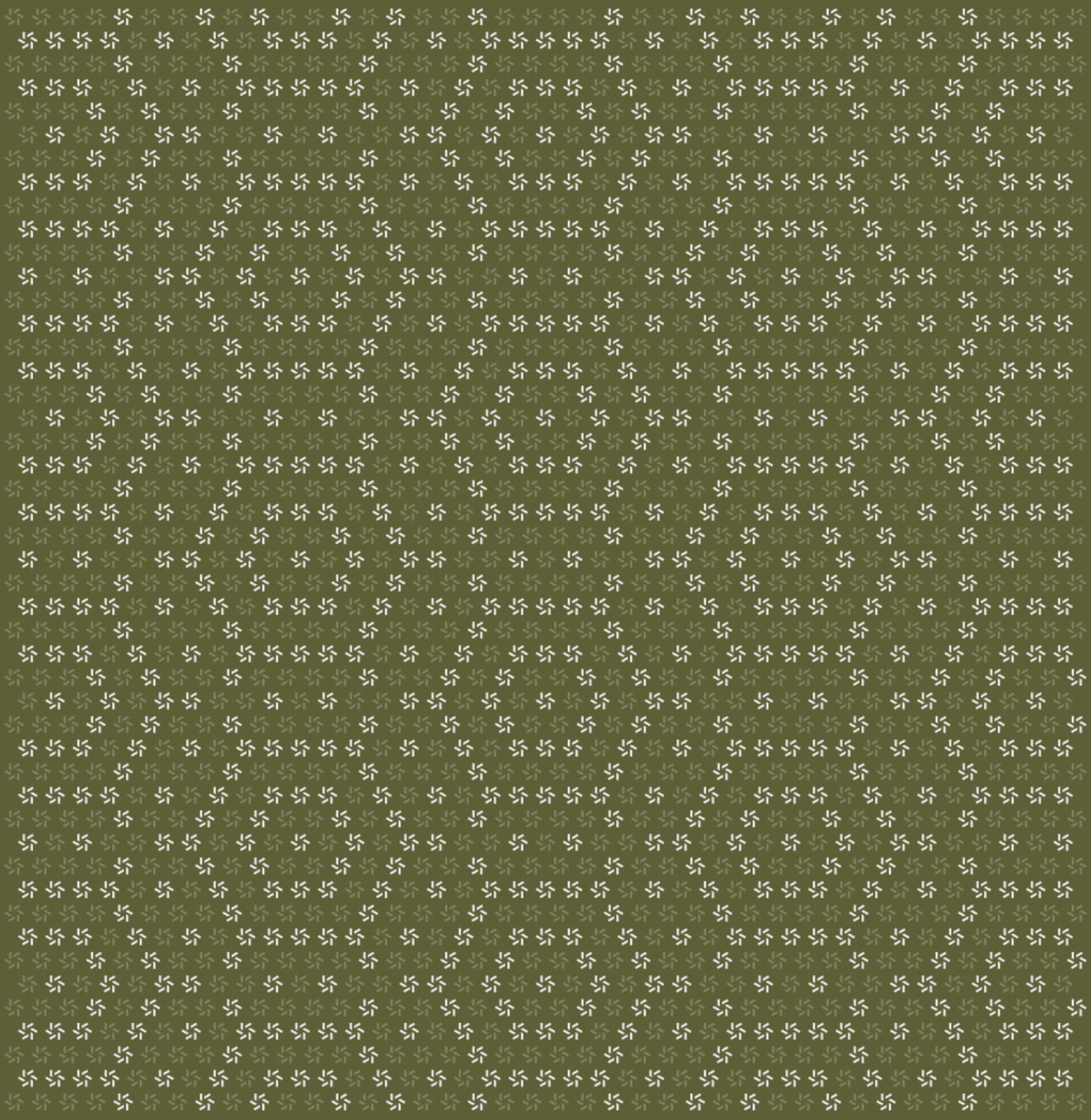
Zellic

Prepared for
Avara Labs Ltd.

