// HALBORN

AVA Labs

Wallet SDK Pentesting

Prepared by: **Halborn**

Date of Engagement: **July 1st, 2021 - July 24th, 2021**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	07/19/2021	Ataberk Yavuzer
0.9	Document Edits	07/24/2021	Ataberk Yavuzer
1.0	Final Version	07/26/2021	Gabi Urrutia
1.1	Remediation Plan	08/25/2021	Ataberk Yavuzer
1.1	Remediation Plan Review	08/25/2021	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Ataberk Yavuzer	Halborn	Ataberk.Yavuzer@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

AVA Labs engaged Halborn to conduct a security assessment on their Wallet Software Development Kit beginning on July 1st and ending on July 24th, 2021.

AVA Labs Wallet SDK is a TypeScript library for creating and managing non-custodial wallets on the Avalanche network. It provides high-level methods to transact on Avalanche's X-Chain, P-Chain and C-Chain.

The security assessment was scoped to the Github repository of Wallet SDK. An audit of the security risk and implications regarding the changes introduced by the development team at AVA Labs prior to its production release shortly following the assessments deadline.

Though this security audit's outcome is satisfactory, only the most essential aspects were tested and verified to achieve objectives and deliverable set in the scope due to time and resource constraints. It is essential to note the use of the best practices for secure SDK development.

1.2 AUDIT SUMMARY

The team at Halborn was provided nearly four weeks for the engagement and assigned a full time security engineer to audit the security of the Wallet SDK. The security engineer is blockchain and smart-contract security experts with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit to achieve the following:

- Ensure that Wallet SDK functions are intended.
- Identify potential security issues with the Wallet SDK.

In summary, Halborn identified few security risks, and recommends performing further testing to validate extended safety and correctness in

context to the whole structure.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of the Wallet SDK. While manual testing is recommended to uncover flaws in logic, process,and implementation; automated testing techniques help enhance coverage of structures and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose.
- Static Analysis of security for scoped SDK, and imported functions. (nodejsscan, Dependency-Check, eslint, LGTM)
- Manual Assessment for discovering security vulnerabilities on code-base.
- Ensuring correctness of the codebase (jest)
- Dynamic Analysis on SDK functions and data types.

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident, and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. It's quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that was used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.

3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

IN-SCOPE:
The security assessment was scoped to ava-labs/avalanche-wallet-sdk repository.

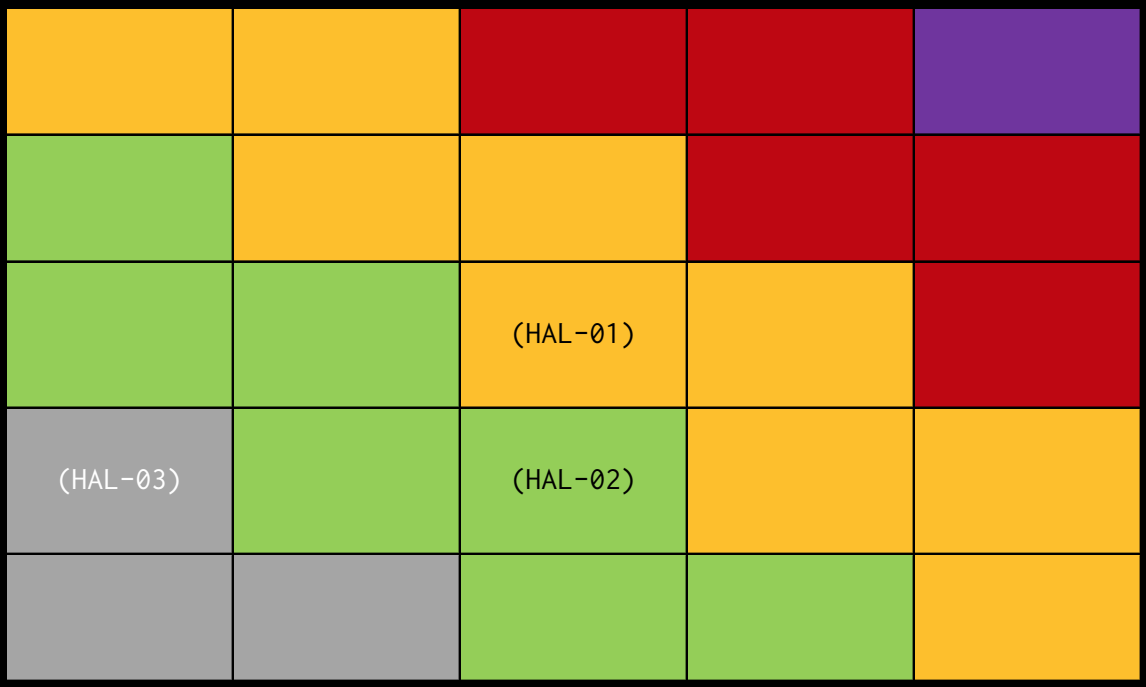
Commit ID: e63e9e10d63bb3bc7e63005c4ab4945948003cee

