Audited by

CredShields

Browser Extension Audit

November 26th, 2024 • CONFIDENTIAL

Description

This document details the process and result of the Browser Extension audit performed by CredShields Technologies PTE. LTD. on behalf of Arcana between October 25th, 2024, and November 13th, 2024. A retest was performed on November 18th, 2024.

Author
Shashank (Co-founder, CredShields) shashank@CredShields.com

Reviewers
Aditya Dixit (Research Team Lead), Shreyas Koli(Auditor), Naman Jain (Auditor), Sanket Salavi (Auditor)

Prepared for
Arcana

Table of Contents

1. Executive Summary --------------------

Arcana engaged CredShields to perform a Browser Extension audit from October 25th, 2024, to November 13th, 2024. During this timeframe, 5 vulnerabilities were identified. **A retest was performed on November 18th, 2024, and all the bugs have been addressed.**

During the audit, 0 vulnerabilities were found with a severity rating of either High or Critical. These vulnerabilities represent the greatest immediate risk to "Arcana" and should be prioritized for remediation, and fortunately, none were found.

The table below shows the in-scope assets and a breakdown of findings by severity per asset. Section 2.3 contains more information on how severity is calculated.

Assets in Scope	Critical	High	Medium	Low	info	Σ
Arcana Wallet Browser Extension	0	0	1	4	0	**5**
	0	**0**	**1**	**4**	**0**	**5**

Table: Vulnerabilities Per Asset in Scope

The CredShields team conducted the security audit to focus on identifying vulnerabilities in the Arcana Wallet Browser Extension’s scope during the testing window while abiding by the policies set forth by Arcana’s team.

State of Security

To maintain a robust security posture, it is essential to continuously review and improve upon current security processes. Utilizing CredShields' continuous audit feature allows both Arcana's internal security and development teams to not only identify specific vulnerabilities but also gain a deeper understanding of the current security threat landscape.

To ensure that vulnerabilities are not introduced when new features are added, or code is refactored, we recommend conducting regular security assessments. Additionally, by analyzing the root cause of resolved vulnerabilities, the internal teams at Arcana can implement both manual and automated procedures to eliminate entire classes of vulnerabilities in the future. By taking a proactive approach, Arcana can future-proof its security posture and protect its assets.

2. The Methodology -------------------

Arcana engaged CredShields to perform an audit for the Arcana Wallet Browser Extension Browser Extension. The following sections cover how the engagement was put together and executed.

2.1 Preparation Phase

The CredShields team meticulously reviewed all provided documents and analyzed the browser extension's source code to gain a comprehensive understanding of its features and functionalities. They carefully examined all key components, including scripts, manifest files, and associated APIs, to systematically identify potential security vulnerabilities. The analysis prioritized issues with significant impact on user data, privacy, and overall extension security. To confirm their findings, the team conducted a hands-on assessment by installing the browser extension in a controlled environment and performing rigorous testing and validations throughout the audit phase.

A testing window from October 25th, 2024, to November 13th, 2024, was agreed upon during the preparation phase.

2.1.1 Scope

During the preparation phase, the following scope for the engagement was agreed upon:

IN SCOPE ASSETS
https://github.com/arcana-network/ca-wallet

2.1.2 Documentation

Documentation was not required as the application and the source code were self-sufficient for understanding the project.

2.1.3 Audit Goals

CredShields employs a combination of proprietary automated tools and manual techniques for a comprehensive security assessment of browser extensions. The majority of the audit focuses on manually reviewing the extension's source code, adhering to OWASP ASVS (Application Security Verification Standard) guidelines and an extended, self-developed checklist tailored for browser extensions. The team emphasizes understanding the extension's core functionalities, preparing test scenarios, and analyzing its business logic to uncover potential vulnerabilities.

2.2 Retesting Phase

Arcana is actively partnering with CredShields to validate the remediations implemented towards the discovered vulnerabilities.

2.3 Vulnerability classification and severity

CredShields follows OWASP's Risk Rating Methodology to determine the risk associated with discovered vulnerabilities. This approach considers two factors - Likelihood and Impact - which are evaluated with three possible values - **Low**, **Medium**, and **High**, based on factors such as Threat agents, Vulnerability factors, and Technical and Business Impacts. The overall severity of the risk is calculated by combining the likelihood and impact estimates.

