# MystenLabs - Sui Wallet

## WebApp Pentest

Prepared by: **Halborn**

Date of Engagement: **October 21st, 2022 - October 31st, 2022**

Visit: **Halborn.com**

# DOCUMENT REVISION HISTORY

| VERSION | MODIFICATION | DATE | AUTHOR |
|---|---|---|---|
| 0.1 | Document Creation | 10/24/2022 | Chris Meistre |
| 0.2 | Draft Review | 10/31/2022 | Constantin Casmir |
| 0.3 | Draft Review | 10/31/2022 | Gabi Urrutia |
| 0.4 | Document Updates | 11/07/2022 | Chris Meistre |
| 1.0 | Remediation Plan | 05/23/2023 | Alvaro Macias |
| 1.1 | Remediation Plan Review | 05/23/2023 | Carlos Polop |
| 1.2 | Remediation Plan Review | 05/26/2023 | 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 |
| Carlos Polop | Halborn | Carlos.Polop@halborn.com |
| Chris Meistre | Halborn | Chris.Meistre@halborn.com |

# EXECUTIVE OVERVIEW

## 1.1 INTRODUCTION

MystenLabs engaged Halborn to conduct a security audit on the wallet application, beginning on October 21st, 2022 and ending on October 31st, 2022.

## 1.2 AUDIT SUMMARY

The team at Halborn assigned a full-time security engineer to audit the security of the SUI wallet application. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols and web applications security and architecture.

The goals of our security audits are to improve the quality of systems and aim for sufficient remediation to help protect users.

During the penetration testing, Halborn discovered three (3) critical and one (1) high issues. There were also four (4) medium and five (5) low priority issues.

There are issues with the mnemonic and password phrases being stored in clear text in memory. An attacker that has compromised a user's machine will have access to this information and will be able to exfiltrate it.

It was also found that various operations and transactions can be done without requiring additional password authentication. An attacker, with either remote or physical access, would be able to drain the funds or NFTs from a user's wallet that is left unlocked.

The repository underwent a general code audit, and it was found they are well secured in terms of the functionality and implemented security mechanisms.

**In summary, Halborn identified some security risks that were mostly addressed by the MystenLabs team.**

## 1.3 SCOPE

Wallet application:
https://github.com/MystenLabs/sui/tree/main/apps/wallet
https://api.wallet.dev.cere.io

Source code:
MystenLabs/sui

Pull requests:
pull/5632

Remediation commit ID: 8462c7fe4d5c047ba5707b9870b8210ca2ba329a.

## 1.4 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy regarding the scope of the penetration test. While manual testing is recommended to uncover flaws in logic, process and implementation, automated testing techniques assist enhance coverage of the infrastructure and can assist in quickly identifying potential issues.

The following categories of vulnerabilities were evaluated during the audit:

- Mapping Application Content and Functionality.
- Application Logic Flaws.
- Access Handling.
- Authentication/Authorization Flaws.
- Rate Limitation Tests.
- Brute Force Attempts.
- Safe Input Handling.
- Fuzzing of all input parameters.
- Injection (SQL/JSON/HTML/Command).

- Attack surface and publicly available services.
- Static Analysis of security for scoped repository, and imported functions.
- Manual Assessment for discovering security vulnerabilities on codebase.
- Ensuring correctness of the codebase.

## 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. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were 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.

EXECUTIVE OVERVIEW

| CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL |
|---|---|---|---|---|

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

# 2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

| CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL |
|---|---|---|---|---|
| 0 | 4 | 4 | 5 | 0 |

EXECUTIVE OVERVIEW

# EXECUTIVE OVERVIEW

| SECURITY ANALYSIS | RISK LEVEL | REMEDIATION DATE |
|---|---|---|
| UNENCRYPTED MNEMONIC PHRASE IN-MEMORY (DEMONIC) | High | FUTURE RELEASE |
| CONFIDENTIAL DATA ACCESSIBLE ON THE CLIPBOARD | High | SOLVED - 05/23/2023 |
| UN-ENCRYPTED USER PASSWORD IN-MEMORY | High | RISK ACCEPTED |
| AUTO LOCK FUNCTION NOT WORKING | High | SOLVED - 05/23/2023 |
| WEAK PASSWORD POLICY | Medium | SOLVED - 05/23/2023 |
| WALLET TRANSFERS FUNDS WITHOUT AUTHENTICATION | Medium | SOLVED - 05/23/2023 |
| MNEMONIC STORED IN ONE INPUT FIELD | Medium | SOLVED - 01/03/2023 |
| USER IP DISCLOSURE | Medium | FUTURE RELEASE |
| UN-RESTRICTIVE/UN-SECURE EXTENSION CONTENT-SECURITY-POLICY | Low | SOLVED - 05/23/2023 |
| DEPENDENCIES NOT PINNED TO AN EXACT VERSION | Low | SOLVED - 05/23/2023 |
| LACK OF MNEMONIC PHRASE VERIFICATION | Low | SOLVED - 05/23/2023 |
| PRESENCE OF TO-DO COMMENTS ON THE CODE | Low | RISK ACCEPTED |
| EXTENSION PERMISSIONS NOT REQUIRED | Low | RISK ACCEPTED |

# FINDINGS & TECH DETAILS

## 3.1 (HAL-01) UNENCRYPTED MNEMONIC PHRASE IN-MEMORY (DEMONIC) - HIGH

Description:

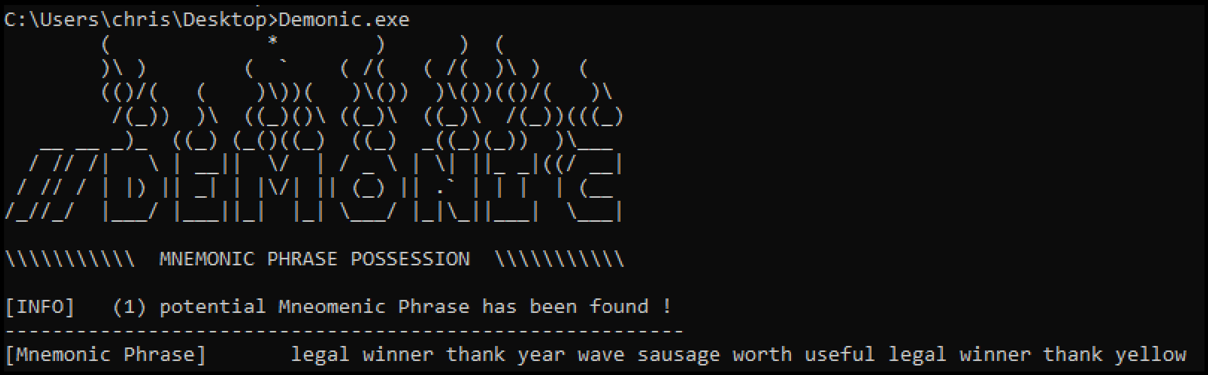
The mnemonic phrase in the wallet is not encrypted in memory. As a result, an attacker who has compromised a user's machine can exfiltrate and steal their mnemonic phrase. It was further found that this mnemonic phrase stays in memory while the application remains open.

This report only contains the vulnerabilities found within the Windows platform. The number of ways to exploit this on Windows is more than on Linux and macOS. If the memory issues are fixed on the Windows platform, they will automatically also cater for those on Linux and macOS.

Proof of concept:

The plain text mnemonic phrase is available in memory during various scenarios. Memory searches were conducted during various stages of the testing.

**Windows - Scenario 1 - Wallet Open**



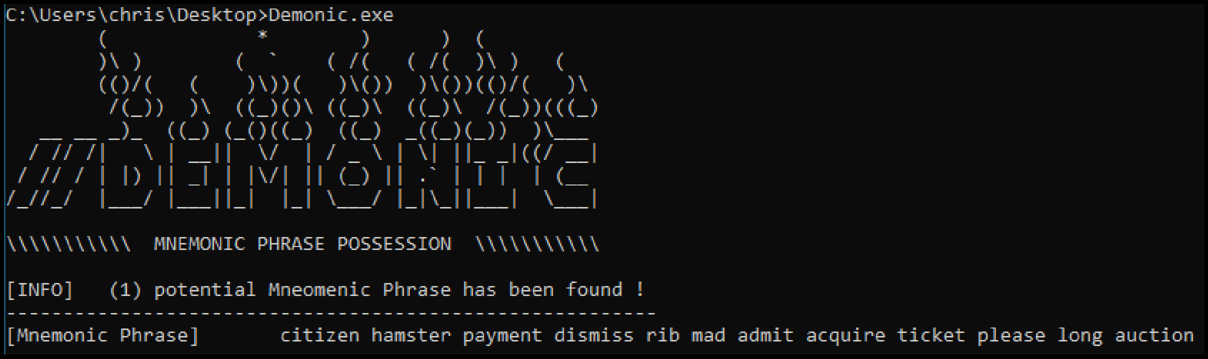
```
C:\Users\chris\Desktop>Demonic.exe
        (                *        )      )  (
        )\ )          (  `     ( /(   ( /(  )\ )    (
       (()/(   (      )\))(    )\())  )\())(()/(    )\
        /(_))  )\   ((_)()\  ((_)\  ((_)\  /(_))(((_)
   __ __ _)_  ((_)  (_()((_)   ((_)  _((_)(_))  )\___
  / // /|   \ | __||  \/  | / _ \ | \| ||_ _|((/ __|
 / // / | |) || _| | |\/| || (_) || .` | | |  | (__
/_//_/  |___/ |___||_|  |_| \___/ |_|\_||___|  \___|

\\\\\\\\\\\  MNEMONIC PHRASE POSSESSION  \\\\\\\\\\\

[INFO]   (1) potential Mneomenic Phrase has been found !
-----------------------------------------------------------
[Mnemonic Phrase]       legal winner thank year wave sausage worth useful legal winner thank yellow
```

Figure 1: Wallet open - clear text mnemonic in memory

**Windows - Scenario 1 - Wallet Locked - After 10 minutes**



```
C:\Users\chris\Desktop>Demonic.exe
          (                *          )       )  (
          )\ )           (  `       ( /(    ( /(  )\ )    (
         (()/(    (      )\))(    )\())   )\())(()/(    )\
          /(_))  )\    ((_)()\  ((_)\   ((_)\  /(_))(((_)
  __ __ _(_))_  ((_)  (_()((_)   ((_)   _((_)(_))  )\___
 / // /|   \ | __||  \/  |  / _ \  | \| ||_ _|((/ __|
/ // / | |) || _| | |\/| | | (_) | | .` | | |  | (__
/_//_/  |___/ |___||_|  |_|  \___/  |_|\_||___|  \___|

\\\\\\\\\\\  MNEMONIC PHRASE POSSESSION  \\\\\\\\\\\

[INFO]   (1) potential Mneomenic Phrase has been found !
---------------------------------------------------------
[Mnemonic Phrase]       citizen hamster payment dismiss rib mad admit acquire ticket please long auction
```

Figure 2: Wallet open - clear text mnemonic in memory

CVSS Vector:

- CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L

Risk Level:

**Likelihood - 4**
**Impact - 4**

Recommendation:

The following recommendation is provided:

- Clear/dereference values of variables which store mnemonic phrases in your code. This will speed up the garbage collector removing the phrase from memory. It is also important to break up the mnemonic phrase into several variables, or obfuscate the original phrase and then dereferencing the variable which used to hold the original phrase. In the cases where you have to handle the mnemonic phrase, you can use the obfuscated variable along with a function that will reconstruct the original mnemonic phrase at the exact point where it is needed.
- Avoid saving the mnemonic phrase on disk, even if it is encrypted. There are cases where you want to identify a wallet, and many times the mnemonic associated with that wallet is retrieved from disk and decrypted. This can result in leakage of the mnemonic in memory.