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	1	1	1

LIKELIHOOD

IMPACT



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - UNFILTERED TOKEN DATA LEADS CROSS-SITE SCRIPTING	Medium	SOLVED: 08/25/2021
HAL02 - UNHANDLING ERRORS LEADS DENIAL OF SERVICE	Low	SOLVED: 08/25/2021
HAL03 - HARDCODED TEST CREDENTIALS	Informational	NOT APPLICABLE
STATIC ANALYSIS	-	-

FINDINGS & TECH DETAILS

3.1 (HAL-01) UNFILTERED TOKEN DATA LEADS CROSS-SITE SCRIPTING - MEDIUM

Description:

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

During the test, it is seen that there are no input/output filtering on the Wallet SDK. For example, some functions directly fetches token data without filtering the output. There is a possibility to trigger Cross-Site Scripting vulnerability by fetching malicious token data. Also, there are many functions that don't filter the input/output.

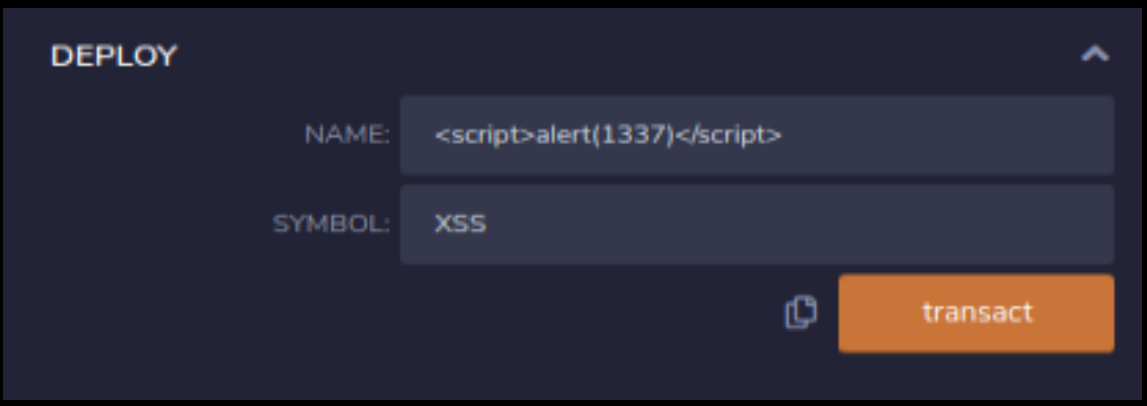


Figure 1: Deploying malicious ERC20 token to the Fuji Network

Figure 2: The output of name() function on the malicious token

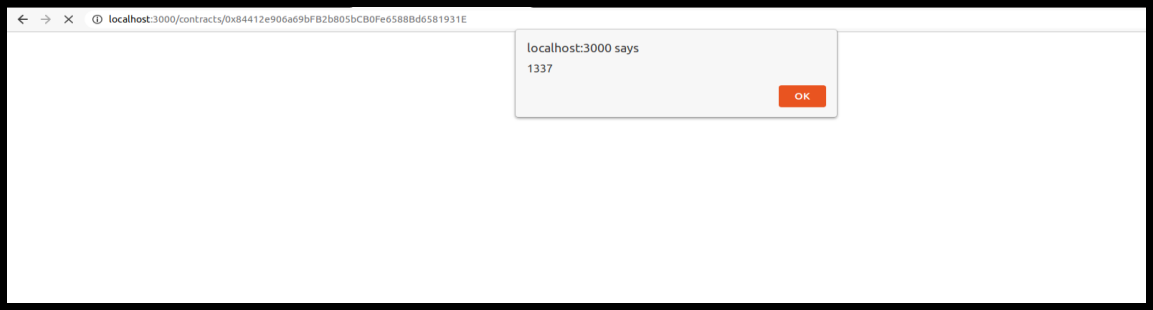


Figure 3: Triggering the XSS vulnerability by reflecting unfiltered token.name on the page

Code Location:

Vulnerable Functions:

Listing 1: Vulnerable Functions

```
1 Assets.getContractData(address: string)
2 Assets.getContractDataErc20(address: string)
3 Assets.getAssetDescription(assetId: string)
4 Assets.getAssetDescriptionSync(assetId: string)
5 Assets.getErc20Token(address: string)
6 Assets.getErc20Store()
7 Assets.getErc20StoreCustom()
```

Proof of Concept:

Listing 2: index.js (Lines 12)

```
1  const Assets = require('@avalabs/avalanche-wallet-sdk').Assets
2  const Network = require('@avalabs/avalanche-wallet-sdk').Network
3  const NetworkConstants = require('@avalabs/avalanche-wallet-sdk').
       NetworkConstants
4
5  const express = require('express')
6  const app = express()
7  const port = 3000
8
9  Network.setNetwork(NetworkConstants.TestnetConfig)
10
11 app.get('/contracts/:address', (req, res) => {
12     Assets.getContractData(req.params.address).then(token => {
13         res.send(token.name) //Unfiltered Token Name on the
               getContractData() function leads to Cross-Site
               Scripting Vulnerability.
14         console.log(data)
15     }).catch(err => {
16         res.send("Error")
17     })
18 })
19
20 app.listen(port, () => {
21     console.log('Example app listening at http://localhost:' +
           port)
22 })
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendations:

It is highly recommended to using HTML Encode before inserting untrusted data into HTML element content. All user inputs and outputs should be filtered properly before sending the necessary data to the user. Also, DOMPurify the powerful XSS sanitizer library that can be used on all inputs/outputs.

References:

OWASP Cross-Site Scripting
DOMPurify

Remediation Plan:

SOLVED: AVA Labs Team solved this issue by using the DOMPurify library suggested by Halborn Team.

3.2 (HAL-02) UNHANDLING ERRORS LEADS DENIAL OF SERVICE - LOW

Description:

Exception handling is one of the most important part of safe and clean programming. It is important that exception handling is implemented properly so that the flow of the application is not disrupted. In some cases, if this operation is not done correctly and the program receives an unexpected input, the program may react unexpectedly to this process and the program may shut itself.

During the test, it is seen that some functions on the Wallet SDK do not manage exceptions properly. For example, one function handles the exception while the similar one does not handle it. There is a function that cannot handle the error properly causes the running program to stop.

```
PS C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests> node .\index.js
Example app listening at http://localhost:3000
C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\avalanche\dist\common\jrpcapi.js:73
                        throw new Error(`${resp.data.error.message}`);
                              ^

Error: asset '1' not found
    at C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\avalanche\dist\common\jrpcapi.js:73:31
    at processTicksAndRejections (node:internal/process/task_queues:96:5)
PS C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests>
```

Figure 4: Unhandled Error leads to DoS on User Application - getAssetDescription()

```
Error: Asset ID 1 is not known.
    at Object.getAssetDescriptionSync (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\@avalabs\avalanche-wallet-sdk\dist\index.js:1533:15)
    at C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\index.js:15:24
    at Layer.handle [as handle_request] (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\layer.js:95:5)
    at next (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\route.js:137:13)
    at Route.dispatch (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\route.js:112:3)
    at Layer.handle [as handle_request] (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\layer.js:95:5)
    at C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:281:22
    at param (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:354:14)
    at param (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:365:14)
    at Function.process_params (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:410:3)
    at next (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:275:10)
    at expressInit (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\middleware\init.js:40:5)
    at Layer.handle [as handle_request] (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\layer.js:95:5)
    at trim_prefix (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:317:13)
    at C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:284:7
    at Function.process_params (C:\Users\goku\Desktop\ata_audits\wallet-sdk-tests\node_modules\express\lib\router\index.js:335:12)
```

Figure 5: Properly Handled Error - getAssetDescriptionSync()

Code Location:

Properly Handled Error:

Listing 3: src/Asset/Assets.ts - Handled Error (Lines 2)

```
1 export function getAssetDescriptionSync(assetId: string):
      iAssetDescriptionClean {
2     if (typeof assetCache[assetId] === 'undefined') throw new
          Error(`Asset ID ${assetId} is not known.`);
3     return assetCache[assetId];
4 }
```

Unhandled Error:

Listing 4: src/Asset/Assets.ts - Unhandled Error

```
1 export async function getAssetDescription(assetId: string):
      Promise<iAssetDescriptionClean> {
2     let cache = assetCache[assetId];
3     if (cache) {
4         return cache;
5     }
6
7     let res = await xChain.getAssetDescription(assetId);
8     let clean: iAssetDescriptionClean = {
9         ...res,
10        assetID: assetId,
11    };
12
13    assetCache[assetId] = clean;
14    return clean;
15 }
```

Possible Vulnerable Public Functions with invalid inputs:

Listing 5: Possible Vulnerable Functions

```
Assets.getAssetDescription(1)
Assets.getContractData(1)
Assets.getErc20Token(1)
LedgerWallet.fromApp(1,1)
LedgerWallet.fromTransport(1)
LedgerWallet.getAvaxAccount(1)
LedgerWallet.getEvmAccount(1)
Utils.waitTxC(1,1,1,1)
Utils.waitTxEvm(1,1)
Utils.waitTxP(1,1)
Utils.waitTxX(1,1)
```

Proof of Concept:

Listing 6: index.js

