



Star Key

# Web Wallet Audit Report

---

April 26, 2024



# Table of Contents

Table of Contents .....	0
Overview .....	3
Project Details .....	3
Audit Checklist.....	4
Risk Classification .....	5
Finding Classification .....	5
Finding Details.....	6
High .....	6
Private Key and Secret Key Phrase Exposure .....	6
Secret Key Phrase Stored in Significant Text in Memory .....	8
Medium .....	11
In recovery-phrase screen, choice words should be selected randomly, not shuffling.....	11
Use a More Secure Password-Based Key Derivation Mechanism .....	13
Login Password Invalidation is weakness about dictionary attack. ....	15
Low .....	19
The logic of adding the newly added Network name as the token name in customNetwork.tsx is incorrect. ....	19
Cannot go to next step when creating new account in the case that you backed from selecting security type to password setting.....	21
There is no validation for Address Input Field in SearchAddress Component.....	24

In the window of “Import Account”, the input of private should be typed as hidden character so that others cannot read it. ....	25
Info .....	27
In Settings page, links of Feedback and Help don't work. ....	27
In the page of Edit Account, if you reopen edit window and click close after changing avatar, the avatar will be reset as default. ....	28
In Address page, if you input address and then select network, already input address is removed. ....	29
When delete registered address, it would be better to ask confirm. ....	30
Pin Dependencies to Specific Versions .....	31
Make Significant Code Comments .....	32
Make Documentation .....	33
Write valid chain id in shared\data\networks.json.....	34
There is no management for NFTs, so this needs to be added.....	35
After adding new token, it would be better to using loading state when loading the price of added token .....	36

## Overview

This document describes the result of auditing the Star Key Web Wallet project. This project is a web extension that provides the functionalities of managing assets and transactions for several chains.

This project is developed using react library and can be built using webpack as web browser extension.

This auditing is performed against the entire project and built extension's workflow in web browsers including its transactions and interacting with chain.

## Project Details

<b>Project Name</b>	Star Key
<b>Project Type</b>	Web Wallet Extension
<b>Audit Period</b>	April 12 ~ April 19

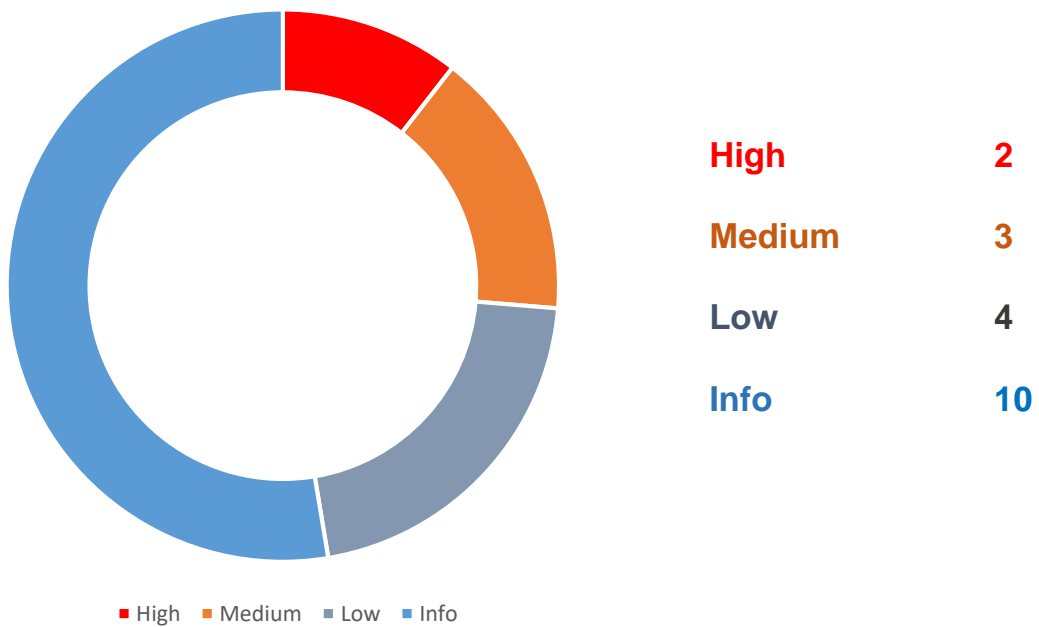
## Audit Checklist

Audit Class	Audit Subclass	Review Status
<b>Transfer Security</b>	Signature	✓
	Deposit / Transfer	✓
	Transaction Broadcast	✓
<b>Secret Key Security</b>	Secret Key Generation	✓
	Secret Key Storage	✓
	Secret Key Usage	✓
	Secret Key Backup	✓
	Secret Key Destruction	✓
	Cryptography Security	✓
<b>Architecture and Business Security</b>	Wallet Lock	✓
	Business Logic	✓
	Architecture Design	✓
	DoS (Denial of Service)	✓
<b>Frontend Security</b>	Cross-Site Scripting	✓
	HTTP Response Header	✓
<b>Components Security</b>	Third-Party Components Security	✓
<b>Communication Security</b>	Communication Encryption	✓
<b>User Interaction Security</b>	Password Complexity Requirements	✓
	Contact Whitelisting	✓
	WYSIWYS	✓

