Mokens League Audit

March 2022

By CoinFabrik

Introduction

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

Scope

The contracts audited are from the https://github.com/Mokens-League/mokens-league-tokens/ git repository. The audit is based on the commit `b2e3a17f7f8c8d220fc135fc9a8364fdf07d51c`.

The audited contracts are:

- `/contracts/Token.sol`: Main ERC777 Token
- `/contracts/Forwarder.sol`: ERC2771 minimal forwarder
- `/contracts/Policy.sol`: Operator functionality
- `/contracts/TokenMintable.sol`: Adds access control to Token contract
- `/contracts/Mock/MockOperator.sol`: Testing code
- `/contracts/IPolicy.sol`: Policy interface

The scope of the audit is limited to those files. No other files in this repository were audited. Its dependencies are assumed to work according to their documentation. Also, no tests were reviewed for this audit.
Fixes and improvements were checked in commit `a70bda94e5918e1fff53d656c056d2db0b882401`.

Analyses

Without being limited to them, the audit process included the following analyses:

- Arithmetic errors
- Outdated version of Solidity compiler
- Race conditions
- Reentrancy attacks
- Misuse of block timestamps
- Denial of service attacks
- Excessive gas usage
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling

- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures
- Centralization and upgradeability

Summary of Findings

We found 1 medium issue.

We checked fixes in commit `a70bda94e5918e1fff53d656c056d2db0b882401`. All issues were resolved.

Security Issues

ID	Title	Severity	Status
ME-01	Outdated version of the solidity compiler	Medium	Resolved

Privileged Roles

These are the privileged roles that we identified on each of the audited contracts.

TokenMintable

MINTER_ROLE

This role is the only address allowed to mint tokens.

Token

Owner

The owner can change the address designed as the operator. And it is a standard owner as defined in the OpenZeppeling's Ownable contract.

Policy

Owner

The owner can change the address designed as the operator. And it is a standard owner as defined in the OpenZeppeling's Ownable contract.

Security Issues Found

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**.

Issues Status

An issue detected by this audit can have four distinct statuses:

- **Unresolved**: The issue has not been resolved.
- **Acknowledged**: The issue remains in the code but is a result of an intentional decision.
- **Resolved**: Adjusted program implementation to eliminate the risk.
- **Partially resolved**: Adjusted program implementation to eliminate part of the risk. The other part remains in the code but is a result of an intentional decision.
- **Mitigated**: Implemented actions to minimize the impact or likelihood of the risk

Critical Severity Issues

No issues found.

Medium Severity Issues

ME-01 Outdated Version of the Solidity Compiler

All contracts require the solidity compiler major version to be 0.8:

```
pragma solidity ^0.8.0;
```

However, this would allow older compiler versions with important bugs. Particularly, it would allow `solidity <0.8.3` that contains an ABI decoder Bug (see https://blog.soliditylang.org/2021/04/21/decoding-from-memory-bug/)

As the contracts uses the `abi.encode()` function in several places (`Forwarder.sol:35,49`) the seriousness of this issue was upgraded to medium.

Recommendation
Specify the latest version of the solidity compiler (0.8.12 at the time of this report)

Status
Resolved. In commit `a70bda94e5918e1fff53d656c056d2db0b882401`, the solidity compiler version was updated to the recommended version.

Minor Severity Issues

No issues found.

Enhancements

No enhancements found.

Other Considerations

Initial token supply is transferred to the contract owner. Token contract adds 'Core contracts' Operator functionality to ERC777, who have no limitations to transfer tokens from/to any account.

Changelog

- 2022-03-04 – Initial report based on commit `b2e3a17f7f8c8d220fc135fc9a8364fdf07d51c`.
- 2022-03-14 - Fixes checked on commit `a70bda94e5918e1fff53d656c056d2db0b882401`.

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