Overall Risk Severity				
Impact	HIGH	● Medium	● High	● Critical
	MEDIUM	● Low	● Medium	● High
	LOW	● None	● Low	● Medium
		LOW	MEDIUM	HIGH
Likelihood				

Overall, the categories can be defined as described below -

1. Informational

We prioritize technical excellence and pay attention to detail in coding practices. Our guidelines, standards, and best practices help ensure software stability and reliability. Informational vulnerabilities are opportunities for improvement and do not pose a direct risk to the extension or its users. Code maintainers should use their own judgment on whether to address them.

2. Low

Low-risk vulnerabilities have minimal impact or require specific circumstances to be exploited, often without repeated success. Alternatively, they may be issues the client considers insignificant based on their business needs.

3. Medium

Medium-severity vulnerabilities arise from weak or flawed logic in the extension code or configurations. They may lead to unauthorized access to or modification of non-sensitive user data. While not immediately critical, these issues can harm the client's reputation under certain conditions and should be resolved within an agreed-upon timeframe.

4. **High**

   High-severity vulnerabilities present significant risks to the browser extension and its users. These issues may enable attackers to access sensitive user data, inject malicious code, or disrupt functionality. Such vulnerabilities can cause reputational damage and financial loss, requiring prompt remediation.

5. **Critical**

   Critical issues represent directly exploitable vulnerabilities that do not require specific conditions. These may result in unauthorized access to or theft of sensitive user data, compromise of user accounts, or complete extension and server takeovers. Such vulnerabilities can severely impact the client's reputation and user trust and must be addressed immediately.

2.4 CredShields staff

The following individual at CredShields managed this engagement and produced this report:

- Shashank, Co-founder CredShields (shashank@CredShields.com)

Please feel free to contact this individual with any questions or concerns you have about the engagement or this document.

3. Findings Summary --------------------

This chapter contains the results of the security assessment. Findings are categorized by their severity and grouped based on the affected component and OWASP classification. Each component section includes a detailed summary. The table in the executive summary outlines the total number of identified security vulnerabilities per component, classified by their risk level.

3.1 Findings Overview

3.1.1 Vulnerability Summary

During the security assessment, 5 security vulnerabilities were identified in the asset.

VULNERABILITY TITLE	SEVERITY	CWE | Vulnerability Type
Partial Denial of Service through Contacts	Medium	Denial of Service
Weak Vault Passwords	Low	CWE-521: Weak Password Requirements
Missing Validation in Token URL	Low	CWE-20: Improper Input Validation
Using Insecure HTTP URLs in Token URL	Low	CWE-319: Cleartext Transmission of Sensitive Information
Using Insecure V2 Manifest for Firefox	Low	CWE-16: Misconfiguration

Table: Findings in Smart Contracts

4. Remediation Status ------------------

Arcana is actively partnering with CredShields from this engagement to validate the remediation of the discovered vulnerabilities.

A retest was performed on November 18th, 2024, and all the issues have been addressed.

Also, the table shows the remediation status of each finding.

VULNERABILITY TITLE	SEVERITY	REMEDIATION STATUS
Partial Denial of Service through Contacts	Medium	**Fixed** [Nov 18, 2024]
Weak Vault Passwords	Low	**Fixed** [Nov 18, 2024]
Missing Validation in Token URL	Low	**Fixed** [Nov 18, 2024]
Using Insecure HTTP URLs in Token URL	Low	**Fixed** [Nov 18, 2024]
Using Insecure V2 Manifest for Firefox	Low	**Fixed** [Nov 18, 2024]

Table: Summary of findings and status of remediation

5. Bug Reports -----------------------

Bug ID #1 [Fixed]

Partial Denial of Service through Contacts

Vulnerability Type

Denial of Service

Severity

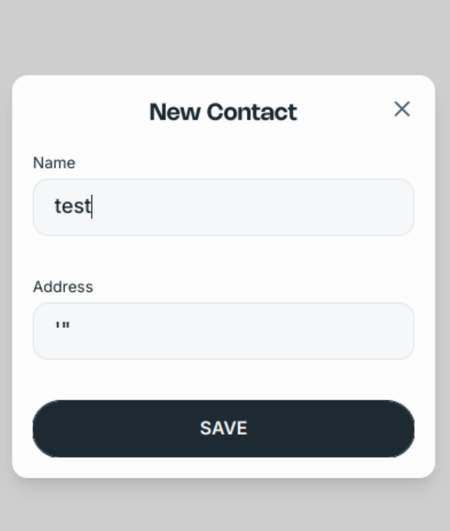
Medium

Description

A partial denial-of-service vulnerability was identified in the Arcana Chrome Wallet Extension. The issue occurs when users add a new contact with an invalid address, which causes the Contacts page to become unresponsive. This vulnerability is due to inadequate input validation, allowing invalid data to be processed and leading to functionality impairment. Although this does not crash the entire application, it restricts access to the Contacts page indefinitely.