```
const express = require('express')
const app = express()
const port = 3000


const WalletSDK = require("@avalabs/avalanche-wallet-sdk")

app.get('/', (req, res) => {
    //property based test --> WalletSDK.{Vulnerable function with
        invalid args}
    //example: WalletSDK.Utils.waitTxX(1,1)
    res.send("Halborn")
})

app.listen(port, () => {
    console.log('Example app listening at http://localhost:' +
        port)
})
```

Risk Level:

Likelihood - 3
Impact - 2

Recommendations:

Exception handling is recommended to be successful for the specified functions. Otherwise, applications using the Wallet SDK will exit unexpectedly if these functions are called incorrectly.

Remediation Plan:

SOLVED: AVA Labs Team solved the issue by adding additional error handling to the mentioned functions on this report. Furthermore, consumer apps should also handle errors while using the Wallet SDK additionally.

3.3 (HAL-03) HARDCODED TEST CREDENTIALS - INFORMATIONAL

Description:

Hardcoded credentials, also referred as Embedded Credentials, are plain-text passwords or other secret keys in source code. Password hardcoding refers to the practice of embedding plain text (non-encrypted) passwords and other secrets into the source code. If attackers access these credentials somehow, it is possible to breach test accounts.

During the test, the Halborn team detected some test credentials on the Wallet SDK codebase. These credentials are only used for test purposes. However, if these key files are used for real purposes, wallets can be hacked using these hardcoded key files.

Code Location:

Listing 7: test/Keystore/keystore.test.ts

```
1 import Keystore from '@/Keystore/keystore';
2
3 // A 2.0 and 3.0 Keystore version of the same keys
4 // Passwords are: 111111111
5 const KEYFILE_2 = `{"version":"2.0","salt":"2
    SjQXSMR87tBvYqbkwTFL61gEdwR","pass_hash":"2
    NJf6rqPshCU69hMkPEMBLBZLfBKshHy68cWgNY7kNmAM988Qt","keys":[{"
    key":"REDACTED","iv":"Fc8Xyxmhd2X55sgjy4aTxN","address":"X-
    EAZkJNdFBjVQQ7zS81hWCnRHfMKf3vpYH"},{"key":"REDACTED","iv":"
    N6fYr5gT4TJfB6Tzs9oLMN","address":"X-7
    r1aDs2jiJHr1reFfLFzzKrprZruXVzqM"}]}`;
6 const KEYFILE_3 = `{"version":"3.0","salt":"
    kwkVtmPkafnwWbp65nYs2z9cQeN","pass_hash":"2
    gid7yJzvyg2Mz4HUJLh3jvgumpDJmRu2PBopqHYacVjwisp1g","keys":[{"
    key":"REDACTED","iv":"L7MojgHmudo2WpbMCdfgCg","address":"X-7
    r1aDs2jiJHr1reFfLFzzKrprZruXVzqM"},{"key":"REDACTED","iv":"
    AN8nLnaK84rfoKXtxm6evy","address":"X-
    EAZkJNdFBjVQQ7zS81hWCnRHfMKf3vpYH"},{"key":"REDACTED","iv":"
    U3VZnEr8HgiD3rcNW6Abko","address":"X-7
```

```
      r1aDs2jiJHr1reFfLFzzKrprZruXVzqM"}],"warnings":["This address
      listed in this file is for internal wallet use only. DO NOT USE
       THIS ADDRESS"]}`;
 7 const KEYFILE_4 = `{"version":"4.0","salt":"
      UWLRsfsyjY51E1s8CVa7cvvMHMz","pass_hash":"
      M6mzyfS4i4bKBxXQZFuQ6BsRnMSVMe7GnBd1HTmVLi2jcscPA","keys":[{"
      key":"REDACTED","iv":"A6jQMX7e6doGS6wVvdLzA"}]}`;
 8 const KEYFILE_5 = `{"version":"5.0","salt":"
      TEUQMGR1XdHs5wb4SVSA7LriUhR","pass_hash":"
      avkdK9YFLn1zfZjvBsB9ipKRzfqr4rvqBryVosB6NUgFiv9kd","keys":[{"
      key":"REDACTED","iv":"Ak8DSMKMy4f1RXHSSt15KK"}]}`;
 9 const KEYFILE_6 = `{"version":"6.0","salt":"2
      NgqFaoYSe5foo8oEtcdB658c7Eb","activeIndex":0,"keys":[{"key":"
      REDACTED","iv":"TQei93ehgGWgUKXkLha8Ef","type":"mnemonic"}]}`;
10
11 const KEY_V2_V3 = '2
      DvMW4ZsNVdiiBsrEdPBTDr47bTtgr4H8qQKXz2D37YKeTLwDw';
12 const KEY_V4 = 'jegD9bfh1qYjnyxUgnG92CEyAx7s4iZRgcYatdN2u1qhy1Tbr'
      ;
13 const KEY_V5_V6 =
14     'solar ordinary sentence pelican trim ring indicate cake
          ordinary water size improve impose gentle frown sound know
          siren sick elder wait govern tortoise unit';
15
```

Risk Level:

Likelihood - 1
Impact - 2

Recommendations:

It is known that these key files are used for testing purposes. However, it is recommended to remove these test key files from the repository for security reasons. If these key files are necessary to be used for test cases, these should be transmitted in another secure way. In addition, all credentials on the test directory should be removed.

Remediation Plan:

NOT APPLICABLE: AVA Labs Team claims that it is not possible to use test cases without defining these test credentials. For that reason, it has been deemed appropriate that this weakness is not applicable.

STATIC ANALYSIS REPORT

4.1 STATIC ANALYSIS

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped items. Among the tools used ESLint, NodeJSScan, Jest and LGTM. ESLint tries to statically analyzes the code to quickly find problems related with JavaScript projects. NodeJSScan contains multiple security assessment tools in it such as Dependency-Scan of OWASP. Jest tool is already appended by AVA Labs Team on the project to state if functions working proper. Lastly, LGTM analyzes the entire history of a project, so you can see how your alerts have changed over time, and which specific events or commits had the biggest impact on your code quality. Also, it checks the codebase to find security vulnerabilities. After Halborn verified the codebase in the repository and was able to work on it correctly, all of these tools were run on the all-scoped structures. These tools can statically verify security related issues across the entire codebase.

4.2 ESLint

According to the following screenshots, there could be multiple improvements on the codebase. It has been decided to not to put these issues on the report in details because these issues do not pose any security risk.

These improvements are listed below:

1. Removing Unused Variables
2. Adding return type to functions
3. Changing “any” keyword to specific data type

