Bit2Me Token Security Audit

August 2021

By CoinFabrik

Introduction

CoinFabrik was asked to audit the contracts for the Bit2Me project. First we will provide a summary of our discoveries and then we will show the details of our findings.

Summary

The contracts audited are from the files delivered to us. We transcribe their hashes for later integrity checks:

SHA-1
8255b832c12202390efbca9627abb5ac47060f03 Token.sol
8bb18db88c1c70bf7b20a83ef75ee99b04782e5f TokenManager.sol

SHA-256
3c9780e71238ca6233311cf382f006d18780c97841497dc2dc42ec180446de7a Token.sol
d20e963a9aa9ba862ff014895959e694ad9a94807c8c506cc42ce3f77b4d61c2 TokenManager.sol

Contracts

The audited contracts are:

- contracts/Token.sol: An ERC20 token contract.
- contracts/TokenManager.sol: The token manager.

Analyses

The following analyses were performed:

- Misuse of the different call methods
- Integer overflow errors
- Division by zero errors
- Outdated version of Solidity compiler
- Front running attacks
- Reentrancy attacks
- Misuse of block timestamps
- Softlock denial of service attacks

- Functions with excessive gas cost
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Failure to use a withdrawal pattern
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures

Findings and Fixes

ID	Title	Severity	Status
EN-1	Misleading Documentation	Enhancement	Not fixed

Severity Classification

Security risks are classified as follows:

- **Critical:** These are issues that we manage to exploit. They compromise the system seriously. They must be fixed **immediately**.
- **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them **as soon as possible**.
- **Minor:** These issues represent problems that are relatively small or difficult to take advantage of but can be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed **when possible**.
- **Enhancement:** These kinds of findings do not represent a security risk. They are best practices that we suggest to implement.

This classification is summarized in the following table:

SEVERITY	EXPLOITABLE	ROADBLOCK	TO BE FIXED

Critical	Yes	Yes	Immediately
Medium	In the near future	Yes	As soon as possible
Minor	Unlikely	No	Eventually
Enhancement	No	No	Eventually

Issues Found by Severity

Enhancements

Misleading Documentation

At the top of the TokenManager.sol contract we can find the following comment

```
Only allow burning once per day and less than 5%
```

However, only 1% of the current supply may be burned each time the burn function is called. Bit2Me confirmed that it is correct to have the 1% limit and hence the problem is with the comment.

Also, the `burn()` function can only be called 24hs after it was called last, and not once per day, as the documentation suggests. Again, Bit2Me confirmed that the code reflects their intention, so that the comment is inexact and should be corrected.

Conclusion

We found the contracts to be simple and straightforward and have an adequate amount of documentation. No issues were found other than a problem with the documentation (a comment documenting the `burn()` function to be precise).

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the Bit2Me project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.