallet internals (keys)
- Unencrypted database storage for caching and nonsensitive data items

Detailed Findings

1. Use of dependencies with know security vulnerabilities

Severity: Major

An analysis of the dependency tree shows 117 known vulnerabilities in the dependency tree. Of these vulnerabilities, 2 are considered of critical severity and 35 are considered of high severity.

Recommendation

We recommend upgrading the reported dependencies to the newest version or replacing them with secure alternatives. To facilitate this, Appendix A provides a listing of all dependencies with known security vulnerabilities and their criticality. It is also recommended to lock dependencies to avoid versioning conflicts or new vulnerabilities being introduced through unexpected changes.

Status: Resolved

2. Logging of secrets

Severity: Major

Dock app uses `RpcClient` from `react-native-sdk` to handle most of the communication between `react-native-sdk` and the app (e. g. `src/features/wallet/wallet-slice.js:68`). `RpcClient` then passes all calls to `rpcRequest`, which explicitly logs all parameters (including credentials) to the console in production environments (`react-native-sdk`: `src/rpc-client.js:9`). All actions, including secrets, might therefore be kept and stored in log files (depending on the device and configuration of the app store/play store). This can be exploited as other apps could access the log files and read the secrets.

Recommendation

We recommend excluding the payload from logging in production.

Status: Resolved

3. Passwords are not required to include any special characters

Severity: Minor

The wallet's backup is stored in an encrypted JSON file. The encryption key is derived from a password with the following requirements:

- 8-12 characters length
- At least one numeric character
- At least one lowercase character
- At least one uppercase character

However, there is no requirement for special characters to be used. Given that the file is accessible in local storage, a brute force attack is not an unlikely scenario. A 12-character alphanumeric string could be attacked relatively easily locally without any network delays or other delays usually present in remote password verification.

Recommendation

We recommend adding a requirement for special characters to increase the search space significantly.

Status: Resolved

4. Secrets could be leaked by bug tracking tools

Severity: Minor

JSON RPC communication is implemented using the DOM based JavaScript event system to handle requests and responses from `react-native-sdk` (`src/rn-rpc-webview/index.js:86`). Regular JS events are often captured and published by bug tracking and analytics tools to simplify the debugging process for production errors. Dock app includes three of those tools. They might leak those secrets unintentionally because DOM based javascript events are also used for user inputs which are often attached to debug and error reports. The following packages potentially send out events:

- `@sentry/react-native`
- `appcenter-analytics`
- `appcenter-crashes`

Recommendation

We recommend removing any bug tracking and analytics tools from wallets applications. Additionally, we recommend using a different, more secure transport mechanism than the DOM's event system for the JSON RPC communication between `app` and `react-native-sdk` or ensuring that error reports do not contain those event types. Please

note that even if the current version of bug tracking tools might not include such events, any future version could introduce a security leak by capturing more information from the JS DOM event system.

Status: Resolved

5. Cleartext RPC communication could be intercepted

Severity: Minor

JSON RPC communication could be intercepted by any JS dependency because all dependencies have access to the JS DOM event system. If there was a malicious dependency (through dependency injection or code injection), it could send sensitive information to the attacker. Additionally, an attacker could re-send JSON RPC messages, or manipulate them to their benefit, e. g. changing the payload to be signed.

Recommendation

We recommend encrypting the payload to make it more difficult for attackers to get access to the payload.

Status: Resolved

6. Short delay for passcode entry could enable brute force attack

Severity: Minor

The wallet can be unlocked with a 6-digit numerical code. There is no block or additional delay when the code is entered incorrectly repeatedly, besides a 100ms delay between each attempt defined in the following code in `src/features/unlock-wallet/UnlockWalletScreen.js`:

```
setTimeout(() => {
        setPasscode('');
}, 100);
```

A 100ms delay would mean a worst-case time for a brute force attack of fewer than 28 hours. Given the nature of the application, there may be an incentive for such an attack.

Recommendation

We recommend adding an exponentially increasing time lock after a certain number of failed unlock attempts, for example, a 1-minute block after 5 failed attempts.

Status: Resolved

7. Missing error check-in account deletion

Severity: Minor

In the `removeAccount` function in `src/features/accounts/account-slice.js`, accounts are removed by making an RPC call to the SDK. However, the result of this call is not checked for success and the account is removed from the Realm DB in either case.
This might lead to inconsistencies, with accounts being shown as active to the user, whilst the corresponding keys have already been deleted.

Recommendation

We recommend checking the account removal call for errors to keep the database consistent.

Status: Resolved

8. Use of cryptographically unsafe random number generator

Severity: Informational

In `src/features/account-creation/CreateAccountVerifyPhraseScreen.js` file, `getRandomNumbers` function uses `Math.random`, which is not cryptographically safe to use for crypto wallet applications. As the `Math.random` function relies on a weak pseudorandom number generator, it should be avoided in security-critical applications or for protecting sensitive data. In such a context, a cryptographically strong pseudorandom number generator (CSPRNG) should be used instead.
This issue is marked informational since the only use of the random number in this application is the selection of the two mnemonic seed words that need to be verified by the user. For this use case, the use of `Math.random` is acceptable. However, since the random number generation is encapsulated in a generically named function, unsafe re-use might occur in the future.

