// Security Assessment

03.31.2025 - 04.01.2025

Biz - Account Abstraction Module

Trust Wallet

HALBORN

Biz - Account Abstraction Module - Trust Wallet

Prepared by: HALBORN

Last Updated 04/16/2025

Date of Engagement: March 31st, 2025 - April 1st, 2025

Summary

100% OF ALL REPORTED FINDINGS HAVE BEEN ADDRESSED

ALL FINDINGS	CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
5	0	0	1	0	4

TABLE OF CONTENTS

1. Introduction
2. Assesment summary
3. Caveats
4. Scope
5. Findings overview

1. INTRODUCTION

`Trust Wallet` engaged `Halborn` to conduct a security assessment on their smart contracts beginning on April 1st, 2025 and ending on April 2nd, 2025. The security assessment was scoped to the smart contracts provided to the `Halborn` team. Commit hashes and further details can be found in the Scope section of this report.

2. ASSESMENT SUMMARY

The team at `Halborn` assigned two full-time security engineers to assess the security of the smart contracts. The security engineers are blockchain and smart-contract security experts with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this assessment is to:

- Ensure that smart contract functions operate as intended.
- Identify potential security issues with the smart contracts.

In summary, Halborn identified some improvements to reduce the likelihood and impact of risks, which were mostly addressed by the `Trust Wallet team`. The main ones were the following:

- `Nonce updates should occur strictly as part of an authorized signature flow, gated by an internal flag set temporarily by the wallet. The external nonce manager must verify this flag before applying any update.`
- `Declare support for standard signature validation by explicitly implementing the relevant interface and registering it for ERC-165 detection.`
- `Add a constructor check in the wallet contract to reject zero addresses and ensure it's deployed with a valid external controller.`

3. CAVEATS

The `check4337Flag` modifier contains a fallback condition that allows execution of 4337-related functions if a specific `MAGIC_VALUE` (6789) is found in persistent storage at the `BIZ_4337_FLAG` slot. This bypasses the usual requirement of having transient storage set via `validateUserOp()`.

Although this may appear as a potential backdoor, the development team confirmed during a review meeting that this mechanism is **intentionally included for testing purposes only**. No function in the production contract writes this value to persistent storage, and the fallback is automatically cleared after one use.

As such, this mechanism does **not constitute a vulnerability**, but its presence should be noted to avoid confusion during assessments or external integrations.

4. SCOPE

FILES AND REPOSITORY

(a) Repository: Biz

(b) Assessed Commit ID: 26510e7

(c) Items in scope:

- src/Biz.sol
- src/BizGuard.sol
- src/BizHandler.sol

Out-of-Scope: Third party dependencies and economic attacks.

REMEDIATION COMMIT ID:

- d013452
- ae00f61
- ed3f4c5

Out-of-Scope: New features/implementations after the remediation commit IDs.

5. FINDINGS OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION
DELEGATED SIGNATURES CAN BE INVALIDATED BY ARBITRARY CALL MECHANISMS	MEDIUM	RISK ACCEPTED - 04/15/2025
INTERFACE SUPPORT DECLARATION MISSING FOR VALID SIGNATURE VERIFICATION IMPLEMENTATION	INFORMATIONAL	SOLVED - 04/09/2025
REDUNDANT ARRAY LENGTH ACCESS INSIDE LOOPS LEADS TO AVOIDABLE GAS USAGE	INFORMATIONAL	SOLVED - 04/07/2025
LACK OF ZERO ADDRESS CHECK ALLOWS INVALID EXTERNAL REFERENCE	INFORMATIONAL	SOLVED - 04/15/2025
NO HANDLING OF FAILED ETH TRANSFER TO ENTRYPOINT	INFORMATIONAL	ACKNOWLEDGED - 04/15/2025

Halborn strongly recommends conducting a follow-up assessment of the project either within six months or immediately following any material changes to the codebase, whichever comes first. This approach is crucial for maintaining the project's integrity and addressing potential vulnerabilities introduced by code modifications.