HackSafe

Smart Contract Security Audit Report

Metafi

April **2022**

Security Status

★ ★ ★ ★ ★

www.hacksafe.io

Audit Details

Audited project
Metafi

Deployer address
0x3445C9423C5728a158d9e90142eEc3851550897C

Client contacts
METAFI Token team

Blockchain
Binance smartchain

Website
www.metafi.org

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and HackSafe hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

HeckSafe was commissioned by Metafi to perform an audit of smart contracts:

- https://bscscan.com/address/0x8f62C11673EF7F8b31111Da018322b2DF7147855

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issue with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 08.04.2022

Contract name	: METAFI
Contract address	: 0x8f62C11673EF7F8b31111Da018322b2DF7147855
Total supply	: 1 billion
Token Ticker	: METAFI
Decimals	: 18
Network	: Binance Smart Chain
Transactions count	: 1
Token Holders	: 1 addresses
Contract deployer address	: 0x3445C9423C5728a158d9e90142eEc3851550897C

Contract functions details

BurnableERC20.sol
+ BurnableERC20 (ERC20Burnable)
    -<constructor> #

ERC20Burnable.sol
+ ERC20Burnable (context, ERC20)
    +[Pub] burn#
    +[Pub] burnFrom#

ERC20.sol
+ERC20 (Context, IERC20, IERC20Metadata)
    +<constructor >#
    +[Pub] name
    +[Pub] symbol
    +[Pub] decimals
    +[Pub] totalSupply
    +[Pub] balanceOf
    +[Pub] transfer#
    +[Pub] allowance
    +[Pub] approve#
    +[Pub] transferFrom#
    +[Pub] increaseAllowance#
    +[Pub] decreaseAllowance#
    +[Int] _transfer#
    +[Int] _mint#
    +[Int] _burn#
    +[Int] _approve#
    +[Int] _beforeTokenTransfer
    +[Int] _afterTokenTransfer

Context.sol
+ Context
    +[Int] msgSender
    +[Int] _msgData

Contract functions details

IERC20.sol
+[Int] IERC20
    -[Ext] totalSupply
    -[Ext] balanceOf
    -[Ext] transfer#
    -[Ext] allowance
    -[Ext] approve#
    -[Ext] transferFrom#

IERC20Metadata.sol
+[Int] IERC20Metadata (IERC20)
    +[Ext] name
    +[Ext] symbol
    +[Ext] decimals

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Low issue
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

- **Critical Severity Issues**
  No critical severity issue found.

- **High Severity Issues**
  No high severity issue found.

- **Medium Severity Issues**
  No Medium severity issue found.

- **Low Severity Issues**
  Three low severity issue found.

1. Unlocked Compiler Version.

- **Description**

  The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

  It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version v0.8.0 the contract should contain the following line:

  pragma solidity 0.8.0;

2. Unused function.

- **Description**

  The _msgData function does nothing.

- **Location**

  Context.sol -> _msgData function

- **Recommendation**

  We advise to remove unused code.

Security Issues

1. Unused function.

- **Description**

  The _mint function does nothing.

- **Location**

  ERC20.sol -> _mint function

- **Recommendation**

  We advise to remove unused code.

Conclusion

Smart contract contains low severity issues!

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.