Recommendation

Consider using a cryptographically secure pseudo-random number generator, such as a Web Crypto API implementation (https://developer.mozilla.org/en-US/docs/Web/API/Crypto/getRandomValues). Alternatively, clearly mark the function as cryptographically insecure, for example by renaming it to `getRandomNumbersUnsafe()`.

9. Hardcoded test account address

Severity: Informational

In `src/features/accounts/account-slice.js` a test account address has been hardcoded in line 70. Whilst this will not affect production, it exposes details on an account

that might have been used by the team once the repository will be open-sourced. This may provide unwanted disclosure of team account activities, and is generally not considered best practice.

Recommendation

We recommend configuring test accounts to be retrieved from the environment context so that they can be easily managed and changed in different environments.

10. Unused imports

Severity: Informational

In several places, unused imports are declared in JavaScript source files:

- `src/core/error-handler.js:17`
- `src/mrklt.js` → whole file

Recommendation

Consider removing unused imports for code readability.

11. Leftover TODO comments

Severity: Informational

The codebase has some unfinished parts that are marked with `TODO` comments:

- `src/features/accounts/AccountsScreen.js` → `appSelectors`
- `src/components/NumericKeyboard.js` → `useState`
- `src/components/ConfirmationModal.js` → `View`

Recommendation

Consider implementing the missing code or removing the `TODO` comments.

12. Unimplemented Feature

Severity: Informational

In `src/features/accounts/AccountDetailsScreen.js` the edit feature is implemented as an alert in line `423-425`:

```
onEdit={() => {
        alert('edit');
}}
```

This seems to be a placeholder for a future implementation.

Recommendation

Consider removing unimplemented feature placeholders in production code.

13. Leftover debug functionality

Severity: Informational

Function `handleLogoPress` in `rc/features/unlock-wallet/UnlockWalletScreen.js` seems to implement a simple error message for testing the Sentry integration when the logo is pressed 5 times. Such functionality might confuse the user.

Recommendation

Consider removing debug features in production mode.

Appendix A

List of high and critical vulnerabilities in the dependency tree.

high	Regular Expression Denial of Service (ReDoS) in Prism
Package	prismjs
Patched in	>=1.24.0
Dependency of	@storybook/addon-actions
Path	@storybook/addon-actions > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1002580

high	Denial of service in prismjs
Package	prismjs
Patched in	>=1.23.0
Dependency of	@storybook/addon-actions
Path	@storybook/addon-actions > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1002944

high	Cross-Site Scripting in Prism
Package	prismjs
Patched in	>=1.21.0
Dependency of	@storybook/addon-actions
Path	@storybook/addon-actions > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1004043

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	jest
Path	jest > jest-cli > @jest/core > jest-config > babel-jest > @jest/transform > jest-haste-map > sane > anymatch > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	jest
Path	jest > jest-cli > @jest/core > jest-runner > jest-config > babel-jest > @jest/transform > jest-haste-map > sane > anymatch > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	jest
Path	jest > jest-cli > @jest/core > jest-runner > jest-runtime > jest-config > babel-jest > @jest/transform > jest-haste-map > sane > anymatch > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	jest
Path	jest > jest-cli > @jest/core > jest-runner > jest-runtime > jest-config > babel-jest > @jest/transform > jest-haste-map > sane > anymatch > micromatch > extglob > expand-brackets > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	jest
Path	jest > jest-cli > @jest/core > jest-runner > jest-runtime > jest-config > babel-jest > @jest/transform > jest-haste-map > sane > anymatch > micromatch > extglob > expand-brackets > snapdragon > base > cache-base > union-value > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Regular expression denial of service in react-native
Package	react-native
Patched in	>=0.64.1
Dependency of	react-native
Path	react-native
More info	https://www.npmjs.com/advisories/1002565

high	Regular Expression Denial of Service (ReDoS) in Prism
Package	prismjs
Patched in	>=1.24.0
Dependency of	@storybook/addon-actions
Path	@storybook/addon-actions > @storybook/components > react-syntax-highlighter > prismjs
More info	https://www.npmjs.com/advisories/1002580

high	Regular Expression Denial of Service (ReDoS) in Prism
Package	prismjs
Patched in	>=1.24.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/ui > @storybook/components > react-syntax-highlighter > prismjs
More info	https://www.npmjs.com/advisories/1002580

high	Regular Expression Denial of Service (ReDoS) in Prism
Package	prismjs
Patched in	>=1.24.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > @storybook/ui > @storybook/components > react-syntax-highlighter > prismjs
More info	https://www.npmjs.com/advisories/1002580

