SAFETIN

AUDIT

MultiFunctional Environmental Token

- Lock.sol

September 22nd, 2022

TABLE OF CONTENTS

AUDIT SUMMARY

This report was written for MultiFunctional Environmental Token in order to find flaws and vulnerabilities in the MultiFunctional Environmental Token project's source code, as well as any contract dependencies that weren't part of an officially recognized library given they were provided.

A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and MultiFunctional Environmental Token Deployment techniques. The auditing process pays special attention to the following considerations:

- ❖ Testing the smart contracts against both common and uncommon attack vectors
- ❖ Assessing the codebase to ensure compliance with current best practices and industry standards
- ❖ Ensuring contract logic meets the specifications and intentions of the client
- ❖ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
- ❖ Through line-by-line manual review of the entire codebase by industry expert

AUDIT OVERVIEW

PROJECT SUMMARY

Project name	MultiFunctional Environmental Token
Description	MFET is an ecosystem that provides consultancy to companies in their environmental works as an environmentally friendly token supporting sustainable projects.
Platform	BNB Smart-chain
Language	Solidity
Codebase	Main.sol https://bscscan.com/address/0x6d23970ce32Cb0F1929bECE7C56D71319e1b4F01 Stake.sol https://bscscan.com/address/0xe0cd1e9820001faff0c98935dae500471f92c6e1 Lock.sol https://bscscan.com/address/0x3b7f0d2BaB984c9dbc70F53Ef0B5C65a8E86c463

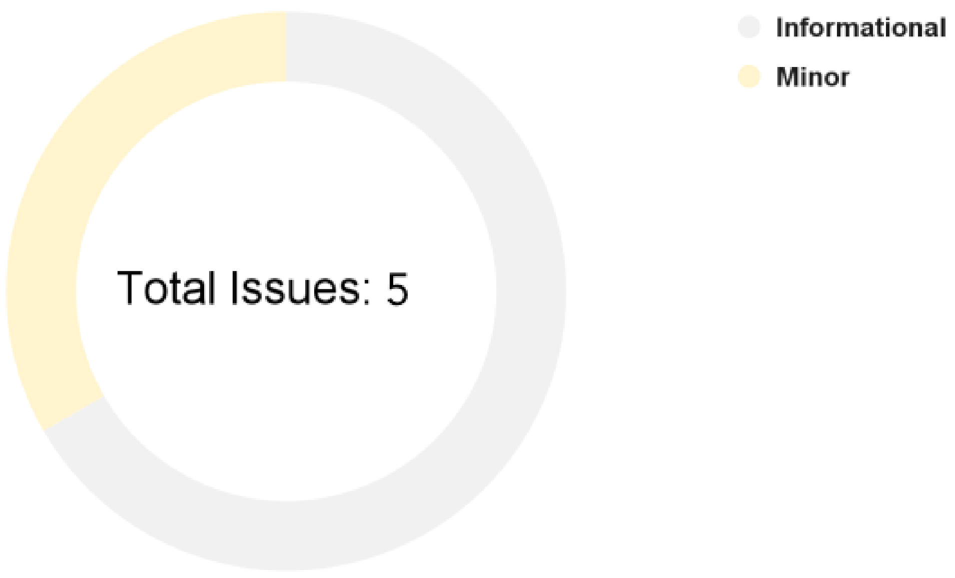
FINDINGS SUMMARY

Vulnerability	Total	Resolved
● Critical	0	0
● Major	0	0
● Medium	0	0
● Minor	1	0
● Informational	4	0

EXECUTIVE SUMMARY

There have been no critical issues related to the codebase and all findings listed here range from informational to minor.

AUDIT FINDINGS



Code	Title	Severity
COMP-1	Unlocked compiler versions	● Minor
FUNC-1	Unused functions	● Informational
GAS-2	Overly long error messages	● Informational
BP-1	Too many zeros	● Informational
MSG-2	Limited NatSpec comments	● Informational

COMP-1 | Unlocked compiler version

Description

MultiFunctional Environmental Token's contract does not have locked compiler versions, meaning a range of compiler versions can be used. This can lead to differing bytecodes being produced depending on the compiler version, which can create confusion when debugging, as bugs may be specific to a specific compiler version(s).