Proof of Concept

1. Log in to your wallet and go to Settings > Contacts.
2. Add a new contact with an invalid address as shown below.
3. Once this is saved the page will stop responding due to improper input validations leading to a partial denial of service.



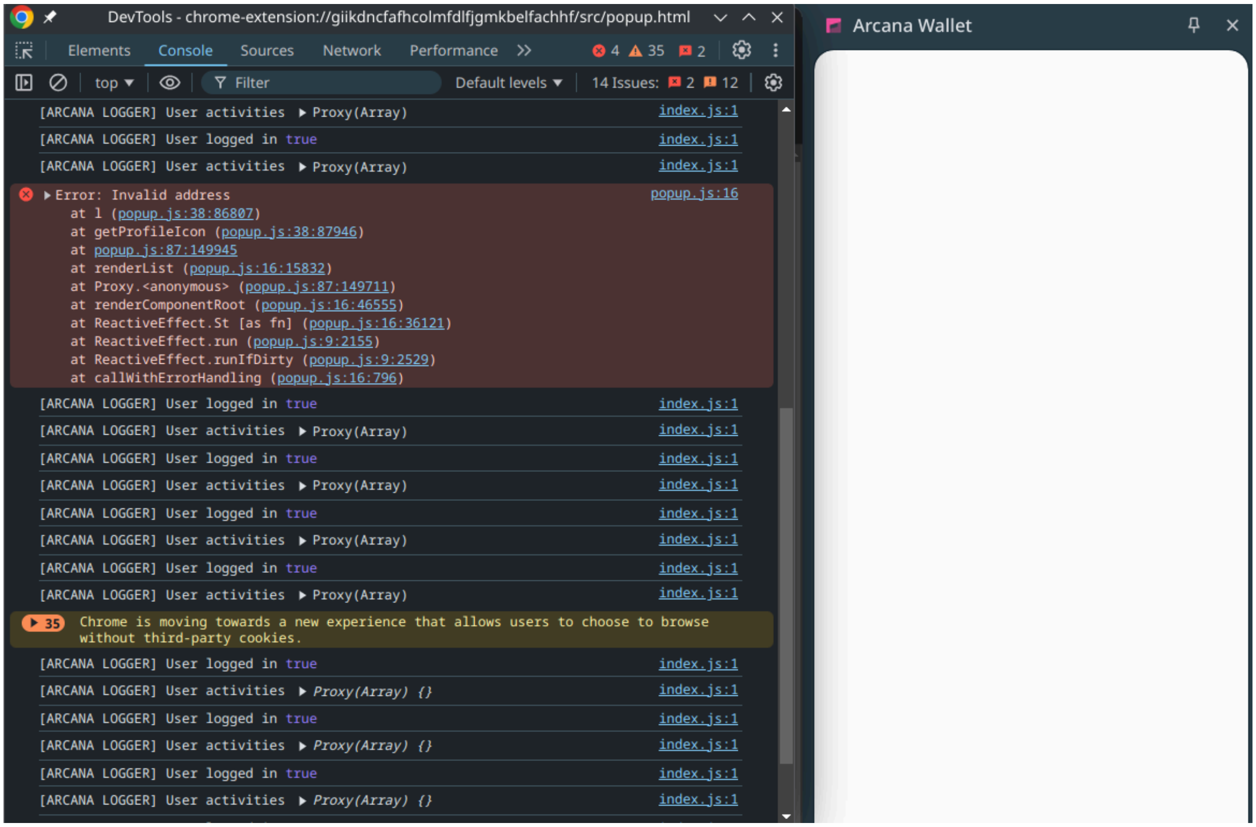
New Contact

Name

test

Address

'"



```
[ARCANA LOGGER] User activities ▶ Proxy(Array)                    index.js:1
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array)                    index.js:1
▶ Error: Invalid address                                         popup.js:16
    at l (popup.js:38:86807)
    at getProfileIcon (popup.js:38:87946)
    at popup.js:87:149945
    at renderList (popup.js:16:15832)
    at Proxy.<anonymous> (popup.js:87:149711)
    at renderComponentRoot (popup.js:16:46555)
    at ReactiveEffect.St [as fn] (popup.js:16:36121)
    at ReactiveEffect.run (popup.js:9:2155)
    at ReactiveEffect.runIfDirty (popup.js:9:2529)
    at callWithErrorHandling (popup.js:16:796)
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array)                    index.js:1
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array)                    index.js:1
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array)                    index.js:1
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array)                    index.js:1
▶ 35  Chrome is moving towards a new experience that allows users to choose to browse
      without third-party cookies.
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array) {}                 index.js:1
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array) {}                 index.js:1
[ARCANA LOGGER] User logged in true                               index.js:1
[ARCANA LOGGER] User activities ▶ Proxy(Array) {}                 index.js:1
```

Impacts

This vulnerability leads to a partial denial of service on the Contacts page, impacting user experience and usability. Although low in severity, it may disrupt wallet functionality if users inadvertently input invalid addresses.