Instead, you should store and use the entropy to identify wallets.

- Instead of having the user enter their whole phrase, use word selection for mnemonic phrase confirmation on wallet creation.
- Add invisible fake words in-between the mnemonic phrase words when displaying the mnemonic, to hide the phrase in memory.

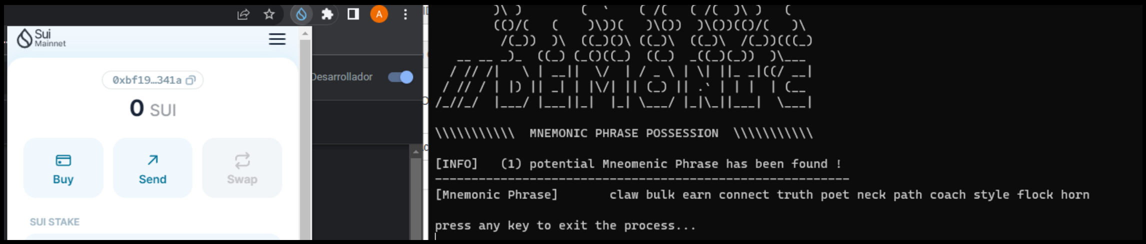
Reference:

Halborn discloses Demonic vulnerability in MetaMask

Remediation Plan:

**PENDING**: The issue would be solved in future release due to their currently task of obfuscate/encrypted the mnemonic.
MystenLabs’s Task:

- When requiring to show mnemonic to the user or private key, request for password, decrypt and show it (many users wanted that to get their mnemonic and import it into another wallet).
- Whenever we need any secret to derive new accounts, let’s have an obfuscation function, never allow keywords of the mnemonic dictionary in memory. There might be generic malware that just scans for a sequence of known words, not necessarily targeting Sui wallet, but any wallet. When you need it, deobfuscate it during the derivation function in a local (to the derivation function) variable.



```
\\\\\\\\\\\  MNEMONIC PHRASE POSSESSION  \\\\\\\\\\\

[INFO]   (1) potential Mneoemenic Phrase has been found !
---------------------------------------------------------
[Mnemonic Phrase]       claw bulk earn connect truth poet neck path coach style flock horn

press any key to exit the process...
```

Figure 3: There is evidence indicating that the issue remains unresolved at present.

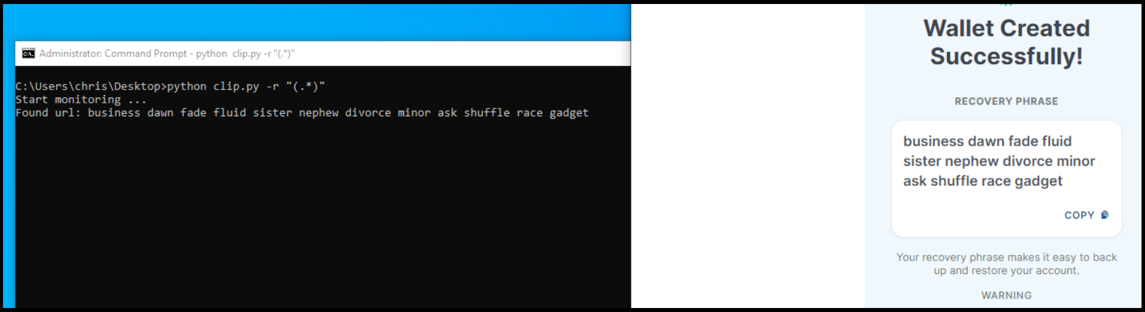
# 3.2 (HAL-02) CONFIDENTIAL DATA ACCESSIBLE ON THE CLIPBOARD - HIGH

## Description:

An attacker can obtain the mnemonic passphrase from the clipboard storage on the host computer. The attack paths should be considered through local and remote access.

A Python script or other process could have access to the clipboard and obtain this sensitive information.

## Proof of concept:



```
C:\Users\chris\Desktop>python clip.py -r "(.*)"
Start monitoring ...
Found url: business dawn fade fluid sister nephew divorce minor ask shuffle race gadget
```

Wallet Created Successfully!

RECOVERY PHRASE

business dawn fade fluid sister nephew divorce minor ask shuffle race gadget

Your recovery phrase makes it easy to back up and restore your account.

WARNING

Figure 4: Clipboard monitoring tool finding the clear text mnemonic

## CVSS Vector:

- CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L

## Risk Level:

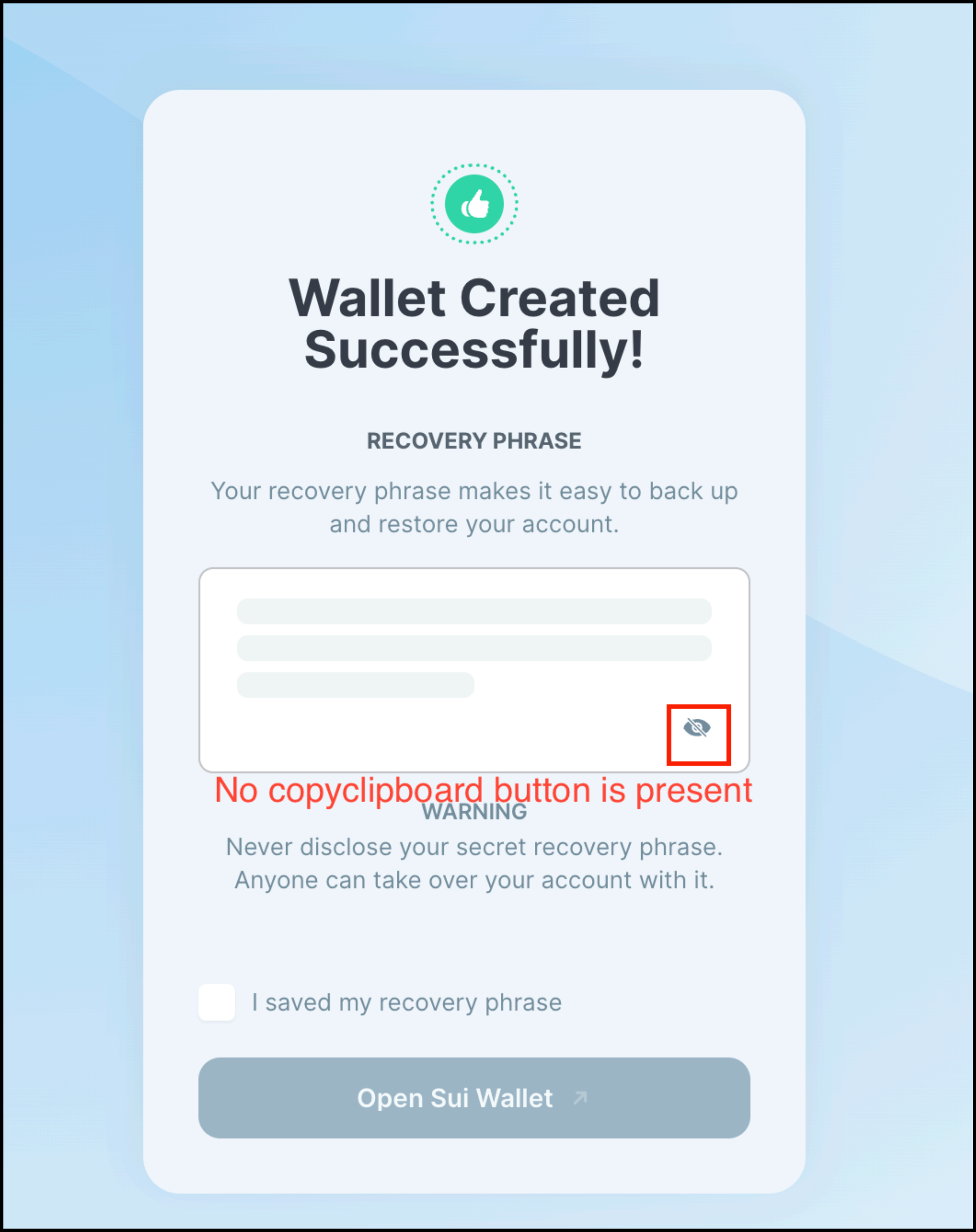
**Likelihood - 4**
**Impact - 4**

Recommendation:

The export of the public and private key should not be done using the clipboard, which could be accessible from other processes. Additionally, disable the copy functionality of the passphrase. The extension should allow only to read the passphrase.

Remediation Plan:

**SOLVED**: The MystenLabs team solved the issue, disabling the copy functionality of the passphrase.



Wallet Created Successfully!

RECOVERY PHRASE

Your recovery phrase makes it easy to back up and restore your account.

No copyclipboard button is present

WARNING

Never disclose your secret recovery phrase. Anyone can take over your account with it.

□ I saved my recovery phrase

Figure 5: No copy button functionality.

## 3.3 (HAL-03) UN-ENCRYPTED USER PASSWORD IN-MEMORY - HIGH

### Description:

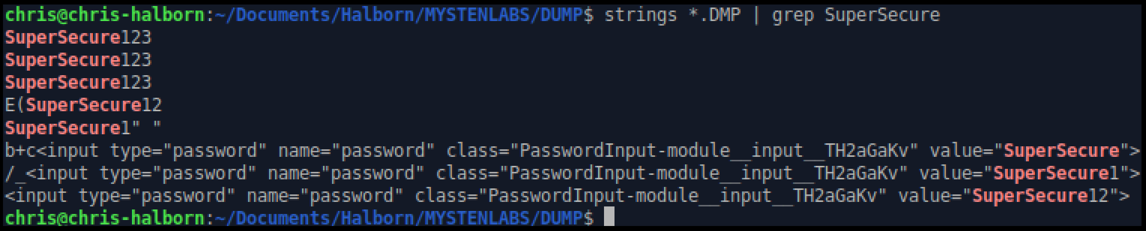
The password in the wallet is not encrypted in memory. As a result, an attacker who has compromised a user machine can exfiltrate and steal the wallet password.

This report only contains the vulnerabilities found within the Windows platform. The number of ways to exploit this on Windows is more than on Linux and macOS. If the memory issues are fixed on the Windows platform, they will automatically also cater for those on Linux and macOS.

### Proof of concept:

The plain text user password is available in memory during various scenarios. Memory dumps were taken throughout the testing process. These memory dumps contained an exact replica of what was in memory while the application was open.