## Risk Classification

		Difficulty		
		High	Medium	Low
Severity	High	High	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Low

## Finding Classification



## Finding Details

### High

#### Private Key and Secret Key Phrase Exposure

##### Severity

High

##### Difficulty

Medium

##### Impact

An attacker can use the secret key phrase or private key to initialize a different extension and take control of all the assets in the wallet.

##### Vulnerability Details

###### *Preconditions*

The victim must be logged into the wallet or show private key while visiting a malicious page. If the victim clicks on a malicious item in the page, the Copy to Clipboard button will be triggered, and the secret key phrase or private key will be available to the attacker in the `navigator.clipboard` object.

###### *Feasibility*

The attack is easy to create, although the user would need to visit the malicious site and perform a single action. Given that an attacker is highly incentivized, they would be motivated to reach as many potential victims as possible.

The Copy to Clipboard button will store the secret key phrase in plaintext in the `navigator.clipboard` object. A malicious site can run a function that checks the contents of this object periodically, and if it recognizes a valid key phrase or private key, it can then exfiltrate the key phrase or private key to a remote location without the victim being aware.

The clickjacking attack facilitates getting the key phrase or private key into the clipboard. As a result, while it can be considered a stage in the overall attack, mitigating clickjacking attacks will not solve the problem of the key phrase being stored in the `navigator.clipboard` object.

## **Recommendation & Mitigations**

Protection against clickjacking attacks is traditionally implemented by utilizing the X-Frame-Options HTTP Header. Since the extension pages are not loaded from an HTTP server, research by the Star Key team can be conducted to assess how this could be implemented in the context of a browser extension. Utilizing Content Security Policies (CSP) that prevent framing resources would be a possible solution.

To protect against the clipboard attack, it would be best to prevent the key phrase or private key from ever being accessible to the clipboard available to the browser. One solution is to disable selection of the text and force users to download a file. Another option would be to investigate if the clipboard object can be disabled for that page entirely. This latter option might prove unfeasible, as users will want to be able to copy and paste addresses.



## Secret Key Phrase Stored in Significant Text in Memory

### Severity

High

### Difficulty

Medium

### Impact

With knowledge of the secret key phrase, an attacker can instantiate a clone of the wallet and gain control of all of its assets.

### Vulnerability Details

#### *Preconditions*

An attacker will need to be able to dump the memory from the extension's process. This would require either physical access to the browser or a post-exploitation condition on the victim's machine.

#### *Feasibility*

The attack is trivial to perform if the preconditions listed above are satisfied. Given the incentive for attackers, common malwares could incorporate a check to dump memory from browsers and search for strings in the format of a secret key phrase for exfiltration.

With the extension open in the browser, open Chrome Developer Tools, select Memory, click "Take Snapshot", and then "Save". With the downloaded file, run the strings command, and for quick results grep for a word in the secret key phrase to view in clear text.

```
TS ShowPrivateKey.tsx TS SecretRecoveryPhrase.tsx Heap-20240425T021744.heapsnapshot TS location.ts alert.png TS PageRouter.tsx TS AddNetwork.tsx
C:\Users\Administrator\Downloads> Heap-20240425T021744.heapsnapshot
1501751 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/fullpage.html#/onboarding/choo
1501752 "/fullpage.html",
1501753 "To add multi-chain accounts, enter or paste your 12 words Secret Recovery Phrase",
1501754 "weapon solid indicate tumble lady crucial dune gain city paddle sniff six",
1501755 "Create account",
1501756 "The account name is only stored locally and will never be shared with Star Key or any third parties",
1501757 "Your account name will become the name of your main account and the prefix of your sub-accounts. You can change the account names at any t:
1501758 "Account Name",
1501759 "Set Password",
1501760 "tabindex",
1501761 "Confirm Password",
1501762 "I agree to the",
1501763 "Terms of Service",
1501764 "Done",
1501765 "z-0 group relative inline-flex items-center justify-center box-border appearance-none select-none whitespace-nowrap subpixel-antialiased on
1501766 "Account name must be at least 3 characters",
1501767 "BlueS",
1501768 "[a-z0-9!#$%&'*+/?^_`{|}~.-]+@[a-z0-9]{1,63}([a-z0-9])?(\.[a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?)?)*",
1501769 "Password must between 9 to 24 characters, one capital letter, one digit, and one special character",
1501770 "Qdfdfdd1995(#",
1501771 "Passwords don\u2019t match",
1501772 "z-0 group relative inline-flex items-center justify-center box-border appearance-none select-none whitespace-nowrap subpixel-antialiased on
1501773 "Set up new password",
1501774 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/scripts/../images/robot.png",
1501775 "Congratulations!",
1501776 "Your wallet has been imported",
1501777 "Open Wallet",
1501778 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/fullpage.html#/",
1501779 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/scripts/../images/Avatar.png",
1501780 "Total Value",
```

