0xCommit

Security Assessment Report

Biconomy - Paymaster

Sponsorship Paymaster

Version: Final

Date: 6 Dec 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 commits:

Version	Commit hash	Github link
Initial	b55d84781af89ebbc57b86db403038280671b4fb	https://github.com/bcnmy/biconomy-paymasters/commits/develop/contracts/verifying/VerifyingSingletonPaymasterV2.sol
Final	45be83feded990df019eba0c7b7ff042da18564a	https://github.com/bcnmy/biconomy-paymasters/blob/develop/contracts/sponsorship/SponsorshipPaymaster.sol

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

SponsorshipPaymaster is a paymaster based on Account abstraction framework built by Biconomy. It allows Dapps to sponsor on behalf of their users using the Smart accounts and Sponsorship Paymaster provided by Biconomy.

Summary of Findings

Sr. No.	Description	Severity	Status
1	Funds can be locked into an entry point contract with no recourse to recover them	Low	Resolved
2	Discrepancy in Fee Deduction Between EntryPoint and Paymaster	Low	Resolved
3	Centralization Risk	Informatio...	Acknowledged

Detailed Findings

1. Funds can be locked into an entry point contract with no recourse to recover them.

Severity: Low

Description

If the user triggers the depositTo with address of Sponsorship Paymaster contract on Entrypoint contract and sends in some native tokens then there is no way to recover that funds from Sponsorship Paymaster contract and this funds is perpetually locked in the entrypoint contract under the name of Sponsorship Paymaster contract.

Remediation

There are two solutions for this

- Add a admin method to withdrawFrom entry point contract (This leads to centralization risk)
- Change StakeManager.sol such that all deposits can only be done by sender to himself (Users can deposit on behalf of) [This may be needed for some other functional needs also]

Status

Resolved

2. Discrepancy in Fee Deduction Between EntryPoint and Paymaster

Severity: Low

Description

There can be a difference between :

- the amount deducted from the paymasterIdBalances for a paymasterId
- the actual amount deducted for the paymaster by the Entry Point

This discrepancy primarily arises due to the variable **unaccountedEPGasOverhead**. If this variable is not precisely calibrated to account for additional gas costs, it leads to a variance between the balances in EntryPoint's deposit and the paymasterIdBalances mapping within the SponsorshipPaymaster.

Impact : The impact of this discrepancy is assessed to be minimal. The difference in balances is expected to be relatively small. Moreover, considering that multiple users will be depositing into the Paymaster, this slight imbalance is unlikely to significantly affect the Paymaster's overall functionality.

Remediation

In consultation with the Biconomy team, it is recommended that unaccountedEPGasOverhead be set to a carefully considered value to minimize this discrepancy. This value should not be static. Also regular monitoring and adjustments are advised to ensure it remains effective and responsive to any changes in gas costs or other relevant factors.

Status

Resolved

A proper value for the unaccountedEPGasOverhead was found after some testing and was set as the default value.

3. Centralization Risk

Severity: Informational

Description

The 'verifyingSigner' is empowered to supply a signature for the paymaster data, granting them the ability to utilize any paymasterId during the signature generation process. However, this confers a centralization risk upon the verifyingSigner since they are required to scrutinize each user's signature request and verify whether the user has obtained authorization to be sponsored by the specified paymasterId.

Impact :

In the event that the backend API responsible for providing signatures for sponsorship paymasters is exploited or if the private keys of the verifyingSigner are compromised, it would enable potential attackers to sponsor userOps with any desired paymasterId, essentially bypassing the intended security measures.

Remediation

- ensure only authorized users can request sponsorship signatures for a specific paymasterId
- employ strict authentication and authorization mechanisms
- securely manage private keys

Status

Acknowledged

Comments from Biconomy: We're ensuring that verifying signer key is managed behind a private network. The signing service makes all these checks and the paymasterAndData signature is signed over a paymasterId.