Making use of the strings tool on Linux, a search through the memory dump file revealed the plain text mnemonic phrase.



```
chris@chris-halborn:~/Documents/Halborn/MYSTENLABS/DUMP$ strings *.DMP | grep SuperSecure
SuperSecure123
SuperSecure123
SuperSecure123
E(SuperSecure12
SuperSecure1" "
b+c<input type="password" name="password" class="PasswordInput-module__input__TH2aGaKv" value="SuperSecure">
/_<input type="password" name="password" class="PasswordInput-module__input__TH2aGaKv" value="SuperSecure1">
<input type="password" name="password" class="PasswordInput-module__input__TH2aGaKv" value="SuperSecure12">
chris@chris-halborn:~/Documents/Halborn/MYSTENLABS/DUMP$
```

Figure 6: User wallet password leaked from Chrome memory dump

### CVSS Vector:

- CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L

Risk Level:

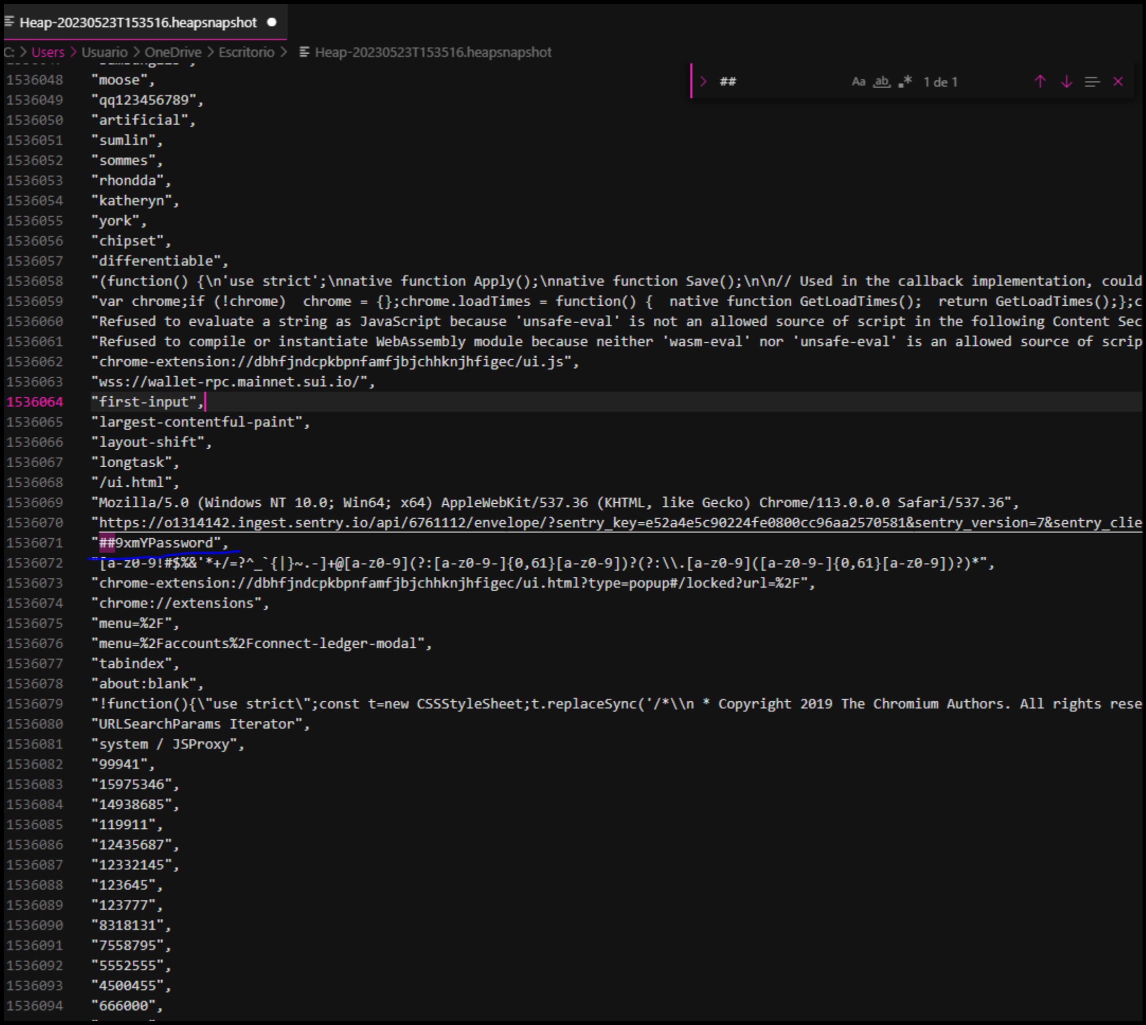
**Likelihood - 4**
**Impact - 4**

Recommendation:

Clear/dereference values of variables which store sensitive information in your code. This will speed up the garbage collector removing the data from memory. In the cases where you have to handle the data, you can use the obfuscated variable along with a function that will reconstruct the original data at the exact point where it is needed.

Remediation Plan:

**RISK ACCEPTED**: The MystenLabs team accepted the risk of unencrypted user password in-memory while the wallet extension is unlocked without indicating further information.


Heap-20230523T153516.heapsnapshot
C: > Users > Usuario > OneDrive > Escritorio > Heap-20230523T153516.heapsnapshot
##
Aa ab .*
1 de 1

```
"moose",
"qq123456789",
"artificial",
"sumlin",
"sommes",
"rhondda",
"katheryn",
"york",
"chipset",
"differentiable",
"(function() {\n'use strict';\nnative function Apply();\nnative function Save();\n\n// Used in the callback implementation, could
"var chrome;if (!chrome)  chrome = {};chrome.loadTimes = function() {  native function GetLoadTimes();  return GetLoadTimes();};c
"Refused to evaluate a string as JavaScript because 'unsafe-eval' is not an allowed source of script in the following Content Sec
"Refused to compile or instantiate WebAssembly module because neither 'wasm-eval' nor 'unsafe-eval' is an allowed source of scrip
"chrome-extension://dbhfjndcpkbpnfamfjbjchhknjhfigec/ui.js",
"wss://wallet-rpc.mainnet.sui.io/",
"first-input",
"largest-contentful-paint",
"layout-shift",
"longtask",
"/ui.html",
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36",
"https://o1314142.ingest.sentry.io/api/6761112/envelope/?sentry_key=e52a4e5c90224fe0800cc96aa2570581&sentry_version=7&sentry_clie
"##9xmYPassword",
"[a-z0-9!#$%&'*+/=?^_`{|}~.-]+@[a-z0-9](?:[a-z0-9-]{0,61}[a-z0-9])?(?:\\.[a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?)*",
"chrome-extension://dbhfjndcpkbpnfamfjbjchhknjhfigec/ui.html?type=popup#/locked?url=%2F",
"chrome://extensions",
"menu=%2F",
"menu=%2Faccounts%2Fconnect-ledger-modal",
"tabindex",
"about:blank",
"!function(){\"use strict\";const t=new CSSStyleSheet;t.replaceSync('/*\\n * Copyright 2019 The Chromium Authors. All rights rese
"URLSearchParams Iterator",
"system / JSProxy",
"99941",
"15975346",
"14938685",
"119911",
"12435687",
"12332145",
"123645",
"123777",
"8318131",
"7558795",
"5552555",
"4500455",
"666000",
```

Figure 7: The user's wallet password was found to be unencrypted while the wallet was in an unlocked state.

## 3.4 (HAL-04) AUTO LOCK FUNCTION NOT WORKING - HIGH

Description:

It was observed that when the wallet is left open in a separate tab (not as the extension popup), the auto-lock function does not function. This meant that the wallet was being left open and unlocked for an indefinite amount of time. An attacker with access to the computer will be able to perform transactions and behalf of this open wallet.

Proof of concept:

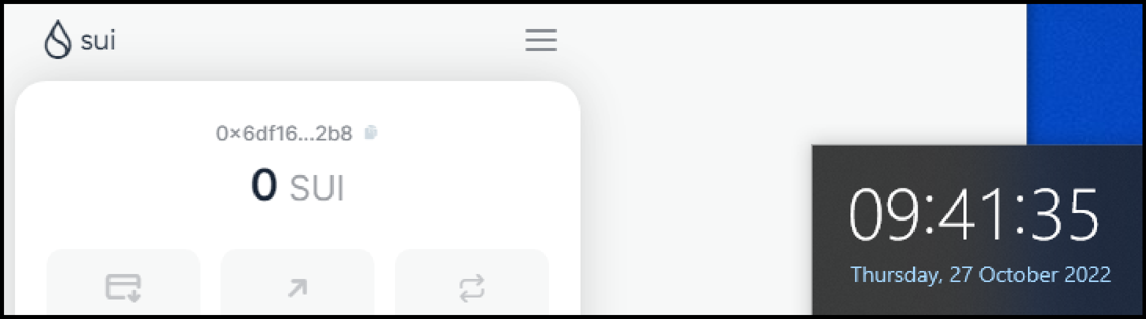


Figure 8: Wallet open

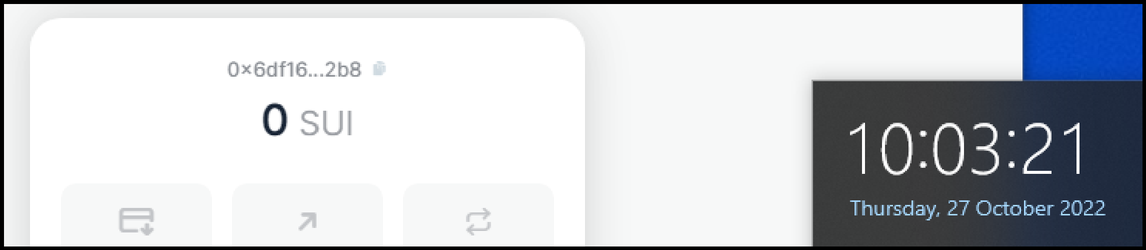


Figure 9: Wallet open - after 22 minutes

CVSS Vector:

- CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L

Risk Level:

**Likelihood - 5**
**Impact - 3**

Recommendation:

It is recommended to make sure the auto-lock function is also enabled when the extension is opened in a browser tab.

Remediation Plan:

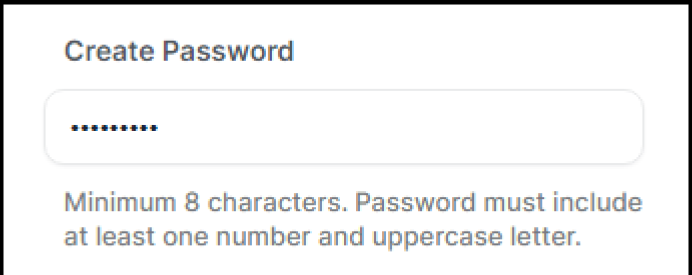
**SOLVED**: The MystenLabs team solved the issue implementing auto-lock function properly.

## 3.5 (HAL-05) WEAK PASSWORD POLICY - MEDIUM

Description:

It was observed that the user can set a weak password like Password1 on the wallet extension as it is only checking the length of the password to be at least 8 characters, with one uppercase character and one number.

Screenshot:



Create Password

••••••••••

Minimum 8 characters. Password must include at least one number and uppercase letter.