## Wallet Password Exposed in Memory

```
TS ShowPrivateKey.tsx TS SecretRecoveryPhrase.tsx Heap-20240425T021744.heapsnapshot TS location.ts alert.png TS PageRouter.tsx TS AddNetwork.tsx
C:\Users\Administrator\Downloads> Heap-20240425T021744.heapsnapshot
1501751 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/fullpage.html#/onboarding/choo
1501752 "/fullpage.html",
1501753 "To add multi-chain accounts, enter or paste your 12 words Secret Recovery Phrase",
1501754 "weapon solid indicate tumble lady crucial dune gain city paddle sniff six",
1501755 "Create account",
1501756 "The account name is only stored locally and will never be shared with Star Key or any third parties",
1501757 "Your account name will become the name of your main account and the prefix of your sub-accounts. You can change the account names at any t:
1501758 "Account Name",
1501759 "Set Password",
1501760 "tabindex",
1501761 "Confirm Password",
1501762 "I agree to the",
1501763 "Terms of Service",
1501764 "Done",
1501765 "z-0 group relative inline-flex items-center justify-center box-border appearance-none select-none whitespace-nowrap subpixel-antialiased on
1501766 "Account name must be at least 3 characters",
1501767 "BlueS",
1501768 "[a-z0-9!#$%&'*+/?^_`{|}~.-]+@[a-z0-9]{1,63}([a-z0-9])?(\.[a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?)?)*",
1501769 "Password must between 9 to 24 characters, one capital letter, one digit, and one special character",
1501770 "Qdfdfdd1995(#",
1501771 "Passwords don\u2019t match",
1501772 "z-0 group relative inline-flex items-center justify-center box-border appearance-none select-none whitespace-nowrap subpixel-antialiased on
1501773 "Set up new password",
1501774 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/scripts/../images/robot.png",
1501775 "Congratulations!",
1501776 "Your wallet has been imported",
1501777 "Open Wallet",
1501778 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/fullpage.html#/",
1501779 "chrome-extension://hcjhpkgbmechpabifbggldplacolbkoh/scripts/../images/Avatar.png",
1501780 "Total Value",
```

## Secret Phrase exposed in Memory

## Recommendation & Mitigations

While removing the key phrase from memory entirely is not possible, it is recommended that the key phrase be encrypted when not immediately in use, and

that neither the encrypted key phrase nor encryption key be stored in a text format that will appear in the output of the strings command. We recommend using a binary object instead. While this would not prevent the key from appearing in a binary search, it would create significant obstacles for an attacker to overcome.

## Medium

In recovery-phrase screen, choice words should be selected randomly, not shuffling.

### Severity

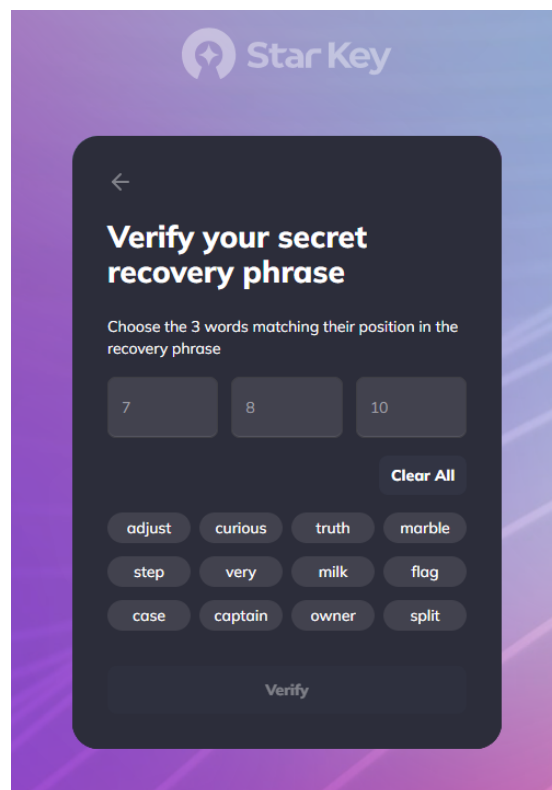
High

### Difficulty

Low

### Impact

In recovery-phrase screen, choice words should be selected randomly, not just shuffling. And whenever clear all, should change order. If you possible, make user should type the words directly in the input.