Prepared by
Philippe Dourassov
Maik Robert
Zellic

March 11, 2025

Family Accounts
Web Wallet Security Assessment



Contents

About Zellic

Zellic is a vulnerability research firm with deep expertise in blockchain security. We specialize in EVM, Move (Aptos and Sui), and Solana as well as Cairo, NEAR, and Cosmos. We review L1s and L2s, cross-chain protocols, wallets and applied cryptography, zero-knowledge circuits, web applications, and more.

Prior to Zellic, we founded the #1 CTF (competitive hacking) team worldwide in 2020, 2021, and 2023. Our engineers bring a rich set of skills and backgrounds, including cryptography, web security, mobile security, low-level exploitation, and finance. Our background in traditional information security and competitive hacking has enabled us to consistently discover hidden vulnerabilities and develop novel security research, earning us the reputation as the go-to security firm for teams whose rate of innovation outpaces the existing security landscape.

For more on Zellic's ongoing security research initiatives, check out our website zellic.io and follow @zellic_io on Twitter. If you are interested in partnering with Zellic, contact us at hello@zellic.io.

1. Overview

1.1. Executive Summary

Zellic conducted a security assessment for Avara Labs Ltd. from March 3rd to March 7th, 2025. During this engagement, Zellic reviewed Family Accounts's code for security vulnerabilities, design issues, and general weaknesses in security posture.

1.2. Goals of the Assessment

In a security assessment, goals are framed in terms of questions that we wish to answer. These questions are agreed upon through close communication between Zellic and the client. In this assessment, we sought to answer the following questions:

- Is sensitive user information (e.g., private keys, mnemonics) handled in a secure way?
- Can a malicious dApp get possession of sensitive user data or user funds?
- Does the cross-window and pop-up communication work as intended?

1.3. Non-goals and Limitations

We did not assess the following areas that were outside the scope of this engagement:

- Components that did not relate to the handling of sensitive user data and cross-window/pop-up communication
- Infrastructure relating to the project

Due to the time-boxed nature of security assessments in general, there are limitations in the coverage an assessment can provide.

1.4. Results

During our assessment on the scoped Family Accounts modules, we discovered three findings. No critical issues were found. Two findings were of high impact and one was of medium impact.

Breakdown of Finding Impacts

Impact Level	Count
Critical	0
High	2
Medium	1
Low	0
Informational	0

2. Introduction

2.1. About Family Accounts

Avara Labs Ltd. contributed the following description of Family Accounts:

> Family accounts is a self-custodial wallet that allows creating a wallet with a username and password/-passkey but in a secure, encrypted way with a focus on improving the UX and enabling easy management of digital assets.

2.2. Methodology

During a security assessment, Zellic works through standard phases of security auditing, including both automated testing and manual review. These processes can vary significantly per engagement, but the majority of the time is spent on a thorough manual review of the entire scope.

Alongside a variety of tools and analyzers used on an as-needed basis, Zellic focuses primarily on the following classes of security and reliability issues:

Unsafe code patterns. Many vulnerabilities in web-based applications arise from oversights during development. Missing an authentication check on a single API endpoint can critically impact the security of the overall application. Failure to properly sanitize and encode user input can lead to vulnerabilities like SQL injection, server-side request forgery, and more. We use both automated tools and extensive manual review to identify unsafe code patterns.

Business logic errors. Business logic is the heart of all web applications. We carefully review logic to ensure that the code securely and correctly implements the specified functionality. We also thoroughly examine all specifications for inconsistencies, flaws, and vulnerabilities, including for risks of fraud and abuse.

Integration risks. Web projects frequently have many third-party dependencies and interact with services and libraries that are not under the developer's control. We review the project's external interactions and identify potentially dangerous behavior resulting from compatibility issues and data inconsistencies.

Code maturity. We look for possible improvements across the entire codebase, reviewing violations of industry best practices and code quality standards. We suggest improvements to code clarity, documentation, and testing practices.