Recommendation

To rectify this, we recommend setting the compiler to a single version, the version tested the most to be compatible with the code, an example of this change can be seen below.

```
pragma solidity 0.8.0;
```

FUNC-1 | Unused functions

Description

Multiple functions within MultiFunctional Environmental Token's contract are defined as private or internal but are never called within the contract. This wastes contract space as there is a maximum size a contract can have. Functions found with this issue have been listed below:

- ❖ Lock.sol
  - ➢ safeTransferFrom -> Line 410
  - ➢ safeTransfer -> Line 399
  - ➢ safePermit -> Line 486
  - ➢ safeIncreaseAllowance -> Line 447
  - ➢ safeDecreaseAllowance -> Line 463
  - ➢ safeApprove -> Line 429
  - ➢ _callOptionalReturn -> Line 511
  - ➢ reset -> Line 556
  - ➢ decrement -> Line 548
  - ➢ _msgData -> Line 609
  - ➢ verifyCallResult -> Line 373
  - ➢ sendValue -> Line 195
  - ➢ isContract -> Line 171
  - ➢ functionStaticCall -> Line 321

- ➢ functionStaticCall -> Line 302
- ➢ functionDelegateCall -> Line 356
- ➢ functionDelegateCall -> Line 338
- ➢ functionCallWithValue -> Line 278
- ➢ functionCallWithValue -> Line 258
- ➢ functionCall -> Line 239
- ➢ functionCall -> Line 226

Recommendation

We recommend safely removing these functions from the contract.

GAS-2 | Overly long error messages

Description

The smart contract has some error messages that are too long. The industry standards specify error messages must have a maximum length of 32 bytes. We recommend having the short error messages within 32 bytes to optimize gas costs. Require statements with this issue have been listed below:

- ❖ Lock.sol
  - ➢ line -> 326
  - ➢ line -> 361
  - ➢ line -> 648

Recommendation

We recommend shortening these error messages to be 32 characters or less in length.

BP-1 | Too many zeros

Description

Some arithmetic operations within MultiFunctional Environmental Token's contract contain integers with too many zeros. This can make it hard to read the code and maintain. This can lead to coding errors such as adding an extra zero or missing one during development The lines within the contract where this issue was identified has been listed below.

- ❖ Lock.sol
  - ➢ require -> Line 788
  - ➢ require -> Line 877
  - ➢ require -> Line 927

Recommendation

Such integers should be represented in standard index form by utilizing powers of 10, for example 3000000 can be represented as 3 * 10**6.

MSG-2 | Limited NatSpec comments

Description

Throughout MultiFunctional Environmental Token’s contracts many functions remain uncommented. This can make understanding the code’s functionality difficult for developers and users (if the code is open source) thus reducing maintainability.

Recommendation

We recommend using NetSpec standard comments throughout all of MultiFunctional Environmental Token’s contracts.

See: https://docs.soliditylang.org/en/v0.5.17/style-guide.html?highlight=natspec%23natspec

Global security warnings

These are safety issues for the whole project. They are not necessarily critical problems but they are inherent in the structure of the project itself. Potential attack vectors for these security problems should be monitored.

Compliance with industry standards

The way the contract is developed and its compliance with industry standards are part of the project. In order to increase the optimization of the latter, we recommend refining the code to best fit industry best practices, in particular the use of error messages and library utilization.

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement.

This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement.

This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without Safetin's prior written consent.This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Safetin to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance. This report should not be used in any way

Safetin security assessment to make decisions around investment or involvement with any particular project.

This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Safetin's position is that each company and individual are responsible for their own due diligence and continuous security. Safetin's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or fun.