## Vulnerability Details

When verifying the recovery phrase, they are using all mnemonic as the candidate words just with changing order.

RecoveryPhrase.tsx

```
useEffect(() => {
  if (wallet) {
    // eslint-disable-next-line @typescript-eslint/restrict-plus-operands
    const words = wallet.mnemonic.phrase.split(' ').map((name: any, id: number)
=> ({ id: id + 1, name })))
    // eslint-disable-next-line @typescript-eslint/no-unsafe-argument
    setPhrase(words) // set phrase to get name as per words indexes
    @> setChips(shuffle(words))
  }
}, [wallet])
```

This may harm safety to prevent private information of wallet from being public to the malicious crackers.

## Recommendation & Mitigations

They should use randomly selected words as candidate words by adding real some phrases into them.

In addition, it would be better to select candidates words again whenever click “Clear All” to prevent using several types of automatic bot.

## Use a More Secure Password-Based Key Derivation Mechanism

### Severity

Medium

### Difficulty

Medium

### Impact

In this instance, the Star Key Wallet Extension may accept messages with a correct signature, created using a key under the control of the attacker.

### Vulnerability Details

#### *Preconditions*

The attacker needs to be able to circumvent other security restrictions on cross-tab messaging by the browser.

#### *Feasibility*

Attacks on browsers are not uncommon, but can usually be quickly patched. The feasibility mostly depends on the users update regimen and information from the development team.

It is feasible to significantly speed up dictionary attacks on CPU bound hashes (e.g., using FPGAs) because the task is easily parallelizable. As a result, using CPU-bound key derivation for passwords has been advised against in favor of memory-hard functions like Argon2. These are more difficult to parallelize, because RAM and fast access to it is expensive to build in hardware.

### Recommendation & Mitigations

Argon2 comes in multiple variants optimized for different attack models, but the balanced Argon2id variant would be well suited in this instance. Section 4 of the Argon2 RFC provides a procedure for choosing good parameters for specific use cases. When deciding on the maximum allowed time the derivation is allowed to take place, keeping in mind that it is slower in the browser, there is still a speedup for an attacker who tries to brute-force it using native code.

We recommend making use of key derivation functions based on memory-hard functions such as Argon2, which is a more favorable and secure alternative.

## Login Password Invalidation is weakness about dictionary attack.

### Severity

Medium

### Difficulty

Medium

### Impact

The attacker can guess the password of wallet by dictionary attack.

### Vulnerability Details

#### *Preconditions*

The attacker gains full access to a user's account by brute-forcing an insecure password, which may result in total loss of user funds.

#### *Feasibility*

Easy. Password attacks are increasingly common. Furthermore, it may be that Blank's anonymous withdrawals feature is an additional incentive to conduct such an attack.

In case the attacker accessed victim's pc, the attacker can open wallet by dictionary attack.

In case the attacker have a chrome profile which includes wallet info, then attacker can get **encryptedPrivateKey** from:

```
C:\Users\Administrator\AppData\Local\Google\Chrome\User Data\Profile 1  
\Local Extension Settings\hcjhpkgbmechpabifbggldplacolbkoh\000003.log.
```





When transmitting, the evaluation of gas fee and expected transmission time is incorrect.

## Severity

Medium

## Difficulty

Medium

## Impact

By showing false information to the user regarding token transfer, it makes the user feel rejected towards the wallet.

## Vulnerability Details

Send.tsx

```
useEffect(() => {
  const fetchTransactionCost = async () => {
    try {
      const cost = await estimatedTransactionCost
      setEstimatedEthersFees(cost || 0)
    } catch (error) {
      console.error('Failed to fetch transaction cost:', error)
      setEstimatedEthersFees(0)
    }
  }

  // eslint-disable-next-line @typescript-eslint/no-floating-promises
  fetchTransactionCost()
}, [estimatedTransactionCost])
```

useGas.ts

```
const estimatedTransactionTime = React.useMemo(() => {
  switch (gasOption) {
    case GasOption.Low:
      return '30 seconds'
    case GasOption.Market:
      return '30 seconds'
```

```
    case GasOption.Aggressive:  
      return '15 seconds'  
    }  
  }, [gasOption])
```

As shown in the code, we only get information about the gas fee and block addition time in the Ethereum network.