For each finding, Zellic assigns it an impact rating based on its severity and likelihood. There is no hard-and-fast formula for calculating a finding's impact. Instead, we assign it on a case-by-case basis based on our judgment and experience. Both the severity and likelihood of an issue affect its impact. For instance, a highly severe issue's impact may be attenuated by a low likelihood. We assign the following impact ratings (ordered by importance): Critical, High, Medium, Low, and Informational.

Zellic organizes its reports such that the most important findings come first in the document, rather

than being strictly ordered on impact alone. Thus, we may sometimes emphasize an "Informational" finding higher than a "Low" finding. The key distinction is that although certain findings may have the same impact rating, their *importance* may differ. This varies based on various soft factors, like our clients' threat models, their business needs, and so on. We aim to provide useful and actionable advice to our partners considering their long-term goals, rather than a simple list of security issues at present.

2.3. Scope

The engagement involved a review of the following targets:

Family Accounts Modules

Type	TypeScript
Platform	Web
Target	family-account
Repository	https://github.com/family/family-account ↗
Version	`3b9704c645c513c9df33f63945d32d3119be9cee`
Programs	`*.ts` `*.tsx`

2.4. Project Overview

Zellic was contracted to perform a security assessment for a total of 1.5 person-weeks. The assessment was conducted by two consultants over the course of one calendar week.

Contact Information

The following project managers were associated with the engagement:

Jacob Goreski
Engagement Manager
jacob@zellic.io ↗

Chad McDonald
Engagement Manager
chad@zellic.io ↗

The following consultants were engaged to conduct the assessment:

Philippe Dourassov
Engineer
philippe@zellic.io ↗

Maik Robert
Engineer
maik@zellic.io ↗

2.5. Project Timeline

The key dates of the engagement are detailed below.

March 3, 2025	Start of primary review period
March 4, 2025	Kick-off call
March 7, 2025	End of primary review period

3. Detailed Findings

3.1. Debouncing in ClickjackGuard can be abused

Target	apps/pocket-web/components/common/ClickjackGuard/index.tsx		
Category	Coding Mistakes	**Severity**	High
Likelihood	Medium	**Impact**	High

Description

The ClickjackGuard was introduced to prevent clickjacking attacks. It does this by using the IntersectionObserver browser API to see if an element is covering any part of the target element it is protecting.

Due to React behaviour, a debouncing feature was added to the initialization of the ClickjackGuard to prevent false positives.

```
const useElementVisibilityStatus = () => {
  const [observerEntry, setObserverEntry]
    = useState<IntersectionObserverEntry | null>(null);
  const previousObserver = useRef<IntersectionObserver | null>(null);
  // debouncing to avoid false positives
  // sometimes observer is triggered  before react renders elements
  const debouncedObserverEntry = useDebounce(observerEntry, 200);

  const customRef = useCallback((node: HTMLDivElement | null) => {
    if (previousObserver.current) {
      previousObserver.current.disconnect();
      previousObserver.current = null;
    }
    ...
```

The documentation for the `useDebounce` hook describes it as follows:

> The `useDebounce` hook is useful for delaying the execution of functions or state updates until a specified time period has passed without any further changes to the input value. This is especially useful in scenarios such as handling user input or triggering network requests, where it effectively reduces unnecessary computations and ensures that resource-intensive operations are only performed after a pause in the input activity.

This means the feature can be abused to bypass the ClickjackGuard by applying a CSS style to the target iframe that makes it invisible. While this should trigger the ClickjackGuard, the debouncing delay allows for a time window during which the element is hidden. A script can be used to hide the

target element most of the time, turn it back to visible when the debouncing delay is over, and then hide it back. Here is a proof-of-concept snippet that can be used to abuse the debouncing feature:

```
setInterval(() => {
  setTimeout(() => {
    const iframes = document.querySelectorAll('iframe');
    iframes.forEach((iframe) => {
      iframe.style.opacity = '0';
      })},70)
  setTimeout(() => {
    const iframes = document.querySelectorAll('iframe');
    iframes.forEach((iframe) => {
      iframe.style.opacity = '1';
       })}, 50)
}, 200)
```

Impact

An attacker can trick a user into performing an action without the user's knowledge by hiding the target element from the user. This can lead to unauthorized transactions or other malicious actions.