Figure 10: Weak password can be entered

CVSS Vector:

- CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N

Risk Level:

**Likelihood - 4**
**Impact - 3**

Recommendation:

It is recommended to implement a password strength checker on the wallet extension to check password input strength, which improves the password policy for a user to consider a strong password during wallet creation.

Reference:

Password Strength Checker

Remediation Plan:

**SOLVED**: The MystenLabs team solved the issue by implementing passwordValidation function on validation.ts file with the zxcvbn library imported.

Listing 1

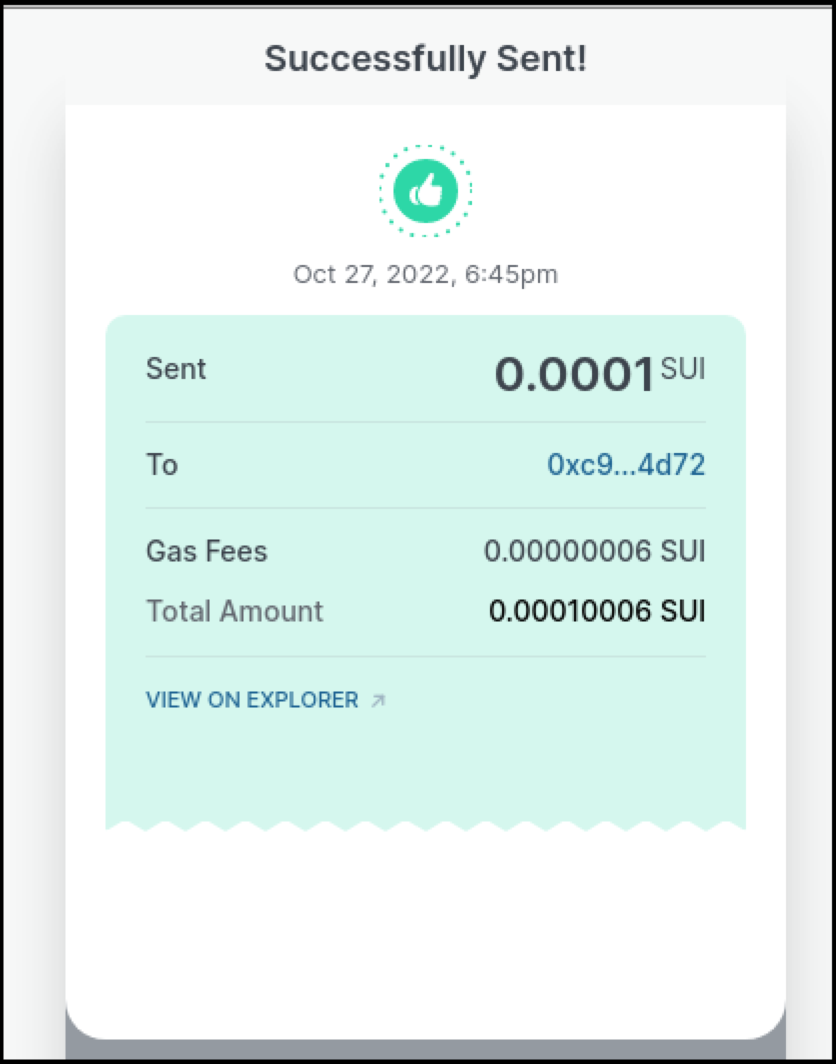
```
export const passwordValidation = Yup.string()
    .ensure()
    .required('Required')
    .test({
        name: 'password-strength',
        test: (password: string) => {
            return zxcvbn(password).score > 2;
        },
        message: ({ value }) => {
            const {
                feedback: { warning, suggestions },
            } = zxcvbn(value);
            return `${addDot(warning) || 'Password is not strong enough.'}${
                suggestions ? ` ${suggestions.join(' ')}` : ''
            }`;
        },
    });
```

## 3.6 (HAL-06) WALLET TRANSFERS FUNDS WITHOUT AUTHENTICATION - MEDIUM

Description:

It was found that an already unlocked wallet allows for fund or NFT transfer from the wallet to another one without any extra authentication. This, with the lack of an auto-locking timer capability, introduces the risk of a malicious user with access to the user’s machine draining the wallet by transferring all the funds or NFTs to an address of theirs.

Screenshots:

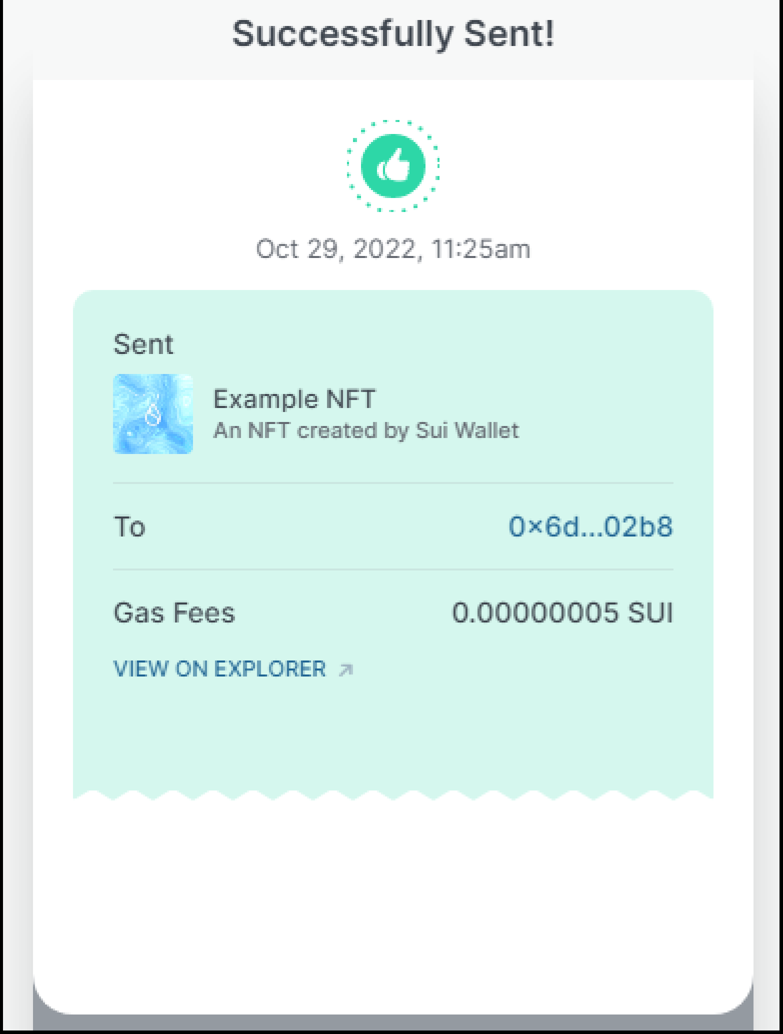


Successfully Sent!

Oct 27, 2022, 6:45pm

| | |
|---|---|
| Sent | 0.0001 SUI |
| To | 0xc9...4d72 |
| Gas Fees | 0.00000006 SUI |
| Total Amount | 0.00010006 SUI |

Figure 11: Transfer without authentication



Successfully Sent!

Oct 29, 2022, 11:25am

Sent

Example NFT
An NFT created by Sui Wallet

| | |
|---|---|
| To | 0x6d...02b8 |
| Gas Fees | 0.00000005 SUI |

Figure 12: Transfer without authentication - NFT

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N

Risk Level:

**Likelihood - 2**
**Impact - 4**

Recommendation:

The wallet should ask the user for a password in order to allow them to transfer funds or NFTs.

Remediation Plan:

**SOLVED**: The MystenLabs team solved the issue by implementing the auto lock mechanism.

## 3.7 (HAL-07) MNEMONIC STORED IN ONE INPUT FIELD - MEDIUM

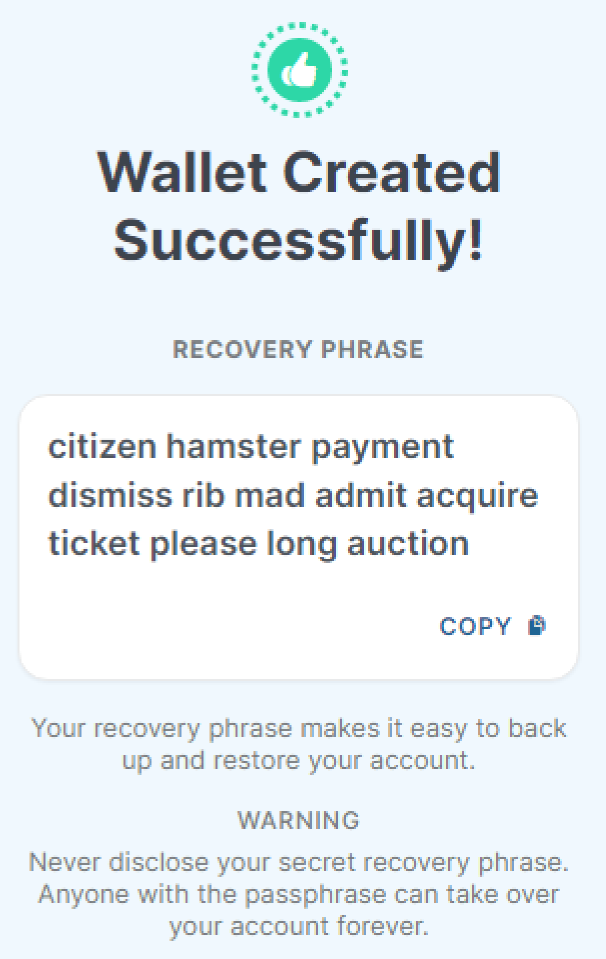
Description:

It was found that when importing a wallet, the mnemonic key is entered by the user into one text field. By having text pasted into one text field, it leaves the application vulnerable to obtaining the clear text mnemonic in memory.

It was found that when the mnemonic that has been created is displayed to the user, it is displayed all in one text field. By having the mnemonic phrase in one text field, it leaves the application vulnerable to obtaining the clear text mnemonic in memory.

With the mnemonic being “grouped” together in one text field, it makes it easier to find it in a memory dump.

Screenshots:



Wallet Created Successfully!

RECOVERY PHRASE

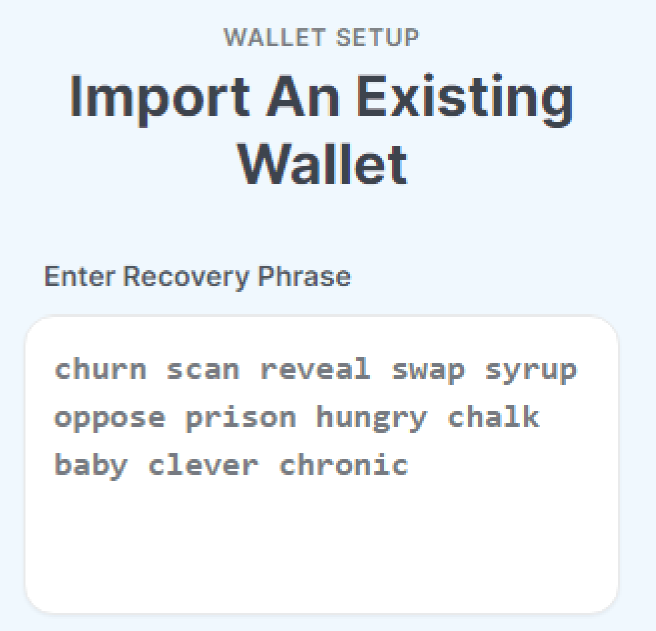
citizen hamster payment dismiss rib mad admit acquire ticket please long auction

Your recovery phrase makes it easy to back up and restore your account.