Remediation

- Implement robust input validation for contact addresses to ensure they meet the required address format before submission.
- Provide user feedback if an invalid address format is detected, preventing the contact from being saved.
- Consider incorporating error handling to ensure the page remains responsive in cases of invalid inputs.

Retest

Input validation has been implemented properly.

Bug ID #2 [Fixed]

Weak Vault Passwords

Vulnerability Type

CWE-521: Weak Password Requirements

Severity

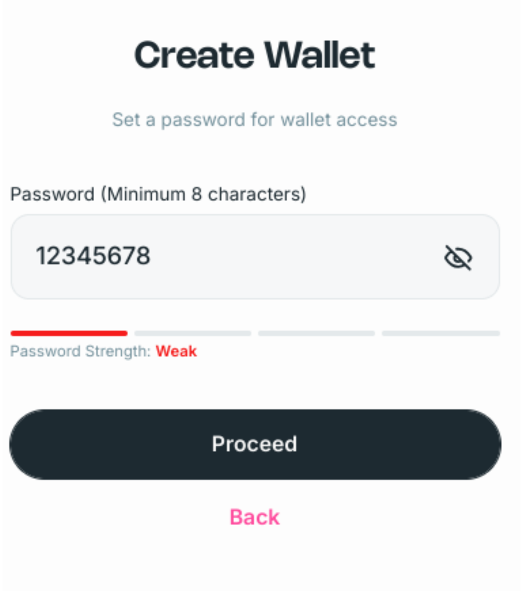
Low

Description

A weak password policy vulnerability has been identified in the Arcana Chrome Wallet Extension. The issue allows users to set weak passwords for their vaults without enforcing minimum complexity requirements. This increases the risk of unauthorized access to wallets if simple, easily guessable passwords are used. Allowing weak passwords like "12345678" reduces the security strength of the vault and makes it vulnerable to brute-force or guessing attacks.

Proof of Concept

1. Create a wallet and use a weak password such as “12345678”.



Create Wallet

Set a password for wallet access

Password (Minimum 8 characters)

12345678

Password Strength: Weak

Impacts

Weak password enforcement can lead to an increased risk of unauthorized access, especially if users select common or easily guessed passwords. This vulnerability undermines the wallet's overall security by relying on passwords that do not meet best practice standards for password complexity.

Remediation

Enforce a stronger password policy by requiring a minimum password length and complexity (e.g., a mix of uppercase letters, lowercase letters, numbers, and special characters).
Implement password strength validation during wallet setup to guide users toward creating stronger passwords.

Retest

The password policy is now stronger and enforced properly.