Recommendations

Disable the debouncing feature in the ClickjackGuard and instead disable the button for a short period of time after the modal is opened.

Remediation

This issue has been acknowledged by Avara Labs Ltd..

3.2. CSS manipulation on iframe bypasses clickjacking defense

Target	apps/pocket-web/components/common/ClickjackGuard/index.tsx		
Category	Coding Mistakes	**Severity**	High
Likelihood	Medium	**Impact**	High

Description

The IntersectionObserver API is used to detect if an element of the iframe is hidden or covered by another element. This is used to prevent clickjacking attacks. However, the IntersectionObserver API does not detect if the iframe is manipulated by CSS to be larger than the viewport. This can be used to bypass the clickjacking defense and hide important information from the victim.

The following script sets specific CSS attributes that can be used to manipulate the iframe and only display a clickable blue button without any information on its associated actions:

```
iframe.style.height = '5000px';
iframe.style.width = '360px';
iframe.style.marginTop = '-700px';
iframe.style.marginLeft = '-340px';
iframe.style.zoom = "800%";
```

The blue button corresponds to the action-confirmation button, such as confirming message signatures.

Impact

An attacker can trick a user into performing a wallet action without the user's knowledge. This can lead to unauthorized transactions or other malicious actions.

Recommendations

To avoid such an issue, it is necessary to guarantee the injected iframe and its information are entirely visible. Alternatively, making use of a window pop-up could guarantee there is no manipulation.

Remediation

This issue has been acknowledged by Avara Labs Ltd..

3.3. Lack of confirmation-button activation delay

Target	apps/pocket-web/surfaces/integrated-modals/sign-approval/PersonalSignApproval.tsx		
Category	Coding Mistakes	**Severity**	High
Likelihood	Low	**Impact**	Medium

Description

When an application initiates a request to perform an action with the user wallet, such as signing a transaction, a confirmation page is displayed to the user via either an iframe or a pop-up window. This page contains a button that the user must click to confirm the action.

```
<Button id="continue" wide onClick={signMessage}>
  Sign Message
</Button>
```

The button is immediately active when displayed. An attacker can abuse this fact to clickjack the user. By overlaying the confirmation button with an attacker-controlled button, the attacker can convince the user to double-click on their malicious button. The first click would display the confirmation page, and the second click would instantly confirm the action.

Impact

An attacker can trick a user into performing a wallet action without the user's knowledge. This can lead to unauthorized transactions or other malicious actions.

Recommendations

Add a delay to the activation of the confirmation button. This delay should be long enough to prevent any accidental click on the confirmation button.

Remediation

This issue has been acknowledged by Avara Labs Ltd..

4. Assessment Results

At the time of our assessment, the reviewed code was not deployed to production.

During our assessment on the scoped Family Accounts modules, we discovered three findings. No critical issues were found. Two findings were of high impact and one was of medium impact.

4.1. Disclaimer

This assessment does not provide any warranties about finding all possible issues within its scope; in other words, the evaluation results do not guarantee the absence of any subsequent issues. Zellic, of course, also cannot make guarantees about any code added to the project after the version reviewed during our assessment. Furthermore, because a single assessment can never be considered comprehensive, we always recommend multiple independent assessments paired with a bug bounty program.

For each finding, Zellic provides a recommended solution. All code samples in these recommendations are intended to convey how an issue may be resolved (i.e., the idea), but they may not be tested or functional code. These recommendations are not exhaustive, and we encourage our partners to consider them as a starting point for further discussion. We are happy to provide additional guidance and advice as needed.

Finally, the contents of this assessment report are for informational purposes only; do not construe any information in this report as legal, tax, investment, or financial advice. Nothing contained in this report constitutes a solicitation or endorsement of a project by Zellic.