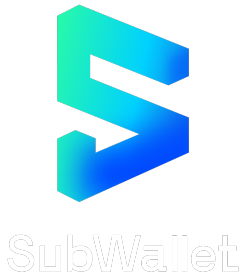
verichains

*SECURITY AUDIT OF*

# SUBWALLET




**Public Report**

*May 23, 2023*


# Verichains Lab

info@verichains.io

https://www.verichains.io

*Driving Technology > Forward*

# ABBREVIATIONS

| Name | Description |
|---|---|
| **Ethereum** | An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications. |
| **Ether (ETH)** | A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network. |
| **Smart contract** | A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract. |
| **Solidity** | A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform. |
| **Solc** | A compiler for Solidity. |
| **ERC20** | ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain. |
| **ERC721** | The ERC-721 introduces a standard for NFT, in other words, this type of Token is unique and can have different value than another Token from the same Smart Contract, maybe due to its age, rarity or even something else like its visual |


# EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on May 23, 2023. We would like to thank the SubWallet for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the SubWallet. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified some vulnerable issues in the application, along with some recommendations.

# TABLE OF CONTENTS

# 1. MANAGEMENT SUMMARY

## 1.1. About SubWallet

SubWallet is a user-friendly Web3 Multiverse Gateway for the Polkadot and Kusama ecosystems. The vision is to provide you with the simplest and most secure way to connect to blockchain-based applications like DeFi and GameFi on Polkadot.

In other words, the SubWallet on Polkadot{.js}'s security backbone while improving all of its current drawbacks, including but not limited to:

- Cumbersome infrastructure
- Complicated user interface
- High latency

## 1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the SubWallet.

It was conducted on commit `98ac5c2302d5a2210b8af343ec3bac00c173bdad` from git repository *https://github.com/Koniverse/SubWallet-Extension*.

## 1.3. Audit methodology

Our security audit process includes four steps:

- Mechanism Design is reviewed to look for any potential problems.
- Source codes are scanned/tested for commonly known and more specific vulnerabilities using public and our in-house security analysis tool.
- Manual audit of the codes for security issues. The source code is manually analyzed to look for any potential problems.
- Set up a testing environment to debug/analyze found issues and verifies our attack PoCs.

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

| SEVERITY LEVEL | DESCRIPTION |
|---|---|
| **CRITICAL** | A vulnerability that can disrupt the functioning; creates a critical risk to the application; required to be fixed immediately. |
| **HIGH** | A vulnerability that could affect the desired outcome of executing the application with high impact; needs to be fixed with high priority. |
| **MEDIUM** | A vulnerability that could affect the desired outcome of executing the application with medium impact in a specific scenario; needs to be fixed. |
| **LOW** | An issue that does not have a significant impact, can be considered as less important. |

*Table 1. Severity levels*

## 1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure application. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

# 2. AUDIT RESULT

## 2.1. Overview

The SubWallet was written in `TypeScript` Programming Language using `React Native` Framework.

The main features of the SubWallet are:

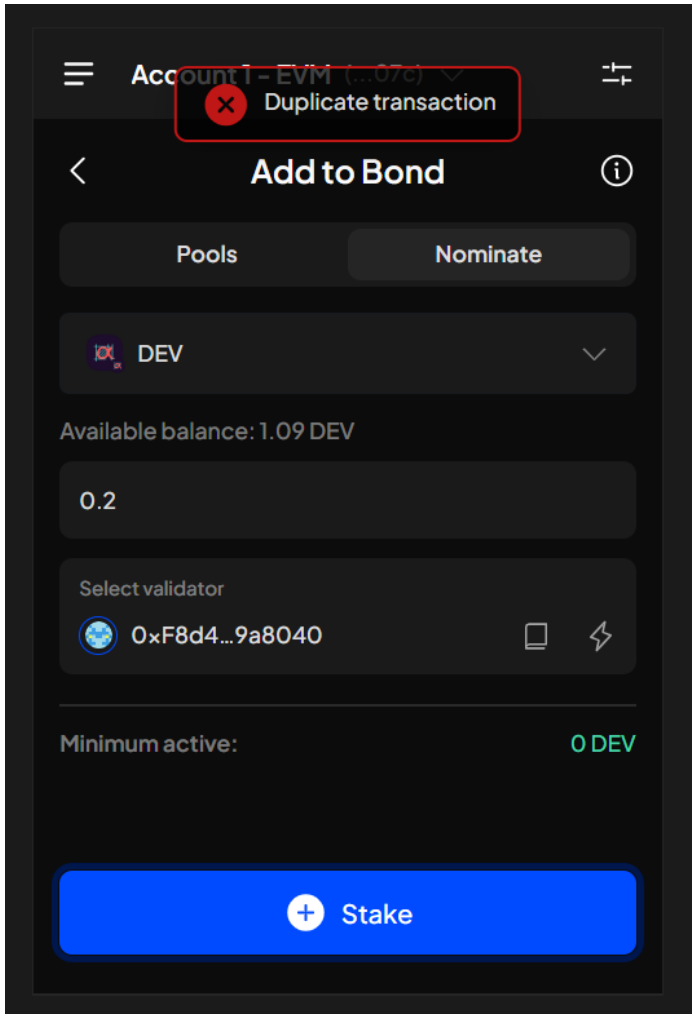
- Manage multiple wallets under one account.
- Import and restore an account
- Import token
- Collect and transfer NFTs
- Transfer and receive assets
- Buy crypto with fiat money
- Perform XCM transfer
- Manage Crowdloan
- Manage Staking
- Export and backup an account

## 2.2. Findings

This section contains a detailed analysis of all the vulnerabilities that were discovered by the audit team during the audit process.

### 2.2.1. A pending transaction frenzy can lead to duplicate transactions being created within the same session - MEDIUM

When users submit a transaction, the wallet stores it in memory with the state pending. If the RPC or user's networking experiences have any issues, the pending transaction freezes in memory, preventing the user from sending any new transactions during the session.