Bug ID #3 [Fixed]

Missing Validation in Token URL

Vulnerability Type

CWE-20: Improper Input Validation

Severity

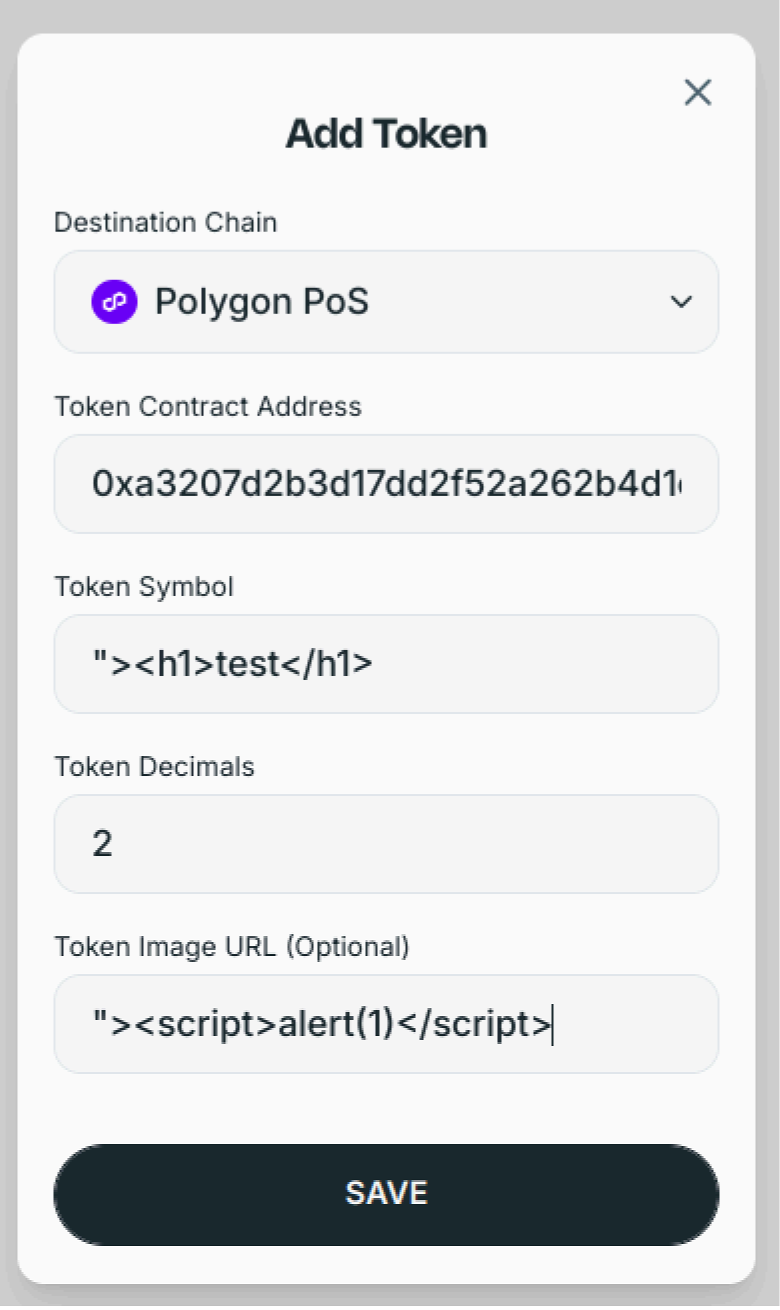
Low

Description

A missing input validation vulnerability has been identified in the Arcana Chrome Wallet Extension's token URL field. This vulnerability occurs when adding a new token to the wallet; users can input any URL and potentially malicious XSS payloads. Due to the lack of validation on the token URL parameter, malicious URLs can be saved, creating potential security risks like Cross-Site Scripting (XSS) and HTML Injections.

Proof of Concept

1. Add a new token to the wallet and enter any invalid value in the token URL such as any malicious payload.
2. It can be seen that there is no validation on the parameter allowing users to input any malicious payloads.



Add Token

Destination Chain

Polygon PoS

Token Contract Address

0xa3207d2b3d17dd2f52a262b4d1

Token Symbol

"><h1>test</h1>

Token Decimals

2

Token Image URL (Optional)

"><script>alert(1)</script>

Impacts

This vulnerability could lead to the injection of malicious scripts, which might result in XSS attacks or expose users to insecure resources. The absence of validation may enable attackers to add harmful payloads, impacting user security and trust.

Remediation

Implement strict input validation on the token URL field to allow only URLs. Use a regex-based validation to reject all the unspecified input characters.

Retest

Only HTTP, HTTPS, and data URL schemes are now allowed.