```
C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\Explorer\Explorer.ts
   4:1  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  17:1  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  30:1  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\History\history.ts
    3:5   warning  'HistoryItemType' is defined but never used       @typescript-eslint/no-unused-vars
   10:5   warning  'iHistoryBaseTxTokens' is defined but never used  @typescript-eslint/no-unused-vars
   17:5   warning  'iHistoryItem' is defined but never used          @typescript-eslint/no-unused-vars
   23:25  warning  'avalanche' is defined but never used             @typescript-eslint/no-unused-vars
  581:14  warning  Unexpected any. Specify a different type          @typescript-eslint/no-explicit-any
  597:14  warning  Unexpected any. Specify a different type          @typescript-eslint/no-explicit-any

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\History\types.ts
  61:18  warning  Unexpected any. Specify a different type  @typescript-eslint/no-explicit-any

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\Keystore\keystore.ts
   44:7  warning  'SUPPORTED_VERSION' is assigned a value but never used  @typescript-eslint/no-unused-vars
  275:9  warning  'chainID' is assigned a value but never used           @typescript-eslint/no-unused-vars

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\Network\index.ts
   6:8  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  13:8  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\Network\providers\AVMWebSocketProvider.ts
  12:19  warning  Unexpected any. Specify a different type  @typescript-eslint/no-explicit-any
  36:5   warning  Missing return type on function           @typescript-eslint/explicit-module-boundary-types
  46:5   warning  Missing return type on function           @typescript-eslint/explicit-module-boundary-types
  50:5   warning  Missing return type on function           @typescript-eslint/explicit-module-boundary-types
  60:5   warning  Missing return type on function           @typescript-eslint/explicit-module-boundary-types
  80:5   warning  Missing return type on function           @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\Network\providers\EVMWebSocketProvider.ts
  68:43  warning  Unexpected any. Specify a different type  @typescript-eslint/no-explicit-any
  72:44  warning  Unexpected any. Specify a different type  @typescript-eslint/no-explicit-any

C:\Users\goku\Desktop\ata_audits\avalanche-wallet-sdk\src\Wallet\EvmWallet.ts
   1:10  warning  'BN' is defined but never used               @typescript-eslint/no-unused-vars
   2:10  warning  'privateToAddress' is defined but never used  @typescript-eslint/no-unused-vars
   2:45  warning  'importPublic' is defined but never used      @typescript-eslint/no-unused-vars
  39:5   warning  Missing return type on function              @typescript-eslint/explicit-module-boundary-types
```

Figure 6: ESLint Results - 1

```
C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\EvmWalletReadonly.ts
   3:27  warning  'importPublic' is defined but never used  @typescript-eslint/no-unused-vars
  32:5   warning  Missing return type on function          @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\HDWalletAbstract.ts
  49:5  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\HdScanner.ts
  39:5  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  47:5  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  51:5  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\LedgerWallet.ts
    8:24  warning  'publicToAddress' is defined but never used               @typescript-eslint/no-unused-vars
   26:17  warning  'AVMExportTx' is defined but never used                   @typescript-eslint/no-unused-vars
   44:17  warning  'PlatformExportTx' is defined but never used              @typescript-eslint/no-unused-vars
   51:18  warning  'BN' is defined but never used                            @typescript-eslint/no-unused-vars
   83:5   warning  Missing return type on function                           @typescript-eslint/explicit-module-boundary-types
   83:32  warning  Argument 'transport' should be typed with a non-any type  @typescript-eslint/explicit-module-boundary-types
   83:43  warning  Unexpected any. Specify a different type                  @typescript-eslint/no-explicit-any
  259:5   warning  Missing return type on function                           @typescript-eslint/explicit-module-boundary-types
  325:5   warning  Missing return type on function                           @typescript-eslint/explicit-module-boundary-types
  373:13  warning  'title' is assigned a value but never used               @typescript-eslint/no-unused-vars
  456:5   warning  Missing return type on function                           @typescript-eslint/explicit-module-boundary-types
  457:46  warning  Unexpected any. Specify a different type                  @typescript-eslint/no-explicit-any
  471:9   warning  Argument 'sigMap' should be typed with a non-any type     @typescript-eslint/explicit-module-boundary-types
  471:17  warning  Unexpected any. Specify a different type                  @typescript-eslint/no-explicit-any