WARNING

Never disclose your secret recovery phrase. Anyone with the passphrase can take over your account forever.

Figure 13: Clear text mnemonic available in one text field

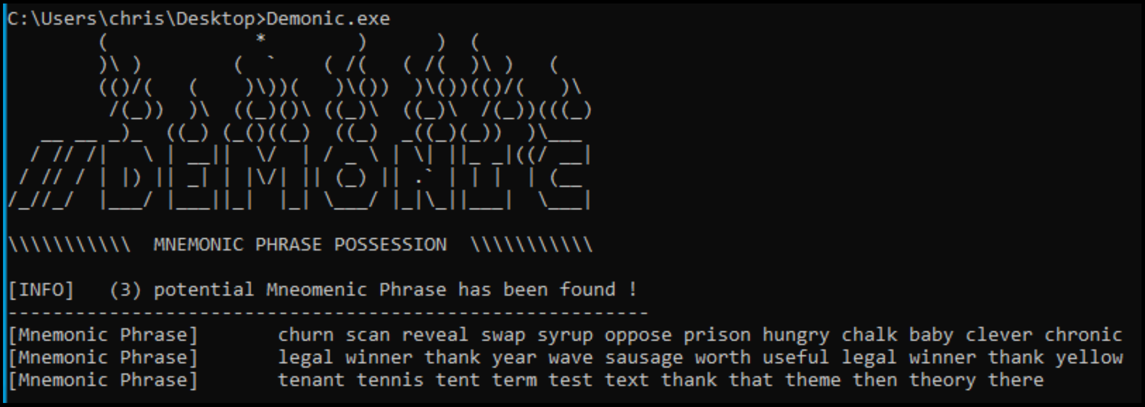


Import An Existing Wallet

Enter Recovery Phrase

churn scan reveal swap syrup oppose prison hungry chalk baby clever chronic

Figure 14: The mnemonic entered into the import text field



```
C:\Users\chris\Desktop>Demonic.exe
        (             *        )      )  (
        )\ )        (  `    ( /(   ( /(  )\ )    (
        (()/(   (    )\))(   )\())  )\())(()/(   )\
         /(_))  )\  ((_)()\ ((_)\  ((_)\  /(_))(((_)
   __ __ _)_  ((_) (_()((_)  ((_)  _((_)(_))  )\___
  / // /|   \ | __||  \/  | / _ \ | \| ||_ _|((/ __|
 / // / | |) || _| | |\/| || (_) || .` | | |  | (__
/_//_/  |___/ |___||_|  |_| \___/ |_|\_||___|  \___|

\\\\\\\\\\\  MNEMONIC PHRASE POSSESSION  \\\\\\\\\\\

[INFO]   (3) potential Mneomenic Phrase has been found !
---------------------------------------------------------
[Mnemonic Phrase]       churn scan reveal swap syrup oppose prison hungry chalk baby clever chronic
[Mnemonic Phrase]       legal winner thank year wave sausage worth useful legal winner thank yellow
[Mnemonic Phrase]       tenant tennis tent term test text thank that theme then theory there
```

Figure 15: Clear text mnemonic available in clear text

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N

Risk Level:

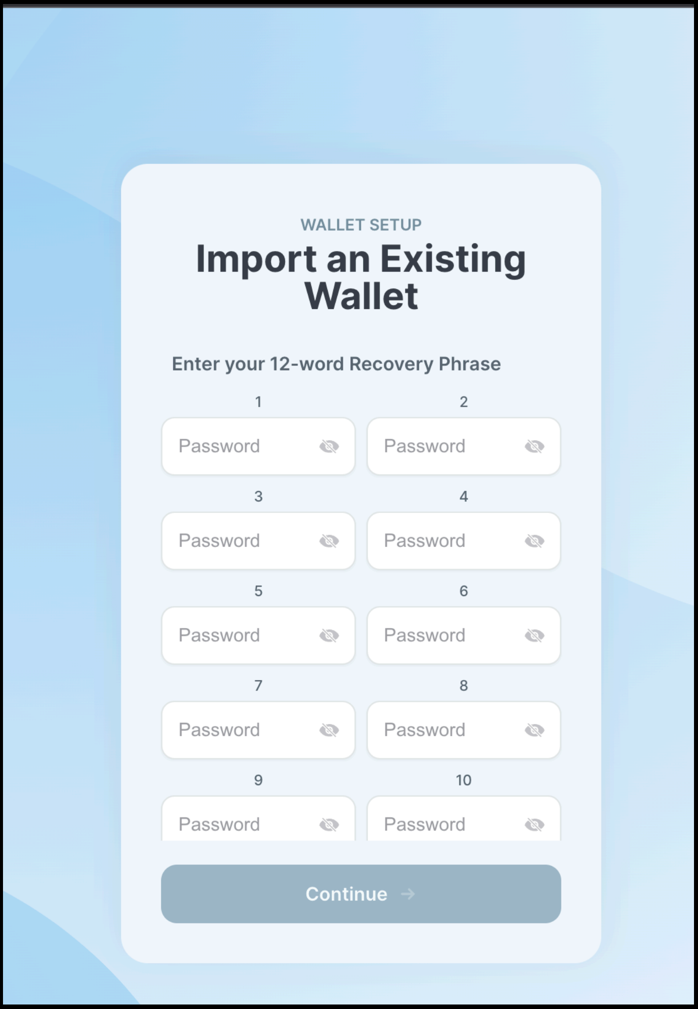
**Likelihood - 2**
**Impact - 4**

Recommendation:

It is recommended that the mnemonic is displayed by making use of multiple text boxes. Further it is recommended that instead of having the user enter their whole phrase, use word selection for mnemonic phrase confirmation on wallet creation.

Remediation Plan:

**SOLVED**: The MystenLabs team solved the issue splitting the recovery phrase field by words.



WALLET SETUP

Import an Existing Wallet

Enter your 12-word Recovery Phrase

| | |
|---|---|
| 1 | 2 |
| Password | Password |
| 3 | 4 |
| Password | Password |
| 5 | 6 |
| Password | Password |
| 7 | 8 |
| Password | Password |
| 9 | 10 |
| Password | Password |

Continue →

## 3.8 (HAL-08) USER IP DISCLOSURE - MEDIUM

Description:

It was found that the wallet allows the user to automatically fetch/show both NFT metadata and NFT images associated with the user's wallet. A highly motivated actor could craft many NFTs, send them to a victim and obtain their IP address, thereby compromising their privacy. This could be because the application is sending a request to retrieve the NFT image directly from the host, thereby exposing its IP address.

If malicious actors obtain more information from the IP address (such as geolocation, GSM carrier, etc.), this can make you a physical risk, such as kidnapping.

Currently, the wallet does not employ any checks to make sure that the NFT's metadata and image property are not under a domain that could be controlled by a malicious actor. This results in attackers being able to gather critical information about the wallet user, which breaches their privacy.

Although not in scope, it was found that the block explorer on https://explorer.devnet.sui.io/ is also vulnerable to the above malicious NFT attack.

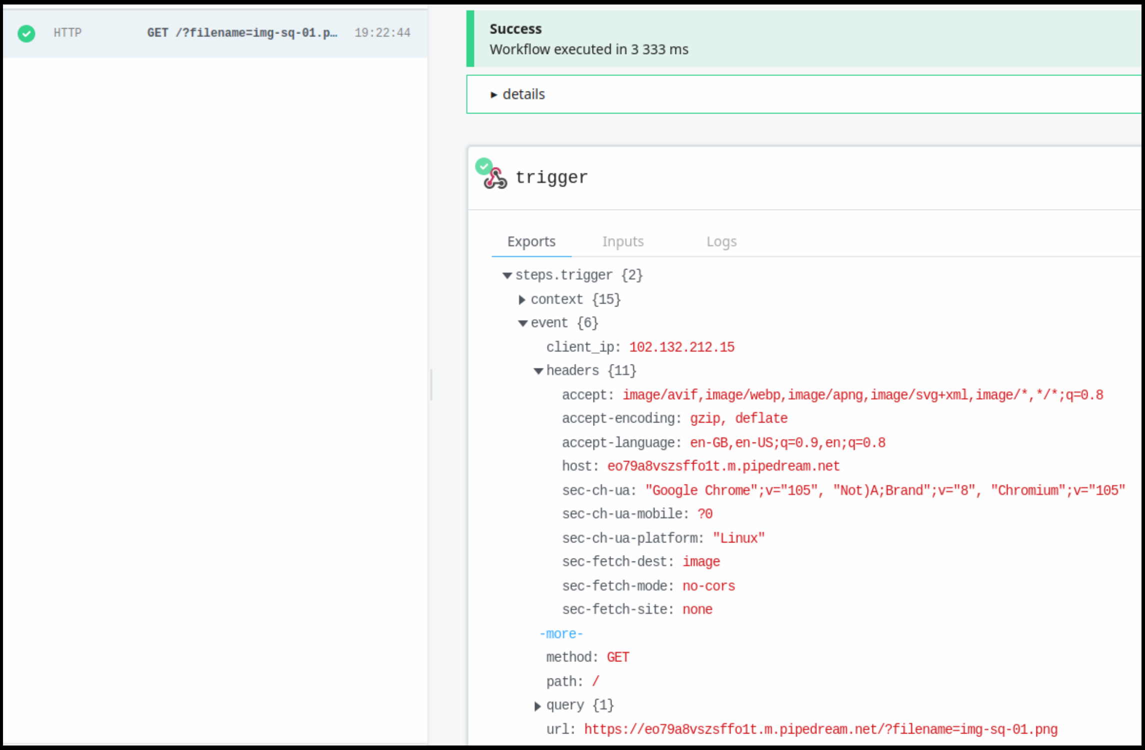
Code Location:

apps/wallet/src/ui/app/components/nft-display/index.tsx, Lines 89-84

Listing 2

```
81                  <img
82                      className={cl(st.img)}
83                      src={filePath}
84                      alt={fileExtentionType?.name || 'NFT'}
85                      title={nftTypeShort}
86                  />
```

