0xCommit

Security Assessment Report

Biconomy

Security Policy Plugins

Version: Final

Date: 29 Dec 2023

Table of Contents

Licence

Disclaimer

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	617a7b1c05534a3bac71920b562f6c94ac31b389	https://github.com/bcnmy/scw-contracts/blob/develop/contracts/smart-account/modules/SecurityPolicyManagerPlugin.sol https://github.com/bcnmy/scw-contracts/blob/develop/contracts/smart-account/modules/SecurityPolicies/ERC7484SecurityPolicy.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

The goal of a security policy plugin is to implement arbitrary conditional checks by combining data from an arbitrary number of attestations, from the same or different schemas for a given plugin. There is a Security Policy plugin manager which allows enabling and disabling the policies.

Summary of Findings

Sr. No.	Description	Severity	Status
1	Required threshold may be 0 for validating security policy	Informatio...	Acknowledged
2	Same configuration(checks) used for all the plugin installation	Informatio...	Acknowledged

Detailed Findings

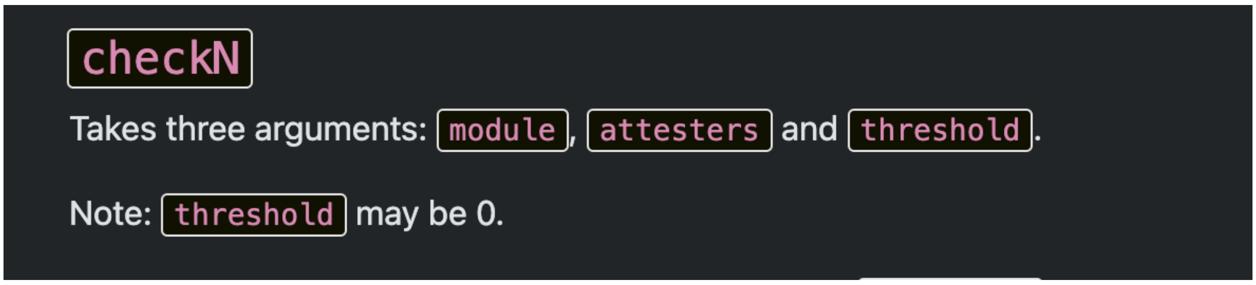
1. Required threshold may be 0 for validating security policy

Severity: Informational

Description

In validateSecurityPolicy, the function reverts if the required threshold for attestors is 0.

However, according to EIP7484, the threshold may be 0 and it is a valid use case.



`checkN`

Takes three arguments: `module`, `attesters` and `threshold`.

Note: `threshold` may be 0.

Remediation

Adhere to the specification by EIP 7484 instead of creating custom implementation which diverges from it.

Status

Acknowledged

2. Same configuration used for all the plugin installation

Severity: Informational

Description

ERC7484SecurityPolicyPlugin uses the same configuration for all the plugin installation and validation for smart accounts. validateSecurityPolicy fetches the configuration for smart account and then checks the registry to ensure that the plugin to be installed has configured trusted attestors. There may be the need to configure different configurations(different number of attestors) based on different modules. Some plugins may not require a high number of attestors if it is not dealing with funds while some may require more number of attestors.

Reference:
https://github.com/bcnmy/scw-contracts/blob/develop/contracts/smart-account/modules/SecurityPolicies/ERC7484SecurityPolicy.sol#L23-L26

https://github.com/bcnmy/scw-contracts/blob/develop/contracts/smart-account/modules/SecurityPolicies/ERC7484SecurityPolicy.sol#L35

Remediation

Try to configure different levels of configurations for smart accounts giving the smart account the option to select configuration based on the plugin to be installed. In the current scenario, all plugins use the same configuration.

Status

Acknowledged

Biconomy : We will consider this suggestion for the next version.