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\MnemonicWallet.ts
  15:17  warning  'EVMKeychain' is defined but never used  @typescript-eslint/no-unused-vars

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\PublicMnemonicWallet.ts
  40:11  warning  'tx' is defined but never used  @typescript-eslint/no-unused-vars
  45:13  warning  'tx' is defined but never used  @typescript-eslint/no-unused-vars
  49:11  warning  'tx' is defined but never used  @typescript-eslint/no-unused-vars
  54:11  warning  'tx' is defined but never used  @typescript-eslint/no-unused-vars

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\Wallet\Wallet.ts
   11:5   warning  'WalletCollectiblesX' is defined but never used  @typescript-eslint/no-unused-vars
   99:59  warning  Unexpected any. Specify a different type         @typescript-eslint/no-explicit-any
  103:60  warning  Unexpected any. Specify a different type         @typescript-eslint/no-explicit-any
  247:13  warning  'oldUtxos' is assigned a value but never used    @typescript-eslint/no-unused-vars
  496:5   warning  Missing return type on function                  @typescript-eslint/explicit-module-boundary-types
  559:5   warning  Missing return type on function                  @typescript-eslint/explicit-module-boundary-types
  594:5   warning  Missing return type on function                  @typescript-eslint/explicit-module-boundary-types
  609:5   warning  Missing return type on function                  @typescript-eslint/explicit-module-boundary-types
  683:5   warning  Missing return type on function                  @typescript-eslint/explicit-module-boundary-types
```

Figure 7: ESLint Results - 2

```
C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\helpers\tx_helper.ts
    6:5   warning  'AssetAmountDestination' is defined but never used  @typescript-eslint/no-unused-vars
    7:5   warning  'BaseTx' is defined but never used                  @typescript-eslint/no-unused-vars
   10:5   warning  'TransferableInput' is defined but never used       @typescript-eslint/no-unused-vars
   11:5   warning  'TransferableOutput' is defined but never used      @typescript-eslint/no-unused-vars
   23:19  warning  'PlatformUnsignedTx' is defined but never used      @typescript-eslint/no-unused-vars
   27:24  warning  'EVMUnsignedTx' is defined but never used           @typescript-eslint/no-unused-vars
  157:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types
  222:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types
  245:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types
  259:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types
  284:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types
  312:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types
  345:8   warning  Missing return type on function                     @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\helpers\universal_tx_helper.ts
  5:18  warning  'web3' is defined but never used    @typescript-eslint/no-unused-vars
  5:24  warning  'xChain' is defined but never used  @typescript-eslint/no-unused-vars

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\helpers\utxo_helper.ts
   50:16  warning  'evmGetAtomicUTXOs' is defined but never used            @typescript-eslint/no-unused-vars
   89:67  warning  Argument 'endIndex' should be typed with a non-any type  @typescript-eslint/explicit-module-boundary-types
   89:78  warning  Unexpected any. Specify a different type                 @typescript-eslint/no-explicit-any
   99:9   warning  'utxos' is assigned a value but never used               @typescript-eslint/no-unused-vars
  126:72  warning  Argument 'endIndex' should be typed with a non-any type  @typescript-eslint/explicit-module-boundary-types
  126:83  warning  Unexpected any. Specify a different type                 @typescript-eslint/no-explicit-any

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\src\utils\utils.ts
   53:8  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  116:8  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  120:8  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types
  124:8  warning  Missing return type on function  @typescript-eslint/explicit-module-boundary-types

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\test\Wallet\EvmWallet.test.ts
  8:13  warning  'wallet' is assigned a value but never used  @typescript-eslint/no-unused-vars

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\test\Wallet\EvmWalletReadonly.ts
  1:8  warning  'EvmWalletReadonly' is defined but never used  @typescript-eslint/no-unused-vars

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\test\Wallet\MnemonicWallet.test.ts
  2:8   warning  'Web3' is defined but never used          @typescript-eslint/no-unused-vars
  6:19  warning  Unexpected any. Specify a different type  @typescript-eslint/no-explicit-any
  8:18  warning  Unexpected any. Specify a different type  @typescript-eslint/no-explicit-any

C:\Users\goku\Desktop\ata audits\avalanche-wallet-sdk\test\Wallet\PublicMnemonicWallet.test.ts
  11:13  warning  'wallet' is assigned a value but never used  @typescript-eslint/no-unused-vars
```

Figure 8: ESLint Results - 3

4.3 Jest

As a result of the tests carried out with **Jest**, it has been decided the project has average code coverage. Also, completed tests have shown that some functions run slower than others.