### Screenshots:

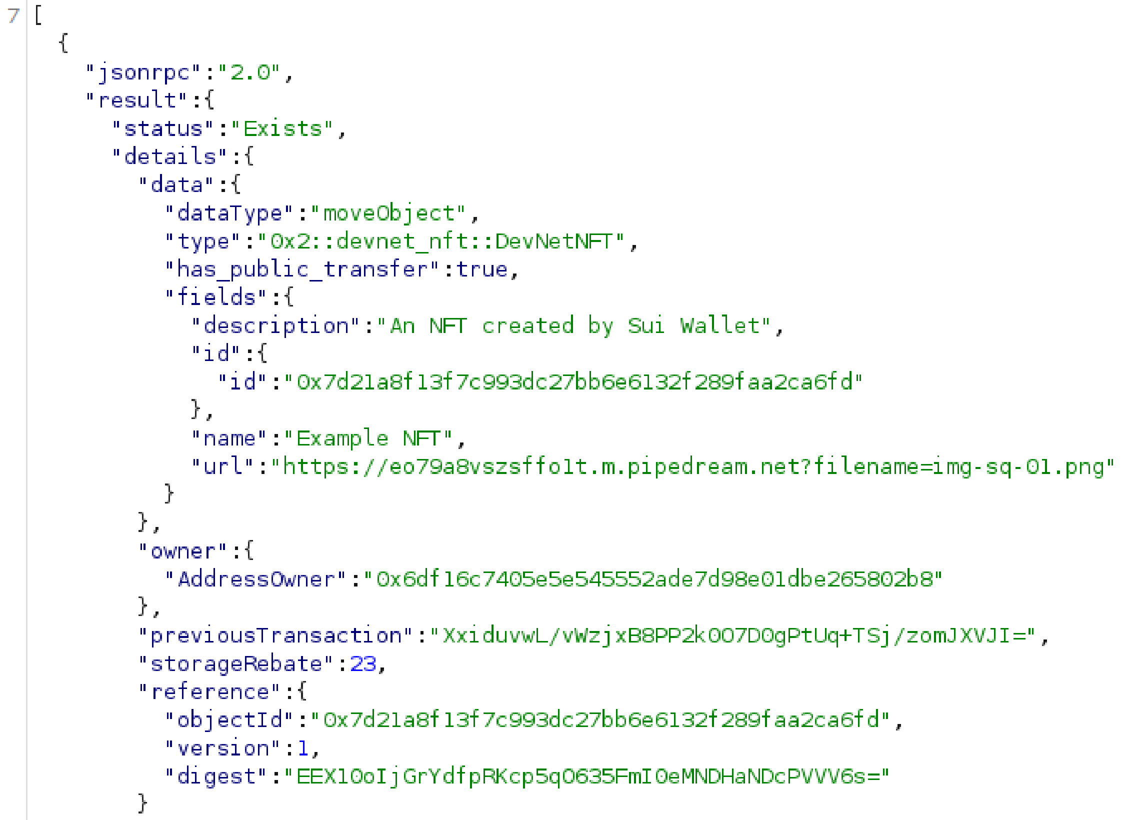


HTTP GET /?filename=img-sq-01.p… 19:22:44

**Success**
Workflow executed in 3 333 ms

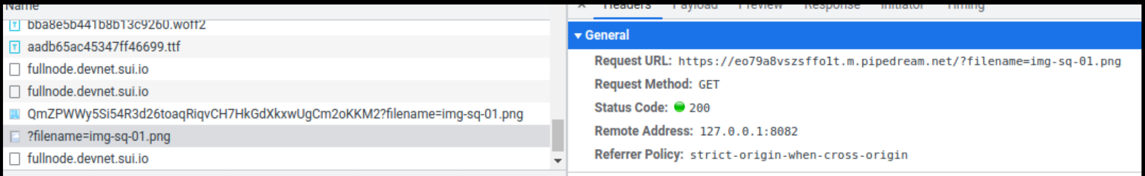
```
▾steps.trigger {2}
  ▸context {15}
  ▾event {6}
     client_ip: 102.132.212.15
    ▾headers {11}
       accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
       accept-encoding: gzip, deflate
       accept-language: en-GB,en-US;q=0.9,en;q=0.8
       host: eo79a8vszsffo1t.m.pipedream.net
       sec-ch-ua: "Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105"
       sec-ch-ua-mobile: ?0
       sec-ch-ua-platform: "Linux"
       sec-fetch-dest: image
       sec-fetch-mode: no-cors
       sec-fetch-site: none
    -more-
     method: GET
     path: /
    ▸query {1}
     url: https://eo79a8vszsffo1t.m.pipedream.net/?filename=img-sq-01.png
```

Figure 16: The attacker’s server receiving HTTP requests from the user’s wallet



```
[
  {
    "jsonrpc":"2.0",
    "result":{
      "status":"Exists",
      "details":{
        "data":{
          "dataType":"moveObject",
          "type":"0x2::devnet_nft::DevNetNFT",
          "has_public_transfer":true,
          "fields":{
            "description":"An NFT created by Sui Wallet",
            "id":{
              "id":"0x7d21a8f13f7c993dc27bb6e6132f289faa2ca6fd"
            },
            "name":"Example NFT",
            "url":"https://eo79a8vszsffolt.m.pipedream.net?filename=img-sq-01.png"
          }
        },
        "owner":{
          "AddressOwner":"0x6df16c7405e5e545552ade7d98e01dbe265802b8"
        },
        "previousTransaction":"XxiduvwL/vWzjxB8PP2k007D0gPtUq+TSj/zomJXVJI=",
        "storageRebate":23,
        "reference":{
          "objectId":"0x7d21a8f13f7c993dc27bb6e6132f289faa2ca6fd",
          "version":1,
          "digest":"EEX10oIjGrYdfpRKcp5q0635FmI0eMNDHaNDcPVVV6s="
        }
```

Figure 17: The NFT's metadata



```
bba8e5b441b8b13c9260.woff2
aadb65ac45347ff46699.ttf
fullnode.devnet.sui.io
fullnode.devnet.sui.io
QmZPWWy5Si54R3d26toaqRiqvCH7HkGdXkxwUgCm2oKKM2?filename=img-sq-01.png
?filename=img-sq-01.png
fullnode.devnet.sui.io
```

Request URL: https://eo79a8vszsffolt.m.pipedream.net/?filename=img-sq-01.png

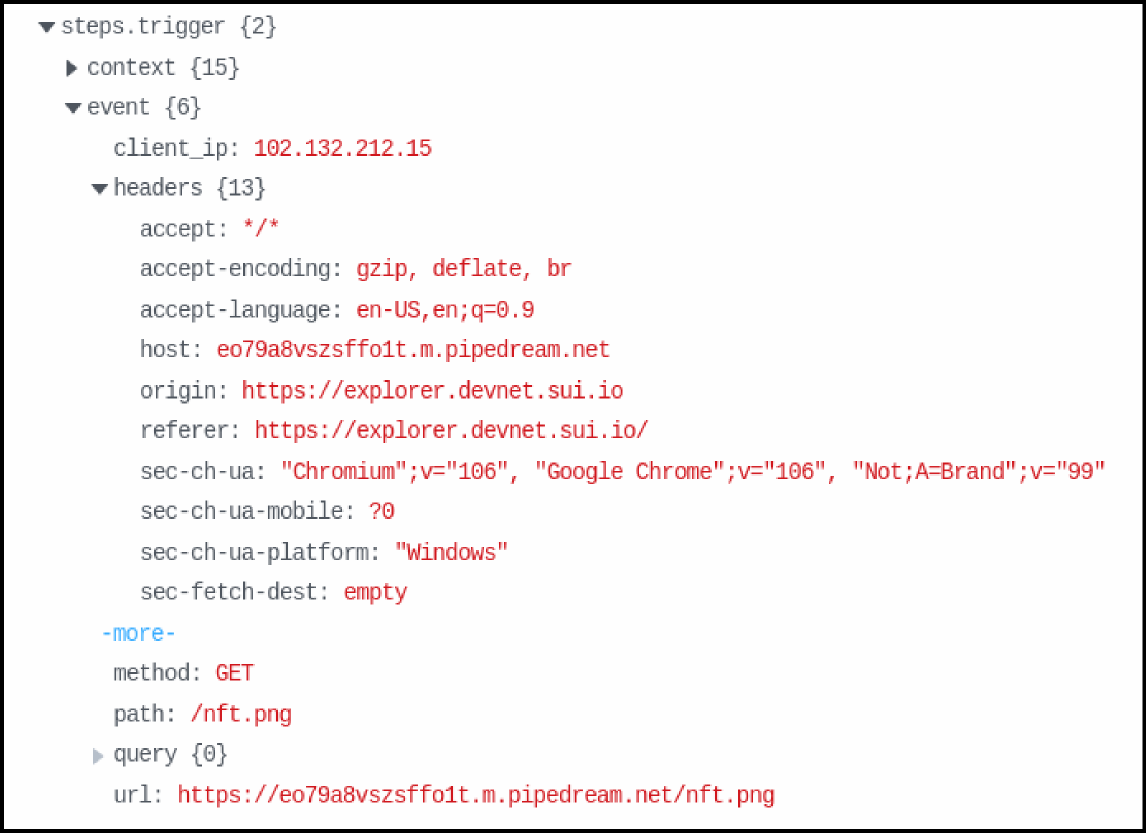
Request Method: GET

Status Code: ● 200

Remote Address: 127.0.0.1:8082

Referrer Policy: strict-origin-when-cross-origin

Figure 18: The wallet sending an HTTP request to the attacker's server to retrieve the metadata and the image



```
▼steps.trigger {2}
  ▶context {15}
  ▼event {6}
     client_ip: 102.132.212.15
    ▼headers {13}
       accept: */*
       accept-encoding: gzip, deflate, br
       accept-language: en-US,en;q=0.9
       host: eo79a8vszsffo1t.m.pipedream.net
       origin: https://explorer.devnet.sui.io
       referer: https://explorer.devnet.sui.io/
       sec-ch-ua: "Chromium";v="106", "Google Chrome";v="106", "Not;A=Brand";v="99"
       sec-ch-ua-mobile: ?0
       sec-ch-ua-platform: "Windows"
       sec-fetch-dest: empty
    -more-
     method: GET
     path: /nft.png
    ▶query {0}
     url: https://eo79a8vszsffo1t.m.pipedream.net/nft.png
```

Figure 19: The attacker's server receiving HTTP requests while viewing the NFT with the block explorer

Example:

https://explorer.devnet.sui.io/objects/0x941e982c73c0d819e6f1674c34370bc5f502de1a

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N

Risk Level:

**Likelihood - 3**
**Impact - 4**

Recommendation:

One of the solutions would be for the wallet application to require an explicit confirmation from the user to access the domain **abc.xyz** when fetching the remote metadata/image from the NFT, informing the user that this may result in an IP address leak. Another option would be to fetch the remote metadata/image in the backend or any kind of network middleware and not in the wallet application. It is also recommended to only allow hosting of NFT metadata on a distributed file system (such as IPFS) and to only allow image hosting on a distributed file system, or being defined by the data:image URI.

Remediation Plan:

**PENDING**: The MystenLabs team has been actively addressing the issue by implementing two distinct approaches in their efforts towards resolving it:

- Only allow loading certain approved NFT collections and manually load (upon approval unknown collections).
- The other long term implementation is use of a proxy server to load images.

# 3.9 (HAL-09) UN-RESTRICTIVE/UN-SECURE EXTENSION CONTENT-SECURITY-POLICY - LOW

Description:

It was found that the extension has a un-restrictive/un-secure declared Content-Security-Policy (CSP).