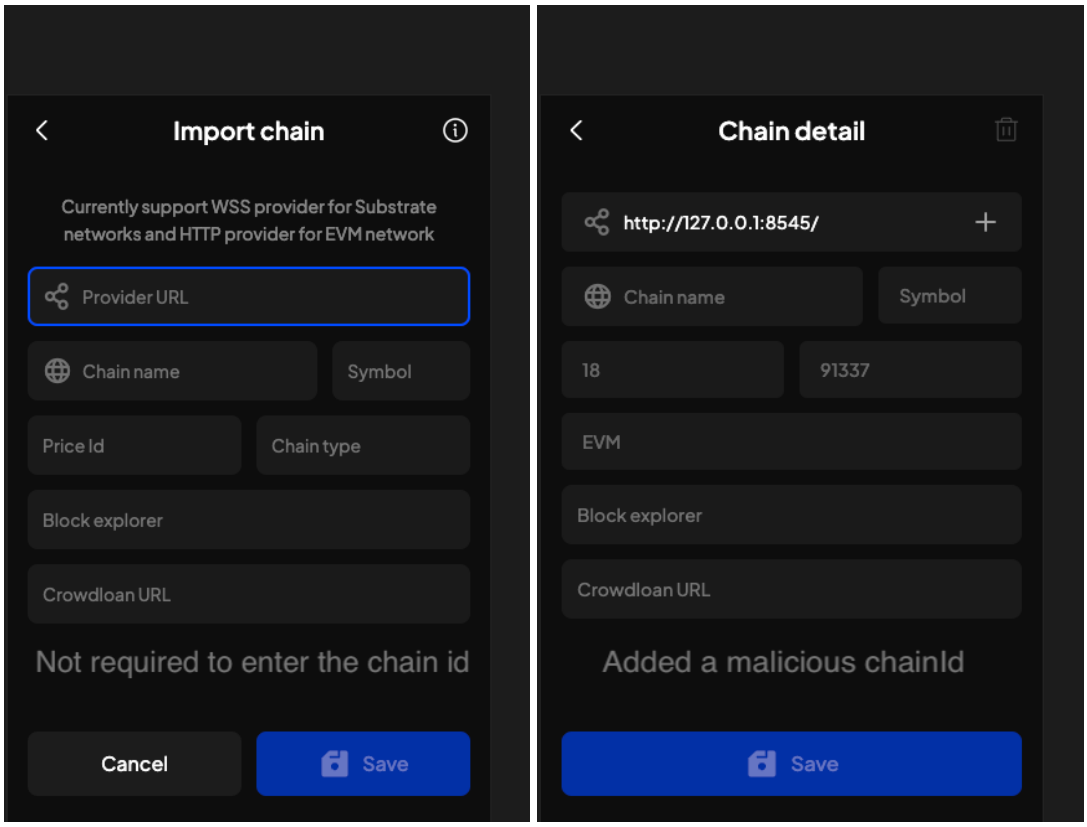



**UPDATES**

- May 04, 2023: This issue has been acknowledged by the team, and they are working on a fix. The team has come up with a solution where is every transaction that is submitted to the chain will automatically change to failed if the wallet is not listening to any events about this transaction. After this timeout, user submits a new transaction with the same network and address. In the future, the team will implement a transaction queue to allow users to send multiple transactions without waiting.
- May 23, 2023: This issue has been fixed by the team in the version `v1.0.5-2`.

### 2.2.2. Custom networks do not require chainId and accept malicious chainId from RPC endpoints - MEDIUM

If a user adds a custom network to their wallet, they are not required to manually enter the chain id of the network. Instead, the wallet will automatically request the eth_chainId from an RPC without any user verification. This poses a risk of phishing attacks if the custom chainId is not listed on `chainid.network`.




#### RECOMMENDATION

We recommend that the team require users to manually enter the chain id of the network when they add a custom network to their wallet.

#### UPDATES

- May 23, 2023: This issue has been acknowledged and fixed by the team in the version `v1.0.5-2`.

### 2.2.3. Using out-of-date version of @Apollo with vulnerable libraries - LOW

The most recent version of Apollo is 2.33.9, but it contains libraries that are vulnerable to issues like regular expression denial of service (reDos), bypassing authorization using a user-controlled key, forging requests on the server, poor input validation, etc.

The latest version is 2.34.0

However, a team from Apollo is working to completely deprecate this library and any projects that are related to it.

**RECOMMENDATION**

We recommend that the team update to the most recent version and consider replacing this library with a new one.

**UPDATES**

- May 04, 2023: This issue has been acknowledged and fixed by the team.

### 2.2.4. Using an out-of-date version of @subwallet/chain-list with vulnerable @gh-pages - LOW

Version 4.0.0 of gh-pages is included in the extension-base package. However, this library is vulnerable to a prototype pollution attack and has been patched as of version 5.0.0 or above.

**RECOMMENDATION**

We recommend that the team update to the most recent version.

**UPDATES**

- May 04, 2023: This issue has been acknowledged and fixed by the team.

### 2.2.5. Using an out-of-date version of @web3 leads to insecure credential storage - LOW

A package extension-base used a vulnerable web3. As there is no fix, all versions were impacted.

**UPDATES**

- May 04, 2023: This issue has been acknowledged by the team.
- May 23, 2023: The team upgraded web3* packages to the most recent version, which is `^1.10.0` in SubWallet version `v1.0.5-2` and the SubWallet not use the web3 to save encrypted password.

### 2.2.6. No policy for a master password - LOW

The SubWallet does not require users to create a strong password when they create a new account or change their password.

**RECOMMENDATION**

We recommend that the team require users to create a strong password when they create a new account or change their password. This will help prevent weak passwords that are easy to guess or crack, and increase the overall security of the platform.

**UPDATES**

- May 23, 2023: This issue has been acknowledged and fixed by the team in the version `v1.0.5-2`.

# 3. VERSION HISTORY

| Version | Date | Status/Change | Created by |
|---|---|---|---|
| **1.0** | *May 15, 2023* | Private Report | Verichains Lab |
| **1.1** | *May 23, 2023* | Public Report | Verichains Lab |

*Table 2. Report versions history*