```
λ yarn test
yarn run v1.22.10
$ jest
 PASS  test/helpers/network_helper.test.ts (7.475 s)
 PASS  test/helpers/address_helper.test.ts (16.349 s)
 PASS  test/helpers/universal_tx_helper.test.ts (16.379 s)
 PASS  test/Wallet/EvmWallet.test.ts (16.962 s)
 PASS  test/Keystore/keystore.test.ts (17.264 s)
 PASS  test/Wallet/MnemonicWallet.test.ts (17.682 s)
 PASS  test/Wallet/PublicMnemonicWallet.test.ts (18.1 s)

Test Suites: 7 passed, 7 total
Tests:       21 passed, 21 total
Snapshots:   0 total
Time:        19.365 s
Ran all test suites.
Done in 24.51s.
```

Figure 9: Jest

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	46.67	80.26	19.55	46.67	
src	100	100	100	100	
common.ts	100	100	100	100	
errors.ts	100	100	100	100	
src/Asset	68.42	80	21.43	68.42	
Assets.ts	43.33	100	0	43.33	8-10,17-30
Erc20.ts	74.54	75	25	74.54	126-129,132-135,143-153,159-160,168-170,173-179,186-209
Erc20Token.ts	58.62	100	25	58.62	30-31,34-52,55-57
src/Explorer	16	100	0	16	
Explorer.ts	16	100	0	16	4-15,17-27,30-48
src/History	13.04	100	0	13.04	
history.ts	13.04	100	0	13.04	...17,219-228,230-234,244-268,270-303,313-337,339-365,367-397,399-421,424-450,452-537,539-555,557-572,574-588,590-604,607-620,623-636,638-644
src/Keystore	73.33	70	66.67	73.33	
Crypto.ts	74.23	72.73	80	74.23	74,85-88,101-102,106-107,123-163
keystore.ts	72.87	68.42	54.55	72.87	69-70,107-108,145-146,184-185,232-233,272-294,296-301,303-308,310-325,334-374
src/Network	69.92	100	0	69.92	
constants.ts	100	100	100	100	
network.ts	43.66	100	0	43.66	30-42,45-71
src/Wallet	45.97	96.3	18.92	45.97	
EvmWallet.ts	74	100	28.57	74	25-26,29-32,35-37,40-41,44-45
EvmWalletReadonly.ts	72.97	100	50	72.97	19-20,27-30,33-36
HDWalletAbstract.ts	56.02	100	25	56.02	27-28,34-35,64-65,71-72,78-79,82-83,89-90,99-107,110-137,140-141,144-145,148-165
HdScanner.ts	40.91	100	35.29	40.91	40-41,44-45,56-62,65-70,73-79,82-88,91-104,107-121,151-161,166-197,202-241
MnemonicWallet.ts	77.71	75	21.43	77.71	39-40,57-60,66-67,74-76,82-83,98-99,106-107,114-115,124-125,132-135,142-143,156-165
PublicMnemonicWallet.ts	82.46	100	28.57	82.46	36-37,41-42,46-47,50-51,55-56