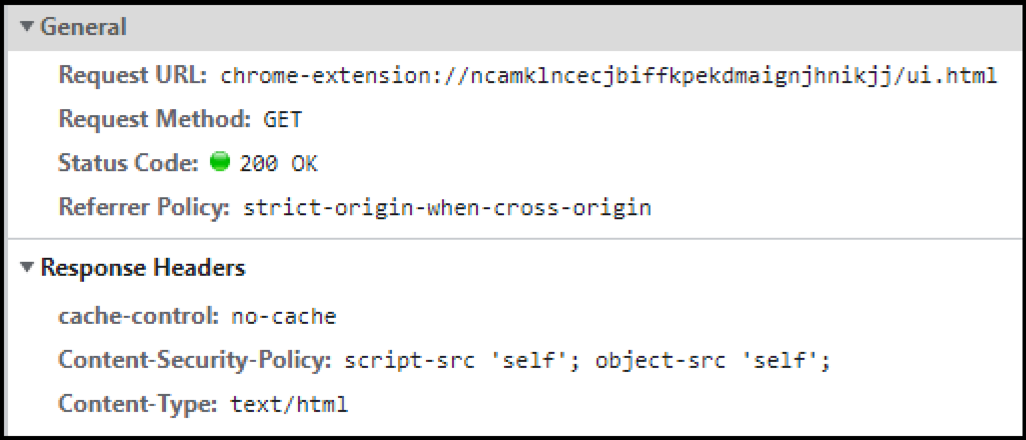
The CSP fails to declare a default-src directive. This is a fallback directive that ensures that no malicious resources are loaded from untrusted sources that have not been declared.

Moreover, elements controlled by object-src are perhaps coincidentally considered legacy HTML elements and aren't receiving new standardized features (such as the security attributes sandbox or allow for <iframe> ). Therefore, it is recommended to restrict this fetch-directive (e.g., explicitly set object-src 'none' if possible).

Lastly, the following directives, which do not inherit from the default-src directive, are not defined:

- base-uri
- form-action
- frame-ancestors
- plugin-types
- report-uri
- sandbox
- reflected-xss
- referrer

Screenshot:



▼ General

Request URL: chrome-extension://ncamklncecjbiffkpekdmaignjhnikjj/ui.html

Request Method: GET

Status Code: ● 200 OK

Referrer Policy: strict-origin-when-cross-origin

▼ Response Headers

cache-control: no-cache

Content-Security-Policy: script-src 'self'; object-src 'self';

Content-Type: text/html

Figure 20: The default CSP the extension is using

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

Risk Level:

**Likelihood - 1**

**Impact - 3**

Recommendation:

It is recommended to add the default-src 'none' directive to your CSP as a fallback. It is also recommended to set the object-src directive to **none** if possible since the elements it controls are considered legacy, and to set the missing directives which do not inherit from the default-src directive.

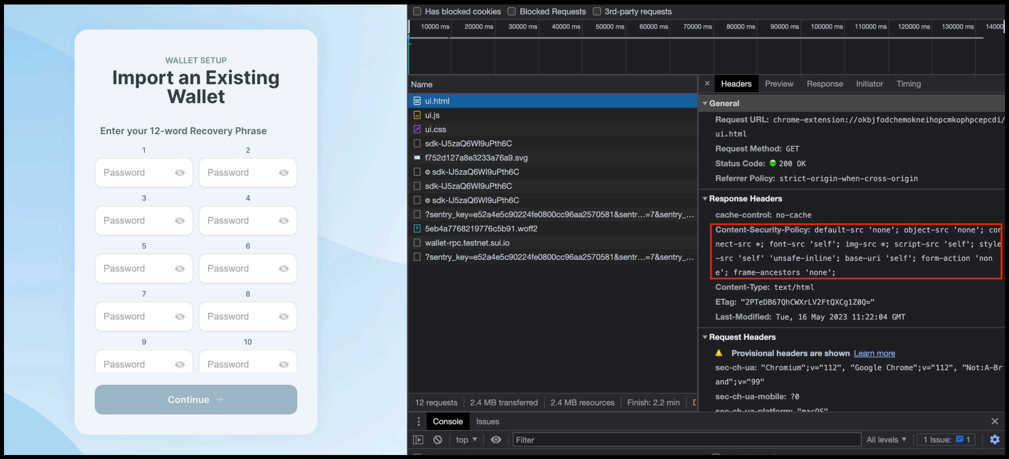
Reference:

Secure Chrome extensions: CSP

object-src directive best practices

## Remediation Plan:

**SOLVED**: The MystenLabs team solved the issue by implementing a properly CSP policy.



```
Name
ui.html
ui.js
ui.css
sdk-lJ5zaQ6WI9uPth6C
f752d127a8e3233a76a9.svg
sdk-lJ5zaQ6WI9uPth6C
sdk-lJ5zaQ6WI9uPth6C
sdk-lJ5zaQ6WI9uPth6C
?sentry_key=e52a4e5c90224fe0800cc96aa2570581&sentr…=7&sentry_…
5eb4a7768219776c5b91.woff2
wallet-rpc.testnet.sui.io
?sentry_key=e52a4e5c90224fe0800cc96aa2570581&sentr…=7&sentry_…

12 requests | 2.4 MB transferred | 2.4 MB resources | Finish: 2.2 min
```

```
▼ General
  Request URL: chrome-extension://okbjfodchemokneihopcmkophpcepcdi/
  ui.html
  Request Method: GET
  Status Code: 200 OK
  Referrer Policy: strict-origin-when-cross-origin
▼ Response Headers
  cache-control: no-cache
  Content-Security-Policy: default-src 'none'; object-src 'none'; cor
  nect-src *; font-src 'self'; img-src *; script-src 'self'; style
  -src 'self' 'unsafe-inline'; base-uri 'self'; form-action 'non
  e'; frame-ancestors 'none';
  Content-Type: text/html
  ETag: "2PTeDB67QhCWXrLV2FtQXCg1Z0Q="
  Last-Modified: Tue, 16 May 2023 11:22:04 GMT
▼ Request Headers
  ⚠ Provisional headers are shown Learn more
  sec-ch-ua: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Br
  and";v="99"
  sec-ch-ua-mobile: ?0
  sec-ch-ua-platform: "macOS"
```

## 3.10 (HAL-10) DEPENDENCIES NOT PINNED TO AN EXACT VERSION - LOW

### Description:

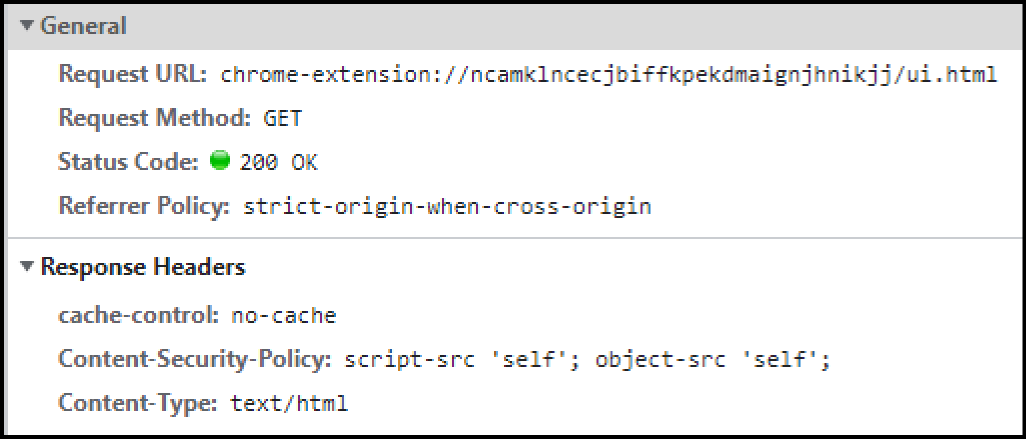
It was found that there are third-party packages that are being used that were not pinned to an exact version but set to compatible version (ˆx.x.x). This could potentially enable dependency attacks, as observed with the event-stream package with the Copay Bitcoin Wallet.

### Code Location:

**Listing 3: package.json**

```
    "dependencies": {
        "@floating-ui/react-dom-interactions": "^0.10.1",
        "@growthbook/growthbook": "^0.18.1",
        "@metamask/browser-passworder": "^3.0.0",
        "@mysten/sui.js": "workspace:*",
        "@mysten/wallet-standard": "workspace:*",
        "@reduxjs/toolkit": "^1.8.3",
        "@scure/bip32": "^1.1.0",
        "@scure/bip39": "^1.1.0",
        "@types/semver": "^7.3.12",
        "bootstrap-icons": "^1.9.1",
        "buffer": "^6.0.3",
        "classnames": "^2.3.1",
        "eslint-plugin-react": "^7.31.8",
        "formik": "^2.2.9",
        "framer-motion": "^7.5.1",
        "mitt": "^3.0.0",
        "react": "^18.2.0",
        "react-dom": "^18.2.0",
        "react-intl": "^6.0.5",
        "react-number-format": "^4.9.3",
        "react-redux": "^8.0.2",
        "react-router-dom": "^6.3.0",
        "react-textarea-autosize": "^8.3.4",
        "rxjs": "^7.5.6",
        "semver": "^7.3.8",
        "stream-browserify": "^3.0.0",
        "tweetnacl": "^1.0.3",
        "uuid": "^8.3.2",
        "webextension-polyfill": "^0.9.0",
        "yup": "^0.32.11"
    }
```

Screenshot:



▼ General

Request URL: chrome-extension://ncamklncecjbiffkpekdmaignjhnikjj/ui.html

Request Method: GET

Status Code: ● 200 OK

Referrer Policy: strict-origin-when-cross-origin

▼ Response Headers

cache-control: no-cache

Content-Security-Policy: script-src 'self'; object-src 'self';

Content-Type: text/html

Figure 21: The default CSP the extension is using

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

Risk Level:

**Likelihood - 2**
**Impact - 2**

Recommendation:

Pinning dependencies to an exact version (=x.x.x) could reduce the possibility of inadvertently introducing a malicious version of a dependency in the future.