## **Recommendation & Mitigations**

Depending on which network you are transmitting from, you must obtain and display the exact gas fee and block creation time.

## Low

The logic of adding the newly added Network name as the token name in `customNetwork.tsx` is incorrect.

### Severity

Low

### Difficulty

Low

### Impact

This reduces users' confidence in the system.

### Vulnerability Details

The logic of adding the newly added Network name as the token name in `customNetwork.tsx` is incorrect.

When adding a new network in Wallet, if the function that obtains token information does not obtain the information properly, the name of the token is set to the name of the network, which is incorrect.

`customNetwork.tsx`

```
if (network.name !== 'unknown') {  
  methods.setValue('name', network.name)  
  
  const tokenInfo = await getTokenInfoBySymbol(network.name)  
  if (tokenInfo) {  
    newToken = {  
      ...newToken,  
      image: tokenInfo.image,  
      shortName: tokenInfo.symbol.toUpperCase(),  
      title: tokenInfo.symbol.toUpperCase(),  
      subTitle: tokenInfo.name,  
      coingeckoTokenId: tokenInfo.id,  
    }  
  }  
}
```

```
}  
}  
}
```

Part of the `handleFetchNetworkInfo` function in `customNetwork.tsx` is as follows. As you can see in the code, when a user adds a new Network, token information is retrieved from the Network name. If there is no information about the token, the name is set to the Network name. This means that the name for the token may be incorrect.

### **Mitigations**

When adding a new network, you must create an input field so that you can add the name of the token used when paying fees on the network.

**Cannot go to next step when creating new account in the case that you backed from selecting security type to password setting.**

### **Severity**

Low

### **Difficulty**

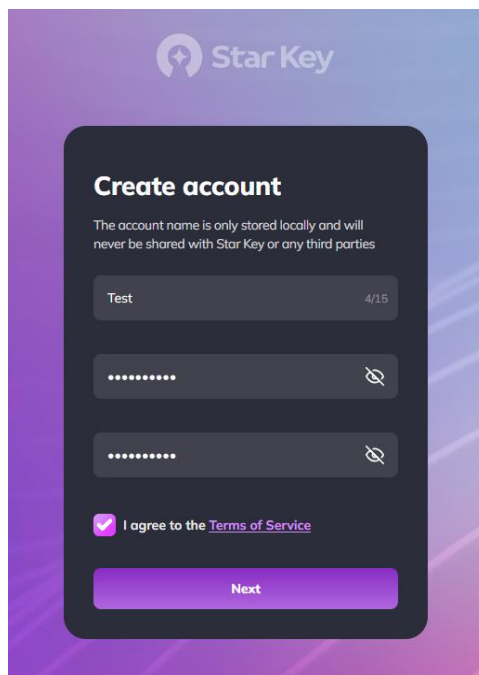
Medium

### **Impact**

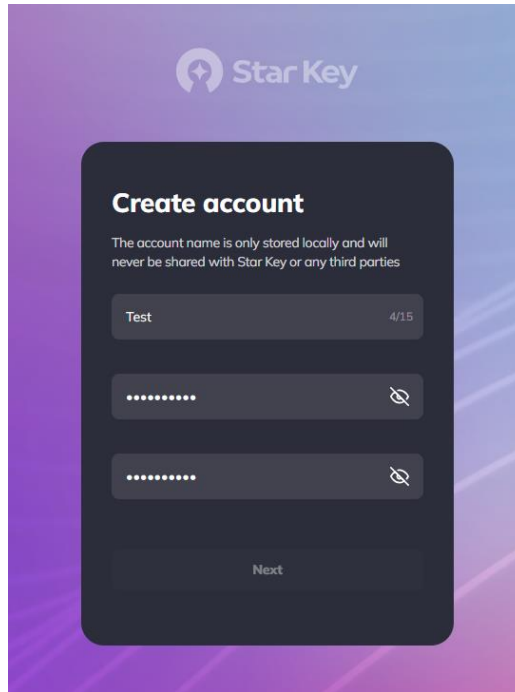
Users cannot go to next step when creating new account in the case that he backed from selecting security.

### **Vulnerability Details**

To create account, users will input their account name, password and confirm password, then check the “Terms of Service” and click the “Next” button.

A screenshot of a mobile application interface for creating a new account. The background is a purple-to-blue gradient. At the top, the 'Star Key' logo is visible. The main form is a dark grey rounded rectangle with the title 'Create account' in white. Below the title, a small note states: 'The account name is only stored locally and will never be shared with Star Key or any third parties'. The form contains three input fields: a text field with 'Test' and a character count '4/15', and two password fields represented by dots. Each password field has an eye icon to toggle visibility. Below the password fields is a checkbox with a checkmark and the text 'I agree to the Terms of Service'. At the bottom of the form is a large purple button labeled 'Next' in white.