Bug ID #4 [Fixed]

Using Insecure HTTP URLs in Token URL

Vulnerability Type

CWE-319: Cleartext Transmission of Sensitive Information

Severity

Low

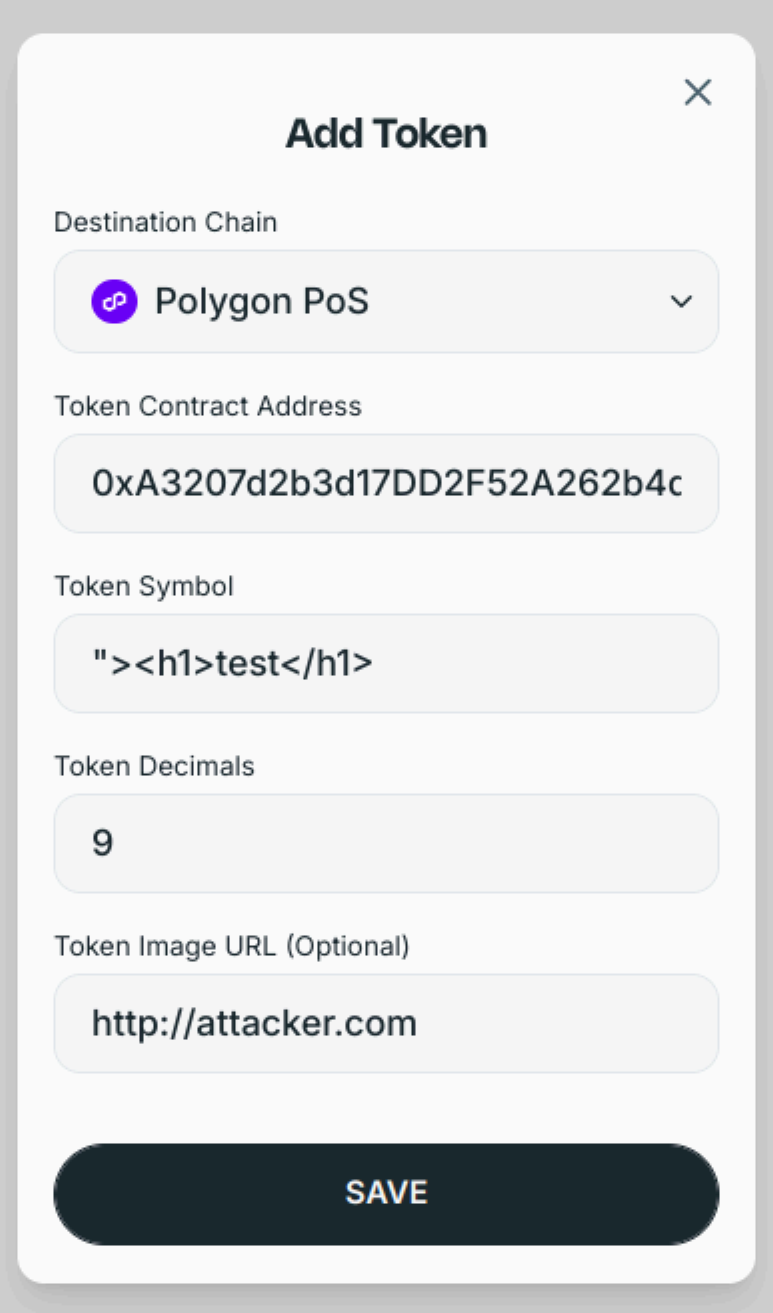
Description

A missing input validation vulnerability has been identified in the Arcana Chrome Wallet Extension’s token URL field.

This issue occurs when adding a new token to the wallet: users are allowed to input URLs using the insecure HTTP protocol, which lacks encryption and may expose users to data interception and tampering risks.

Proof of Concept

1. Add a new token to the wallet and enter any invalid value in the token URL such as an HTTP address.
2. It can be seen that there is no validation on the parameter allowing users to use insecure HTTP protocol in the URL.



Destination Chain: Polygon PoS

Token Contract Address: 0xA3207d2b3d17DD2F52A262b4c

Token Symbol: "><h1>test</h1>

Token Decimals: 9

Token Image URL (Optional): http://attacker.com

Impacts

By allowing insecure HTTP URLs, this vulnerability could expose users to the risk of data interception by malicious third parties, as HTTP lacks encryption.

Remediation

Implement strict input validation on the token URL field to allow only HTTPS URLs. Reject all HTTP URLs and any unspecified input characters. Use a regex-based validation method to enforce this rule.

Retest

HTTP URL schemes are not supported anymore.