Remediation Plan:

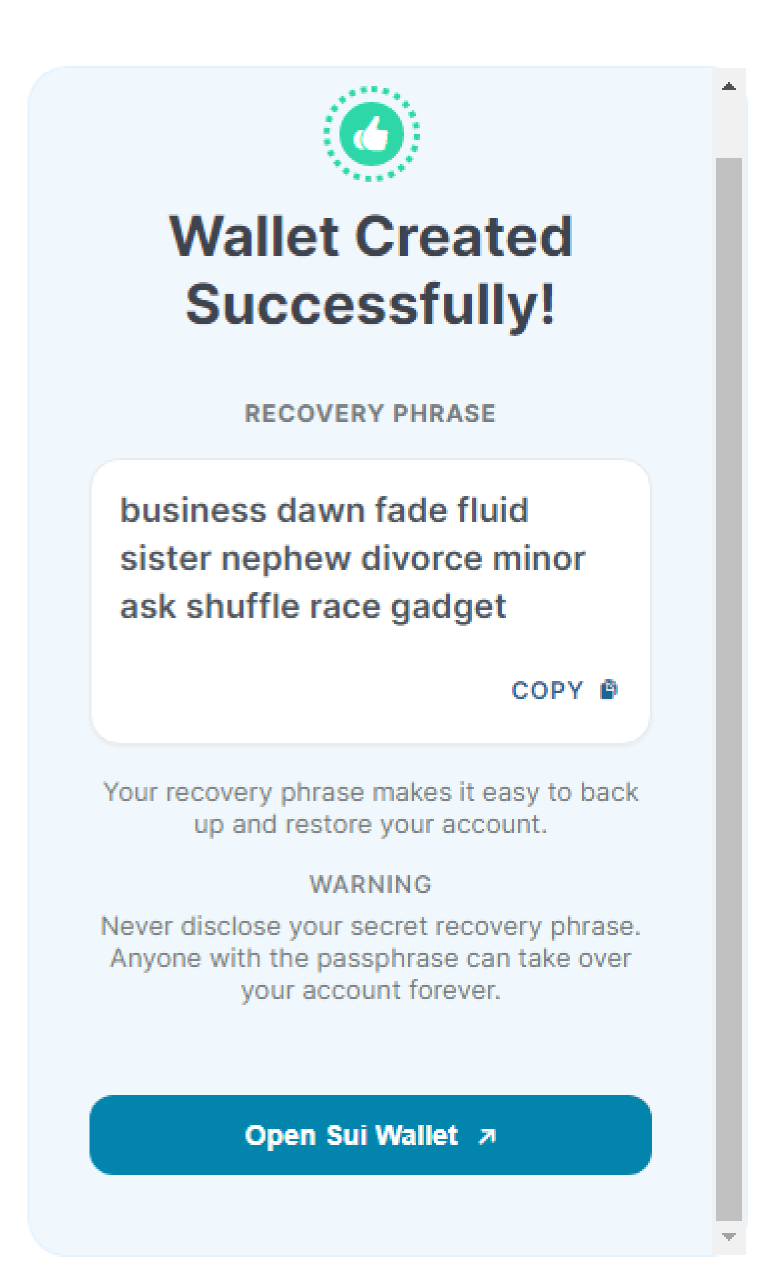
**SOLVED**: The MystenLabs team has indicated that the versions are not pinned in the package.json file, but instead through the pnpm-lock.yaml file that includes exact versions of all dependencies.

## 3.11 (HAL-11) LACK OF MNEMONIC PHRASE VERIFICATION - LOW

Description:

It was found that the wallet did not have any mechanism to verify the provided mnemonic phrase during the wallet creation process after being copied by the user. Lack of this mechanism may pose a significant risk and end up in a fund loss, if the user saves incorrectly the passphrase or forgets to write it down securely.

Screenshot:



Wallet Created Successfully!

RECOVERY PHRASE

business dawn fade fluid sister nephew divorce minor ask shuffle race gadget

COPY

Your recovery phrase makes it easy to back up and restore your account.

WARNING

Never disclose your secret recovery phrase. Anyone with the passphrase can take over your account forever.

Open Sui Wallet ↗

Figure 22: After wallet is created

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

Risk Level:

**Likelihood - 2**
**Impact - 2**

Recommendation:

It is recommended that after wallet creation that a user is forced to complete a challenge to verify that the user saved correctly the mnemonic phrase.

Remediation Plan:

**SOLVED**: The MystenLabs team solved the issue checking the recovery phrase once wallet is created.

## 3.12 (HAL-12) PRESENCE OF TO-DO COMMENTS ON THE CODE - LOW

Description:

Multiple TO-DO comments were found on the code. From the security perspective, the use of these comments does not imply a security risk. However, it could mean that the developed application did not reach an appropriate level of maturity to be on a production environment.

Code Location:

Listing 4: TODO

```
./src/background/Permissions.ts:125:            // TODO: expose those errors to dapp?
./src/background/connections/UiConnection.ts:77:                // TODO: we should depend on a better message to know if app is initialized
./src/shared/messaging/messages/payloads/permissions/Permission.ts:7://TODO: add description, name, tags
./src/shared/messaging/messages/payloads/permissions/Permission.ts:8://TODO add PageLink for instance where the origin and the wallet landing page are different.
./src/dapp-interface/utils.ts:17:                    // TODO: throw proper error
./src/dapp-interface/WalletStandardInterface.ts:41:// TODO: rebuild event interface with Mitt.
./src/dapp-interface/WalletStandardInterface.ts:58:        // TODO: Improve this with ideally a vector logo.
./src/dapp-interface/WalletStandardInterface.ts:64:        // TODO: Extract chain from wallet:
./src/dapp-interface/WalletStandardInterface.ts:122:                   // TODO: Expose public key instead of address:
./src/ui/styles/utils/index.scss:233:        // TODO chanage to mono font
./src/ui/app/staking/selectors.ts:48:    // TODO this is limited only to the active and next set of validators. Is there a way to access the list of all validators?
```

```
./src/ui/app/staking/home/delegation-card/index.tsx:57:
↳           {/* TODO: show the APY of the validator. How can we get
↳  it? */}
./src/ui/app/staking/home/index.tsx:61:
↳                           {/* TODO: show the actual Rewards
↳ Collected value https://github.com/MystenLabs/sui/issues/3605 */}
./src/ui/app/staking/stake/StakeForm.tsx:23:    // TODO(ggao):
↳ remove this if needed
./src/ui/app/staking/stake/StakeForm.tsx:31:    // TODO(ggao):
↳ remove this if needed
./src/ui/app/components/navigation/Navigation.module.scss:125:/*
↳ TODO update icons so they are all the same size */
./src/ui/app/components/active-coins-card/index.tsx:38:
↳                   //TODO: default coin icon switch to on chain
↳ metadata
./src/ui/app/components/filters-tags/index.tsx:17:// TODO: extend
↳ this interface to include params and functions for the filter tags
./src/ui/app/components/explorer-link/Explorer.ts:14:// TODO:
↳ rewrite this
./src/ui/app/components/menu/content/menu-list/index.tsx:57:
↳                   // TODO: import and use the icon from Figma
./src/ui/app/components/transactions-card/index.tsx:52:    // TODO
↳ : update to account for bought, minted, swapped, etc
./src/ui/app/components/sui-apps/DisconnectApp.tsx:49:    // TODO:
↳  add loading state since this is async
./src/ui/app/components/sui-apps/ConnectedAppsCard.tsx:16:
↳ //TODO - move to action
./src/ui/app/components/sui-apps/ConnectedAppsCard.tsx:30:
↳                   //TODO: add a name and descriptions field to
↳ the app data
./src/ui/app/components/sui-apps/index.tsx:75:            //TODO:
↳ add notification on success
./src/ui/app/ApiProvider.ts:76:        // TODO: based on
↳ _featureGating.isOn('deprecate-gateway') value,
./src/ui/app/helpers/formatDate.ts:4:// TODO - handle multiple
↳ date formats
./src/ui/app/hooks/useFileExtentionType.ts:48:// TODO: extend this
↳  list with more file types.
./src/ui/app/redux/slices/txresults/index.ts:74:// TODO: This is a
↳  temporary solution to get the NFT data from Call txn
./src/ui/app/redux/slices/txresults/index.ts:125:
↳                           // TODO handle batch transactions
./src/ui/app/redux/slices/transactions/index.ts:66:        // TODO
↳ : better way to sync latest objects
```

```
./src/ui/app/redux/slices/transactions/index.ts:68:        // TODO
↳ : is this correct? Find a better way to do it
./src/ui/app/redux/slices/transactions/index.ts:99:        // TODO
↳ : fetch the first active validator for now,
./src/ui/app/redux/slices/sui-objects/Coin.ts:30:// TODO use sdk
./src/ui/app/redux/slices/sui-objects/Coin.ts:128:        // TODO:
↳  use PaySui Transaction when it is ready
./src/ui/app/redux/slices/sui-objects/Coin.ts:131:            //
↳ TODO: improve the gas budget estimation
./src/ui/app/redux/slices/sui-objects/NFT.ts:6:// TODO: Remove
↳ this after internal dogfooding
./src/ui/app/redux/slices/sui-objects/NFT.ts:35:    // TODO marge
↳ this method with mintExampleNFT. Import type from @mysten/sui.js
./src/ui/app/redux/slices/app/index.ts:34:// TODO: add clear
↳ Object state because edge cases where use state stays in cache
./src/ui/app/pages/home/transaction-details/index.tsx:43:    //
↳ TODO: load tx if not found locally
./src/ui/app/pages/home/receipt/index.tsx:44:    //TODO: redo the
↳ CTA links
./src/ui/app/pages/home/receipt/index.tsx:66:    //TODO : add more
↳  transfer types and messages
./src/ui/app/pages/home/nft-details/NFTDetails.module.scss:103:/*
↳ TODO use either css or svg so the div can */
./src/ui/app/pages/home/transfer-coin/validation.ts:57:
↳                       // TODO: implement more sophisticated
↳ validation by taking
./src/ui/app/pages/home/transfer-coin/index.tsx:43:// TODO: show
↳ out of sync when sui objects locally might be outdated
```

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

Risk Level:

**Likelihood - 2**
**Impact - 2**

Recommendation:

Review all the comments on the code and ensure that this situation does not affect or offer any risk to the security of the application.

Remediation Plan:

**RISK ACCEPTED**: The MystenLabs team accepted the risk of this issue with the presence of to-do comments on the code.

## 3.13 (HAL-13) EXTENSION PERMISSIONS NOT REQUIRED - LOW

Description:

It was found that the extension requires the tabs permission, but it might not be needed for the extension to be used.

The tabs permission provides access to privileged fields in the Tab object.

Code Location:

apps/wallet/src/manifest/manifest.json, Line 9

Listing 5

```
9     "permissions": ["storage", "tabs", "alarms"],
```

CVSS Vector:

- CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

Risk Level:

**Likelihood - 2**
**Impact - 2**

Recommendation:

It is recommended to review the usage requirement of this permission.

Remediation Plan:

**RISK ACCEPTED**: The MystenLabs team accepted the risk of this issue.

# AUTOMATED TESTING

Description:

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped component. After Halborn verified all the scoped structures in the repository and was able to compile them correctly, these tools were leveraged on scoped structures. With these tools, Halborn can statically verify security related issues across the entire codebase.

## 4.1 semgrep sample

Listing 6

```
  configs/webpack/webpack.config.common.ts
     javascript.lang.security.detect-child-process.detect-child-process
        Detected calls to child_process from a function argument `tsConfigFilePath`. This could lead
        to a command injection if the input is  user controllable. Try to avoid calls to
        child_process, and if it is needed ensure user input is correctly sanitized or sandboxed.
        Details: https://sg.run/l2lo

         31  `${resolve(
         32      PROJECT_ROOT,
         33      'node_modules',
         34      '.bin',
         35      'tsc'
         36 )} -p ${tsConfigFilePath} --showConfig`,

Some files were skipped or only partially analyzed.
  Scan skipped: 4 files larger than 1.0 MB, 29 files matching .semgrepignore patterns
  For a full list of skipped files, run semgrep with the --verbose flag.

Ran 1021 rules on 383 files: 1 finding.
```

## 4.2 yarn audit sample

Listing 7

```
0 vulnerabilities found - Packages audited: 1431
```

## 4.3 njsscan sample

Listing 8

```
njsscan: v0.3.3 | Ajin Abraham | opensecurity.in
No issues found.
```

THANK YOU FOR CHOOSING

// HALBORN