verichains

SECURITY AUDIT OF

MYTHERIA SMART CONTRACT

Public Report

Nov 26, 2021

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report prepared by Verichains Lab on Nov 26, 2021. We would like to thank the Mytheria for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Mytheria Smart Contract. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY 5

1.1. About Mytheria Smart Contract 5

1.2. Audit scope 5

1.3. Audit methodology 5

1.4. Disclaimer 6

2. AUDIT RESULT 7

2.1. Overview 7

2.2. Contract codes 7

2.2.1. IERC20 interface 7

2.2.2. IERC20Metadata interface 7

2.2.3. Context abstract contract 8

2.2.4. Ownable abstract contract 8

2.2.5. ERC20 contract 8

2.2.6. Pausable abstract contract 8

2.2.7. ERC20Snapshot abstract contract 8

2.2.8. ERC20Burnable abstract contract 8

2.2.9. MYRA contract 8

2.3. Findings 9

2.4. Additional notes and recommendations 9

2.4.1. Unnecessary usage of SafeMath library in Solidity 0.8.0+ INFORMATIVE 9

3. VERSION HISTORY 11

1. MANAGEMENT SUMMARY

1.1. About Mytheria Smart Contract

Mytheria is the first pioneering NFT game offering an exclusive Create-To-Earn community for artists including you, with an name of artist community: GODFORGE.

MYRA is an ERC20 token that Mytheria players can use in the game.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Mytheria Smart Contract.

The audited contract is the Mytheria Smart Contract that deployed on Binance Smart Chain Mainnet at address 0x6ef238e9e8cd2a96740897761c18894fc086b9d0. The details of the deployed smart contract are listed in Table 1.

FIELD	VALUE
Contract Name	MYRA
Contract Address	0x6ef238e9e8cd2a96740897761c18894fc086b9d0
Compiler Version	v0.8.2+commit.661d1103
Optimization Enabled	Yes with 200 runs
Explorer	*https://bscscan.com/address/0x6ef238e9e8cd2a96740897761c18894fc086b9d0*

Table 1. The deployed smart contract details

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 2. Severity levels

1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The initial review was conducted in Nov 23, 2021 and a total effort of 3 working days was dedicated to identifying and documenting security issues in the code base of Mytheria Smart Contract.

Table 2 lists some properties of the audited Mytheria Smart Contract (as of the report writing time).

PROPERTY	VALUE
Name	Mytheria
Symbol	MYRA
Decimals	18
Total Supply	200,000,000 ($x10^{18}$) Note: the number of decimals is 18, so the total representation token will be 200,000,000 or 200 million.

Table 3. The Mytheria Smart Contract properties

2.2. Contract codes

The Mytheria Smart Contract was written in Solidity language, with the required version to be 0.8.2.

The source codes consist of two contracts, five abstract contracts, two interfaces. Almost all source codes in the Mytheria Smart Contract imported OpenZeppelin contracts.

2.2.1. IERC20 interface

This is the interface of the ERC-20 token standard. The source code is referenced from OpenZeppelin's implementation.

2.2.2. IERC20Metadata interface

This is the interface that extends IERC20 interface to add functions for interacting with some metadata. The source code is referenced from OpenZeppelin's implementation.

2.2.3. Context abstract contract

This abstract contract provides information about the current execution context, including the sender of the transaction and its data. The source code is referenced from OpenZeppelin's implementation.

2.2.4. Ownable abstract contract

This abstract contract makes the Mytheria Smart Contract ownable. The source code is referenced from OpenZeppelin's implementation.

2.2.5. ERC20 contract

This is the abstract contract implement ERC20 token. The source code is referenced from OpenZeppelin's implementation.

2.2.6. Pausable abstract contract

Contract module allows children to implement an emergency stop mechanism that can be triggered by an authorized account. The source code is referenced from OpenZeppelin's implementation.

2.2.7. ERC20Snapshot abstract contract

This abstract contract extends an ERC20 token with a snapshot mechanism. When a snapshot is created, the balances and total supply at the time are recorded for later access.

2.2.8. ERC20Burnable abstract contract

This abstract contract extends ERC20 abstract contract that allows token holders to destroy their tokens. The source code is referenced from OpenZeppelin's implementation.

2.2.9. MYRA contract

This is the main contract in the MYRA Smart Contract, which extends ERC20, ERC20Burnable, ERC20Snapshot, Pausable, Ownable abstract contract. Almost all functions in the MYRA Smart Contract are inherited, some functions are overridden by composing between the old function and the modifiers.

In addition, this contract has a mechanism to block Bot transfer in the IDO with maxTransferWhenAntiBot value. After a short time of IDO, contract owner can disable antibot by setAntiBot function.