After that, when the user click the back button (e.g. to change the account name), then the user cannot click “Next” button again even he input all requirements correctly.



That's because the component of checkbox for “Terms of Service” is hidden in this case.

Therefore, the user should reinstall the wallet again to overcome this problem.

It needs to modify the filtering checkbox component by `onboardingBy` state.

CreateAccount.tsx

```
@> {onboardingBy === '' && (  
  <div className="mt-4 space-y-4">  
    <div className="flex items-center">  
      <Checkbox  
        size="lg"  
        radius="sm"  
        icon={<CheckboxIcon />}  
        onChange={(e) => acceptTermsConditions(e)}  
        isSelected={isTermsConditionsChecked}  
      />  
    </div>  
  </div>  
)}
```

```
        <span className="text-sm font-  
bold">{t('Onboarding.agreefortermsAndConditions')}</span>  
        </Checkbox>  
        &nbsp;{termsAndServicesLink}  
    </div>  
</div>  
    )}
```



**There is no validation for Address Input Field in SearchAddress Component.**

### **Severity**

Low

### **Difficulty**

Low

### **Impact**

Sending a `token` to an invalid address may result in loss of funds.

### **Vulnerability Details**

In the `getWalletAddressRegex` function of `constant.ts`, validation of three types of addresses, "Supra", "SOL", and "default", is performed, so the wallet is designated as a bitcoin network and coins must be transferred to another bitcoin address. In this case, validation cannot proceed.

### **Mitigation**

Validity checks must be performed on different types of addresses.

In the window of “Import Account”, the input of private should be typed as hidden character so that others cannot read it.

### Severity

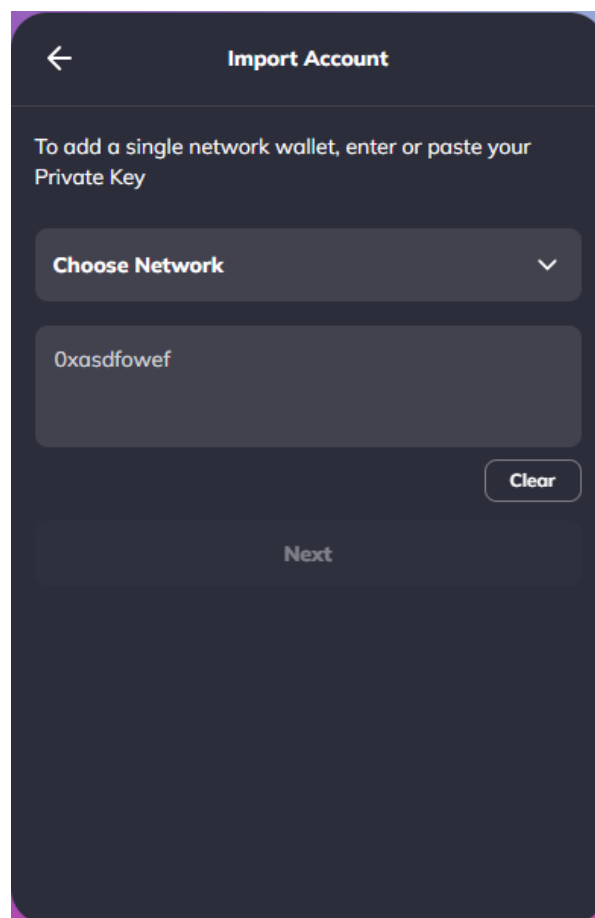
Medium

### Difficulty

Low

### Vulnerability Details

To prevent critical information such as private key to be public, you should hidden its strings so that others cannot read it.



However, in the window of “Import Account”, the typed or copied private key is shown as real strings itself.

It may harm the security of user private information.

## Info

In Settings page, links of Feedback and Help don't work.

