BICONOMY SMART CONTRACT AUDIT REPORT

December 8, 2020

Biconomy

MixBytes()

TABLE OF CONTENTS

01 INTRODUCTION TO THE AUDIT

General Provisions

Biconomy provides Mexa platform to implement meta transactions and allow dapp users to perform blockchain operations without holding any ether or other crypto currency.

Scope of the Audit

The scope of the audit includes the following smart contracts at:

RelayerManager.sol
GasTokenForwarder.sol
GasTokenImplementation.sol
ICHITOKEN.sol
chiToken.sol
Ownable.sol

BiconomyForwarder.sol
ERC20FeeProxy.sol
ERC20ForwardRequestCompatible.sol
ERC20TransferHandler.sol
OracleAggregator.sol

EIP712MetaTransaction.sol

02 SECURITY ASSESSMENT PRINCIPLES

Classification of Issues

- CRITICAL: Bugs leading to Ether or token theft, fund access locking or any other loss of Ether/tokens to be transferred to any party (for example, dividends).

- MAJOR: Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.

- WARNINGS: Bugs that can break the intended contract logic or expose it to DoS attacks.

- COMMENTS: Other issues and recommendations reported to/ acknowledged by the team.

Security Assessment Methodology

Two auditors independently verified the code.

Stages of the audit were as follows:

- "Blind" manual check of the code and its model
- "Guided" manual code review
- Checking the code compliance with customer requirements
- Discussion of independent audit results
- Report preparation

03 DETECTED ISSUES

CRITICAL

Not found

MAJOR

Not found

WARNINGS

1. Possible overflow leading to incorrect gas values.

This warning is about overflow-unsafe math used in here: **GasTokenImplementation.sol#L37**.

The overflow in this particular statement can break the application logic by not allowing this function to be executed with `gasLimit` values close to `uint256` max value.

It is recommended to replace highlighted statement as per the example below:

```
require(gasleft() > gasLimit.div(63).add(gasLimit).add(1000), "Not enough gas");
```

Status

Fixed at 60466055

2. Trusted forwarder ownership.

This warning is about trusted `BiconomyForwarder` which is used without explicit ownership. Since the functions this particular part of business logic manage for meta-transactions mechanism are sensitive (e.g. reformatting meta-transactions domain separators with `registerDomainSeparator` in here: **BiconomyForwarder.sol#L41**), malicious usage of such functions can lead to making the whole contract state invalid.

It is recommended to introduce explicit ownership over this particular contract (e.g. making it `Ownable`) with appending ownership modifiers (e.g. `onlyOwnert`) to sensitive functions (e.g. `registerDomainSeparator` in here:
BiconomyForwarder.sol#L41) to avoid invalidating meta-transactions functionalities.

Status

Fixed at b99486be

3. Meta-transaction incorrect signature verification timings.

This warning is about possibility to make an expired meta-transaction valid by manipulating the `block.timestamp` value (which keyword `now` is alias to) in here: **BiconomyForwarder.sol#L203** and in here: **BiconomyForwarder.sol#L247**.

It is required to extend the time range under the check in here: **BiconomyForwarder.sol#L203** and in here:
BiconomyForwarder.sol#L247
by a little bit more than average block time, checking `require(req.deadline == 0 || now + 20 <= req.deadline, "request expired");` instead of `require(req.deadline == 0 || now <= req.deadline, "request expired");`.

Status

Fixed at b99486be

4. Relayer duplication possible.

RelayerManager.sol#L33

RelayerManager.sol#L42

Potentially it is possible to add the same relayer several times because functions do not check if the relayer is already added.

We recommend adding relayer existence check before pushing to the array.

Status

Fixed at 60466055

5. Exclude recursive calls.

EIP712MetaTransaction.sol#L37

Here we have meta transaction execution via `call` of `this` contract. Potentially this code allows to call any methods including `executeMetaTransaction`.

We recommend filtering that call to avoid potential issues regarding this.

Status

Fixed at 8a6564b8

COMMENTS

1. Unnecessary Signature Verification Calculations.

This comment is about preventing redundant meta-transaction signature verification and recovery in case user address `== address(0)` right from the very beginning in here: **EIP712MetaTransaction.sol#L59**.

In case `user == address(0)`, the meta-transaction verification will be considered failed in any way, so there is no point in recovering the signature. It would be enough to implement a construction like `if (user == address(0)) { return false; }` right from the very beginning of the function to save CPU time on that.

Status

No issue

2. Mixed code style.

EIP712MetaTransaction.sol#L18-L22

EIP712MetaTransaction.sol#L44-L51

BiconomyForwarder.sol#L115

BiconomyForwarder.sol#L180

We recommend enabling some code formatter to keep the same style code(indentations, spaces between operators, etc.).

Status

Fixed at **8a6564b8**, **7a5e6e20**

04 CONCLUSION AND RESULTS

Findings list

Level	Amount
CRITICAL	-
MAJOR	-
WARNING	5
COMMENT	2

Final commit identifier with fixes:

- mexa master-gasToken: `604660552832a37b6819d4e2474c6709b2236d26`
- mexa master: `7a5e6e20d57b19b4b2695c0b6b4399c5f46da4fa`
- metatx-standard: `8a6564b8fac5d2aceebb1972bcffdd5e88ceec63`

About MixBytes

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

Contacts

https://github.com/mixbytes/audits_public

https://mixbytes.io/

hello@mixbytes.io

https://t.me/MixBytes

Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Biconomy. If you are not the intended recipient(s) of this document, please note that any disclosure, copying or dissemination of its content is strictly forbidden.