Bug ID #5 [Fixed]

Using Insecure V2 Manifest for Firefox

Vulnerability Type

CWE-16: Misconfiguration

Severity

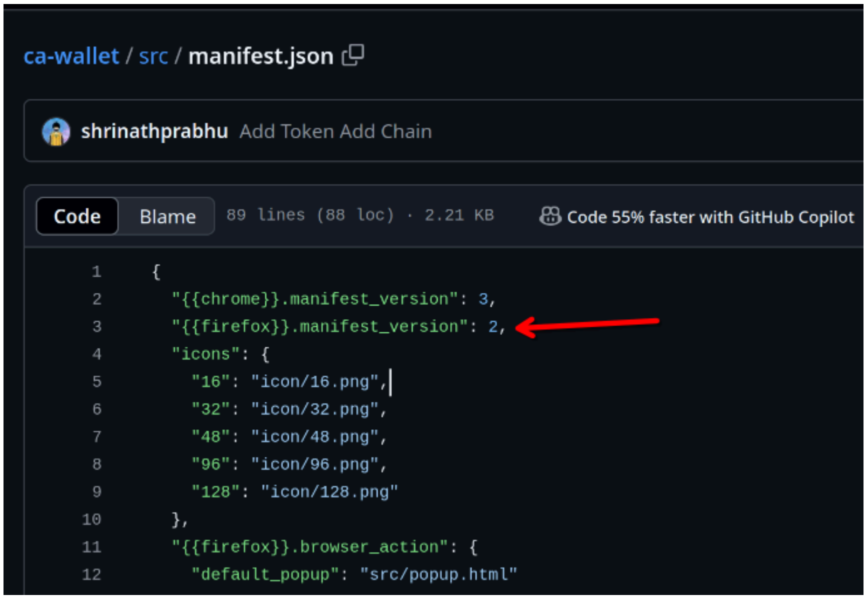
Low

Description

The extension is currently using Manifest Version 2 for Firefox, which is deprecated and no longer supported for new submissions on the Mozilla Add-ons store. Manifest Version 2 does not support newer security, performance, and compatibility improvements found in Manifest Version 3, which is now recommended for all Firefox extensions. Continuing to use Manifest Version 2 can lead to compatibility and security issues and prevent future updates from being accepted by Mozilla.

Proof of Concept

1. View the manifest.json file to see the V2 version being used for Firefox.


ca-wallet / src / manifest.json
Code
Blame
89 lines (88 loc) · 2.21 KB
Code 55% faster with GitHub Copilot

shrinathprabhu Add Token Add Chain

```
{
  "{{chrome}}.manifest_version": 3,
  "{{firefox}}.manifest_version": 2,
  "icons": {
    "16": "icon/16.png",
    "32": "icon/32.png",
    "48": "icon/48.png",
    "96": "icon/96.png",
    "128": "icon/128.png"
  },
  "{{firefox}}.browser_action": {
    "default_popup": "src/popup.html"
```

Impacts

Using Manifest Version 2 may lead to the following issues:

- Incompatibility with future Firefox updates that drop support for Manifest Version 2.
- Reduced Security: Manifest Version 3 introduces a more robust permission model, making extensions more secure.
- Submission Block: Mozilla has started phasing out Manifest Version 2, and extensions using it may not be approved for new updates or submissions on the Add-ons store.

Remediation

Update the extension to use Manifest Version 3 for Firefox as well.

Retest

Manifest V3 is now being used.

6. The Disclosure -----------------------

The reports provided by CredShields are neither an endorsement nor a condemnation of any specific project, team, or browser extension and do not guarantee the security of any specific extension. The contents of this report are not intended to influence decisions about purchasing, using, or endorsing browser extensions, services, or related products and should not be interpreted as such.

Web technologies and online platforms often carry a high level of technical risk and uncertainty. CredShields does not provide any warranty or representation regarding the quality of the code, the business model, the extension's proprietors, or its compliance with legal or regulatory standards.

The CredShields Audit team is not responsible for any decisions or actions taken by third parties based on the contents of this report.

YOUR SECURE FUTURE STARTS HERE

CRED SHiELDS

At CredShields, we're more than just auditors. We're your strategic partner in ensuring a secure Web3 future. Our commitment to your success extends beyond the report, offering ongoing support and guidance to protect your digital assets

Audited by CRED SHiELDS