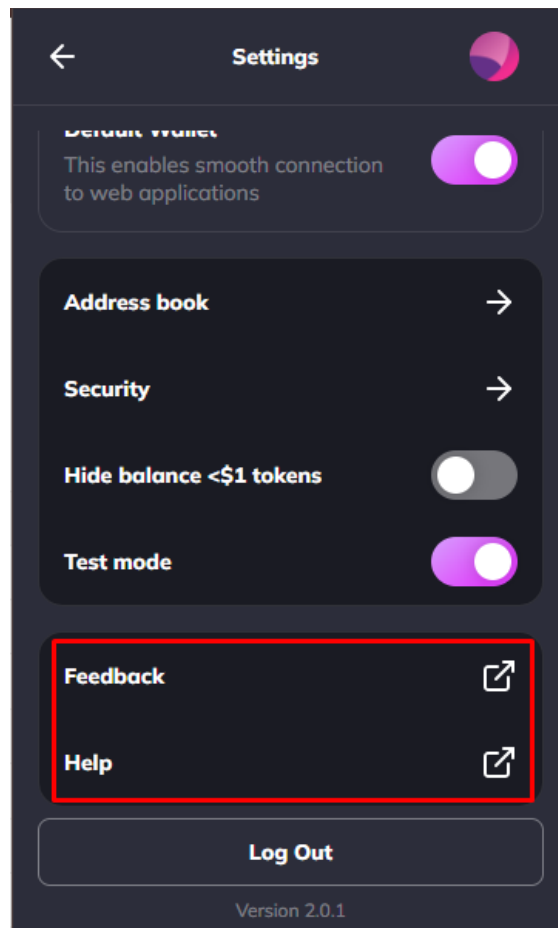
### Severity

Info

### Impact

Users cannot browser contents of feedback and help of this wallet.

### Vulnerability Details



In Settings of the wallet, there are 2 links of “Feedback” and “Help”, but their links don’t work now.

**In the page of Edit Account, if you reopen edit window and click close after changing avatar, the avatar will be reset as default.**

## **Severity**

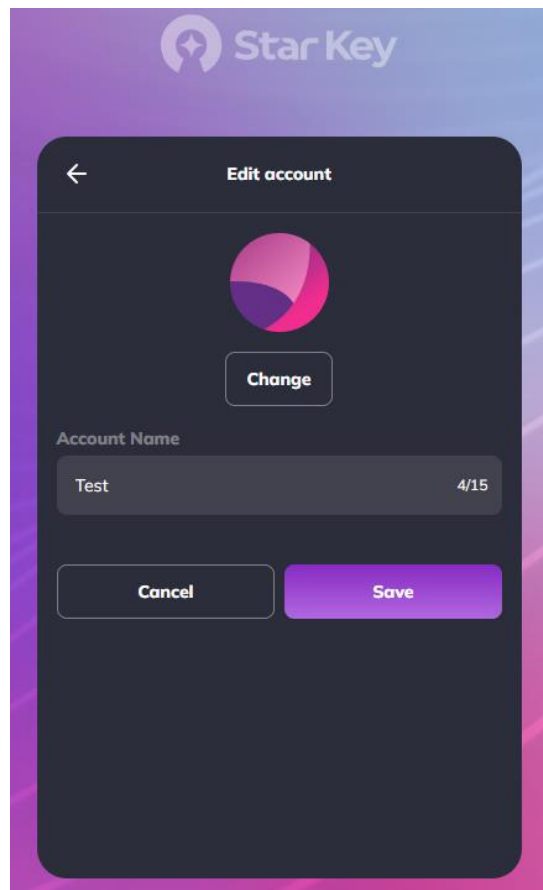
Info

## **Impact**

Users will have issue to change their avatar images of star key wallet.

## **Vulnerability Details**

In the page of “Edit Account”, if you reopen edit window and click close after changing avatar, the avatar will be reset as default.



**In Address page, if you input address and then select network, already input address is removed.**

## Severity

Info

## Impact

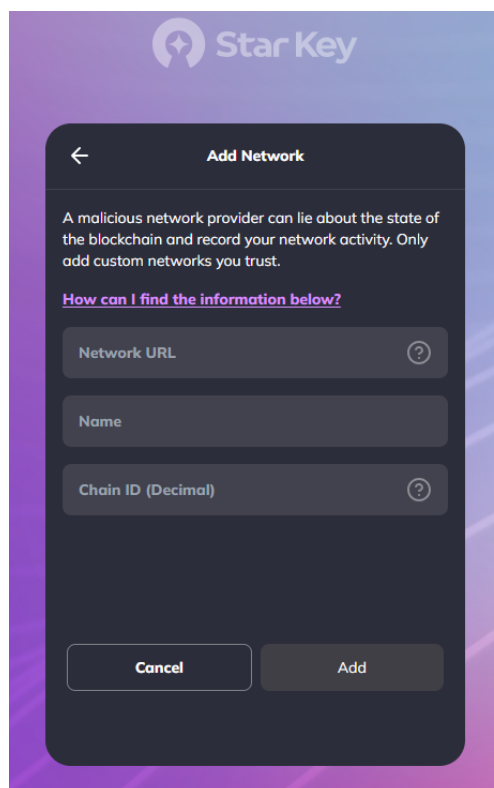
Users will have issue to input and save their contract information.