Wallet.ts	34.16	100	2.27	34.16	...86,497-517,528-548,560-592,595-598,601-603,610-645,648-681,684-703,706-726,729-747,763-818,821-871,874-876,879-881,884-886,889-891,894-930
constants.ts	100	100	100	100	
src/helpers	48.44	71.43	22.73	48.44	
address_helper.ts	64.71	50	50	64.71	11-12,20-29
network_helper.ts	100	60	100	100	4,9
tx_helper.ts	50.96	100	0	50.96	158-187,190-220,223-243,246-257,260-282,285-310,313-343,346-375
universal_tx_helper.ts	74.24	50	100	74.24	50-66
utxo_helper.ts	20.14	100	0	20.14	14-29,33-47,50-57,60-73,76-87,90-108,112-124,127-144
src/utils	40.31	100	0	40.31	
utils.ts	40.31	100	0	40.31	...,67-68,83-84,92-93,101-102,110-114,117-118,121-122,125-126,142-144,147-167,180-182,191-192,195-200,203-231,234-262,265-284,288-307,312-320

Figure 10: Jest Coverage

Furthermore, the “hdkey” library causes some performance issues such as working slow while generating new wallets. Instead of using the “hdkey” library, the “hdkey-wasm” library can be used to achieve higher speed on generating a wallet.

hdkey-wasm:
https://www.npmjs.com/package/hdkey-wasm

4.4 NodeJSScan

As a result of the scans completed with NodeJSScan, many tests were carried out and some issue outputs were produced as a result of these tests. These generated issues were analyzed manually. It has been decided that these issues are not real security vulnerabilities and they do not risk security of the Wallet SDK. As one of the important security assessments of NodeJSScan that is OWASP Dependency-Scan, it is seen that used libraries are safe.

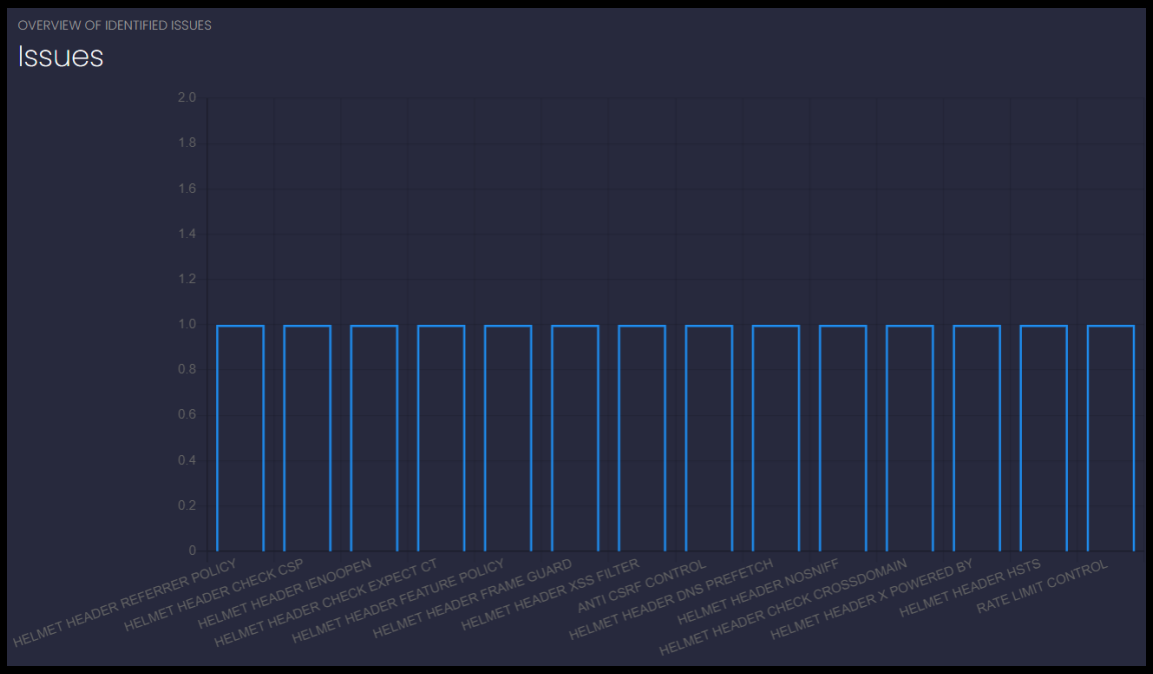


Figure 11: NodeJSScan Results

4.5 LGTM

The project has been security scanned with LGTM and results similar to ESLint outputs have been achieved. As a result of the scans with this tool, no security vulnerabilities were found. During the tests with this tool, only **Unused Import** findings were reached. The reason for scanning the project publicly with LGTM is that the Github repository is a public repository.

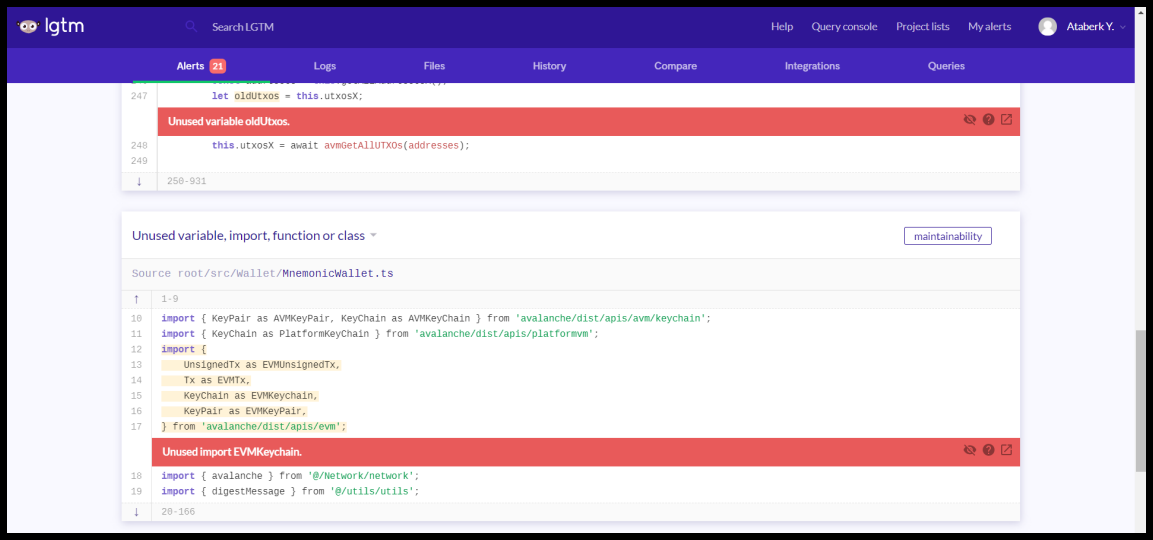


Figure 12: LGTM Scan

Scan results can be accessed via the link below:

Scan Results - LGTM

THANK YOU FOR CHOOSING

// HALBORN