2.3. Findings

During the audit process, the audit team found no vulnerability in the given version of Mytheria Smart Contract.

2.4. Additional notes and recommendations

2.4.1. Unnecessary usage of SafeMath library in Solidity 0.8.0+ INFORMATIVE

All safe math usage in the contract are for overflow checking, solidity 0.8.0+ already checked that by default, the only usage of safemath now is to have a custom revert message which isn't the case in the auditing contracts.

RECOMMENDATION

We suggest changing all methods from SafeMath library to normal arithmetic operator in the contract for readability and gas saving.

APPENDIX

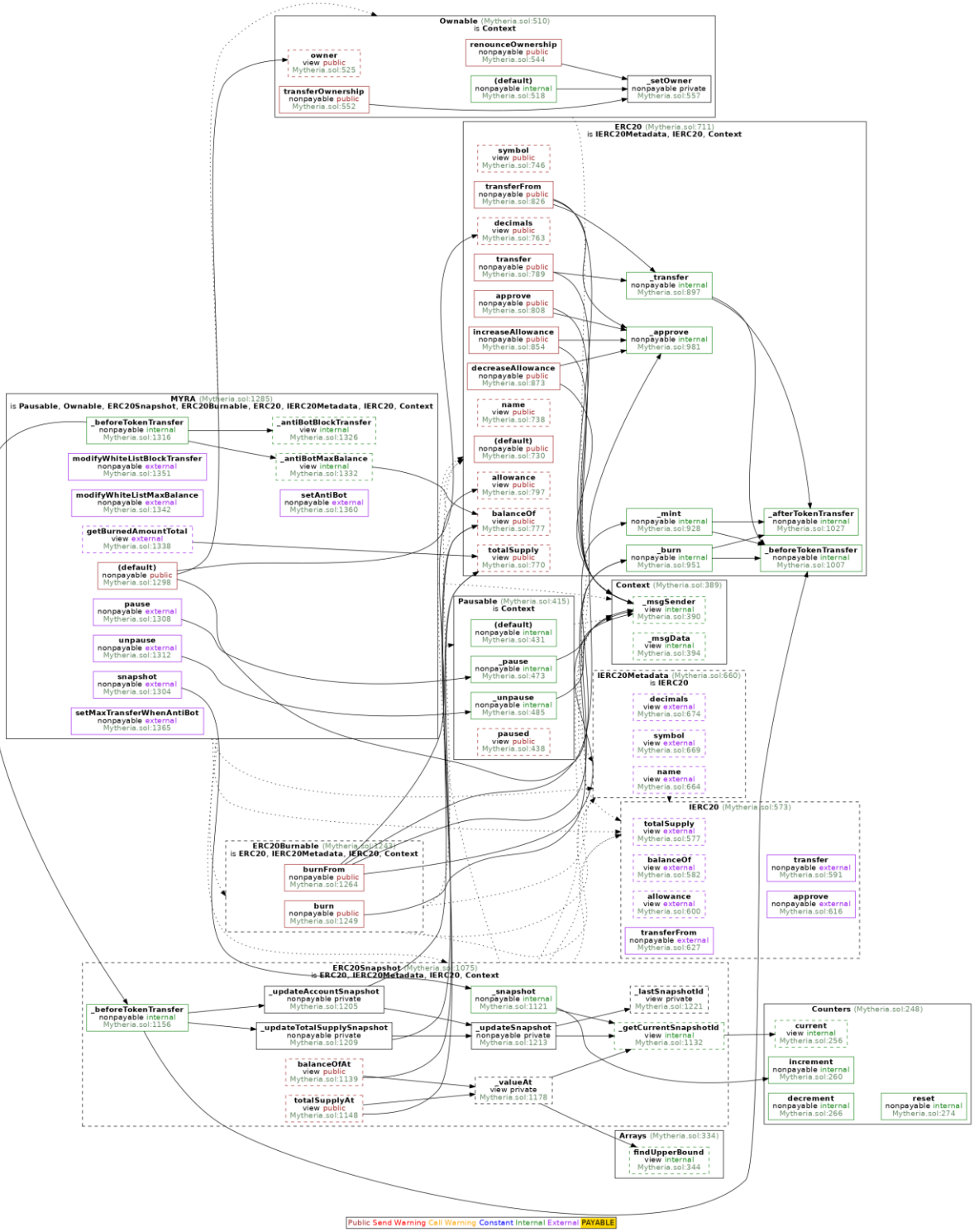

Image 1. Mytheria Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Nov 26, 2021*	Public Report	Verichains Lab

Table 4. Report versions history