## Vulnerability Details

When type address before changing chain, the content of typed address will be removed and user should type it again.

It will be uncomfortable for the case that type same address between EVM chains.

To keep user experience, it would be better to check address validation for selected network, then decide whether remove it or not.



**When delete registered address, it would be better to ask confirm.**

## Severity

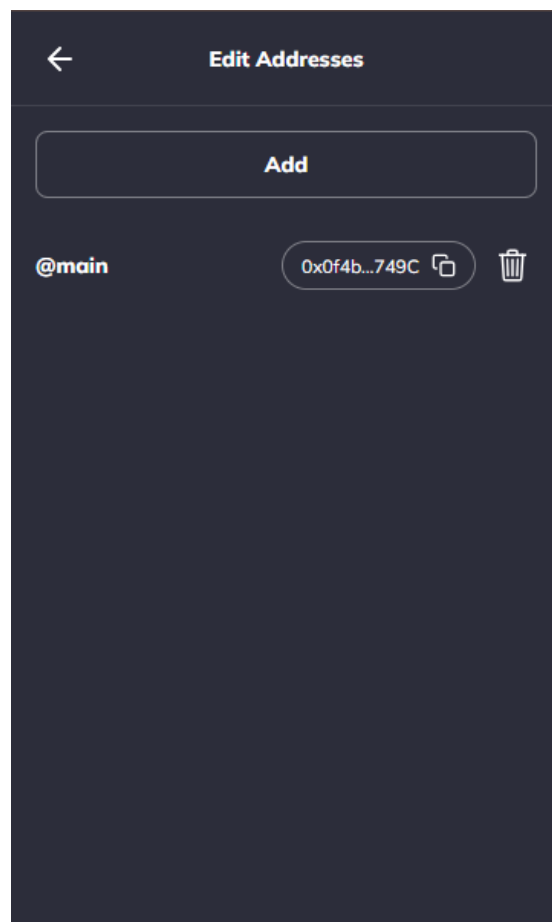
Info

## Impact

It needs to show confirm message when a user delete his address.

## Vulnerability detail

When, a user try to delete the address in his book, it is deleted instantly. To keep user experience, it would be better to ask confirm message that check if user really wants to delete it.



## Pin Dependencies to Specific Versions

### Severity

Info

### Impact

This will prevent unwanted versions from being inadvertently installed, thus minimizing the attack surface.

### Vulnerability Details

Many dependencies are not pinned to a specific version, instead they are pinned to a closed range of releases (e.g. `package.json` is set to accept major and minor changes). Pinning dependencies to exact versions is a sound approach to retaining more control over when and where upgrades take place. This will prevent unwanted versions from being inadvertently installed, thus minimizing the attack surface. In addition, this will help both developers and reviewers to identify new vulnerabilities discovered in dependencies, which is paramount to the security of the codebase.

Upgrading pinned versions of dependencies in accordance with an internal review to assess whether an upgraded version introduces compatibility issues adheres to best practices. In doing so, the necessary changes can be made to the Stacks Wallet Extension code base to mitigate against potential and existing issues. In addition, it's critical to check if the upgraded version of the dependency has reported vulnerabilities. This will allow for informed decisions to use a more secure alternative.



## **Make Significant Code Comments**

### **Severity**

Info

### **Impact**

This will make modifying and maintenance will be easy and raise readability.

### **Vulnerability Details**

All codes have sufficient code comments.

### **Mitigation**

I recommend that code comments coverage is comprehensive and consistent throughout all packages of the code base, as they highlight key information and contribute to easier readability and understanding of the code for both users and checkers.

## **Make Documentation**

### **Severity**

Low

### **Impact**

There is no document so I had a trouble in auditing.

### **Mitigation**

Must make significant documentation for deep audit.

**Write valid chain id in `shared\data\networks.json`**

### **Severity**

Low

### **Impact**

The chain id of SUI was wrong written by 1. Must have to be:

21/103 (Mainnet/Testnet)

The chain id of Solana and Aptos are wrong written. Must have to be correct.

### **Mitigation**

	Mainnet	Testnet
SUI chain id:	21	103
Solana chain id:	900	901
Aptos chain id:	1	2

**There is no management for NFTs, so this needs to be added**

**Severity**

Info

**Impact**

There is no management for NFTs.

**Mitigation**

Should have NFT management interface on code.

**After adding new token, it would be better to using loading state when loading the price of added token**

## Severity

Info

## Impact

While loading latest price of the token after it is added, the price is shown as 0. This may harm user experience in wallet usage.

## Vulnerability Detail

To keep user experience and quality, it would be better to change the price component as loading component while loading latest price of newly added token.