high	Prototype Pollution in immer
Package	immer
Patched in	>=9.0.6
Dependency of	@reduxjs/toolkit
Path	@reduxjs/toolkit > immer
More info	https://www.npmjs.com/advisories/1002487

critical	Prototype Pollution in immer

Package	immer
Patched in	>=9.0.6
Dependency of	@reduxjs/toolkit
Path	@reduxjs/toolkit > immer
More info	https://www.npmjs.com/advisories/1002492

high	Regular Expression Denial of Service (ReDoS) in Prism
Package	prismjs
Patched in	>=1.24.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/ui > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1002580

high	Regular Expression Denial of Service (ReDoS) in Prism
Package	prismjs
Patched in	>=1.24.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > @storybook/ui > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1002580

high	Denial of service in prismjs
Package	prismjs
Patched in	>=1.23.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/ui > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1002944

high	Denial of service in prismjs
Package	prismjs
Patched in	>=1.23.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > @storybook/ui > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1002944

high	Cross-Site Scripting in Prism
Package	prismjs
Patched in	>=1.21.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/ui > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1004043

high	Cross-Site Scripting in Prism
Package	prismjs
Patched in	>=1.21.0
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > @storybook/ui > @storybook/components > react-syntax-highlighter > refractor > prismjs
More info	https://www.npmjs.com/advisories/1004043

high	Prototype Pollution in set-value

Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > webpack > micromatch > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > webpack > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > webpack > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > corejs-upgrade-webpack-plugin > webpack > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > react-dev-utils > globby > fast-glob > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > webpack > watchpack > watchpack-chokidar2 > chokidar > anymatch > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > webpack > watchpack > watchpack-chokidar2 > chokidar > anymatch > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in set-value
Package	set-value
Patched in	>=4.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > corejs-upgrade-webpack-plugin > webpack > watchpack > watchpack-chokidar2 > chokidar > anymatch > micromatch > braces > snapdragon > base > cache-base > set-value
More info	https://www.npmjs.com/advisories/1002475

high	Prototype Pollution in immer
Package	immer
Patched in	>=9.0.6
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > react-dev-utils > immer
More info	https://www.npmjs.com/advisories/1002487

critical	Prototype Pollution in immer
Package	immer
Patched in	>=9.0.6
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > react-dev-utils > immer
More info	https://www.npmjs.com/advisories/1002492

high	Prototype Pollution in immer
Package	immer
Patched in	>=8.0.1
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > react-dev-utils > immer
More info	https://www.npmjs.com/advisories/1002973

high	Uncontrolled Resource Consumption in ansi-html
Package	ansi-html
Patched in	No patch available
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > webpack-hot-middleware > ansi-html
More info	https://www.npmjs.com/advisories/1002522

high	Regular expression denial of service
Package	glob-parent
Patched in	>=5.1.2
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > webpack > watchpack > watchpack-chokidar2 > chokidar > glob-parent
More info	https://www.npmjs.com/advisories/1002627

high	Regular expression denial of service
Package	glob-parent
Patched in	>=5.1.2
Dependency of	@storybook/react-native-server
Path	@storybook/react-native-server > @storybook/core > webpack > watchpack > watchpack-chokidar2 > chokidar > glob-parent
More info	https://www.npmjs.com/advisories/1002627

high	Regular expression denial of service
Package	glob-parent

```
┌───────────────┬──────────────────────────────────────────────────────────────┐
│ Patched in    │ >=5.1.2                                                      │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of │ @storybook/react-native-server                               │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Path          │ @storybook/react-native-server > @storybook/core >           │
│               │ corejs-upgrade-webpack-plugin > webpack > watchpack >        │
│               │ watchpack-chokidar2 > chokidar > glob-parent                 │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ More info     │ https://www.npmjs.com/advisories/1002627                     │
└───────────────┴──────────────────────────────────────────────────────────────┘

┌───────────────┬──────────────────────────────────────────────────────────────┐
│ high          │ Remote Memory Exposure in bl                                 │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Package       │ bl                                                           │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Patched in    │ >=1.2.3                                                      │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of │ react-native-level-fs                                        │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Path          │ react-native-level-fs > levelup > bl                         │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ More info     │ https://www.npmjs.com/advisories/1003058                     │
└───────────────┴──────────────────────────────────────────────────────────────┘

┌───────────────┬──────────────────────────────────────────────────────────────┐
│ high          │ Regular Expression Denial of Service in semver               │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Package       │ semver                                                       │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Patched in    │ >=4.3.2                                                      │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of │ react-native-level-fs                                        │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Path          │ react-native-level-fs > levelup > semver                     │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ More info     │ https://www.npmjs.com/advisories/1004705                     │
└───────────────┴──────────────────────────────────────────────────────────────┘

117 vulnerabilities found - Packages audited: 2570
Severity: 6 Low | 74 Moderate | 35 High | 2 Critical
```