0xCommit

Security Assessment Report

Biconomy - Smart Wallet Contracts

Batched Session Router

Version: Final

Date: 1 Nov 2023

Table of Contents

Licence

Disclaimer

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHOR AND HIS EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS AUDIT REPORT.

Introduction

Purpose of this report

0xCommit has been engaged by **Biconomy** to perform a security audit of several Smart Account components.

The objectives of the audit are as follows:

1. Determine the correct functioning of the protocol, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine smart contract bugs, which might lead to unexpected behaviour.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebases Submitted for the Audit

The audit has been performed on the following GitHub repositories:

Repository :
https://github.com/bcnmy/scw-contracts/blob/develop/contracts/smart-account/modules/BatchedSessionRouterModule.sol

Version	Commit hash
Initial Codebase	a4ee210ebabe04d0e6ebabedeaa8c0963068b86a
Fixed Codebase	906d1b6f8e95df36d18c64cb671e885c2d6be369

How to Read This Report

This report classifies the issues found into the following severity categories:

Severity	Description
Critical	A serious and exploitable vulnerability that can lead to loss of funds, unrecoverable locked funds, or catastrophic denial of service.
High	An attacker can successfully execute an attack that clearly results in operational issues for the service. This also includes any value loss of unclaimed funds permanently or temporary.
Medium	The service may be susceptible to an attacker carrying out an unintentional action, which could potentially disrupt its operation. Nonetheless, certain limitations exist that make it difficult for the attack to be successful.
Low	The service may be vulnerable to an attacker executing an unintended action, but the impact of the action is negligible or the likelihood of the attack succeeding is very low and there is no loss of value.
Informational	Comments and recommendations of design decisions or potential optimizations, that are not relevant to security. Their application may improve aspects, such as user experience or readability, but is not strictly necessary

The status of an issue can be one of the following: **Pending**, **Acknowledged**, or **Resolved**.

Note that audits are an important step to improving the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of the system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**. We include a table with these criteria below.

Note that high complexity or low test coverage does not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than in a security audit and vice versa.

Overview

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the code base's intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under-/overflow issues
    c. Key management vulnerabilities
    d. Access Control Issues
    e. Boundary Analysis
4. Report preparation

Functionality Overview

Biconomy is an extensible Account Abstraction protocol, which allows users to create Smart wallets. Biconomy has developed the Modular Session keys framework which provides great flexibility and allows for quick building of the Session Validation Modules for every new use case without touching the core Session Keys logic.

Batch Session Router adds composability, allowing batching of several session key signed operations which should be validated by different Session Validation modules into one User Operation and execute them atomically.

Summary of Findings

Sr. No.	Description	Severity	Status
1	Attacker can use arbitrary Session Key Manager to take over user's Smart Account	Critical	Resolved

Detailed Findings

1. Attacker can use arbitrary Session Key Manager to take over user's Smart Account

Severity: Critical

Description

The BatchedSessionRouter contract currently allows any Session Key Manager module to be utilized, even if it has not been officially enabled by the user. This, in itself, is permissible due to the appropriate signature checks on (userOpHash + Session Key Manager). However, upon thorough examination, we have discovered that an attacker can create their own moduleSignature and deploy a modified version of the Session Key Manager and validation modules to pass the validateUserOp function.

Impact : This vulnerability allows attackers to gain control of smart wallets and access associated funds without any interaction or approval from the wallet owner.

Remediation

Check if a Session Key Manager has been enabled as a module for that smart account.

Status

Resolved