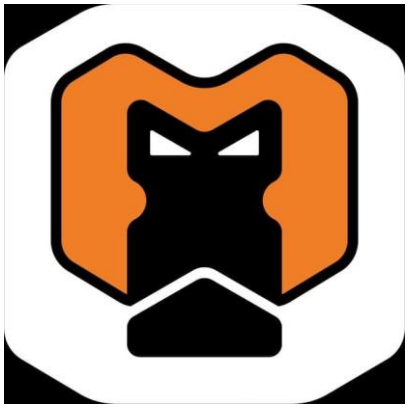
verichains

SECURITY AUDIT OF

MAD MONKEY GUILD TOKEN SMART CONTRACT



Public Report

Apr 15, 2022

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 Apr 15, 2022. We would like to thank the Mad Monkey Guild 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 Mad Monkey Guild Token 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

1.1. About Mad Monkey Guild Token Smart Contract

Mad Monkey Guild is a play-to-earn (P2E) gaming guild that aims to create job opportunities and rewarding experiences for our community members.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the smart contracts of Mad Monkey Guild Token Smart Contract. It was conducted on the source code provided by the Mad Monkey Guild team.

The latest version of the following files were made available in the course of the review:

SHA256 Sum	File
1ef58cb9d9183acf9895d5a55747dda28d46319e7aae7f640639082f8e6a7ee6	MadMonkeyGuide.sol

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 1. 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

2.2. Contract codes

The Mad Monkey Guild Token Smart Contract was written in Solidity language, with the required version to be 0.8.2.

2.3. MadMonkeyGuide contract

MadMonkeyGuide token contract extends ERC20, ERC20Snapshot, Pausable, ERC20Burnable, Ownable contracts. With Ownable, by default, Token Owner is contract deployer but he can transfer ownership to another address at any time. He can pause/unpause contract using Pausable contract, users can only transfer tokens when contract is not paused. The ERC20Burnable allows token holders to destroy both their own tokens and those that they have an allowance for. The ERC20Snapshot help Token Owner take a snapshot of the balances and total supply at a time for later access.

For each transfer transaction, a small amount of token may be sent to taxAddress. Only owner of contract may change the rate of transfer tax through setTaxFeePercent function.

Table 2 lists some properties of the audited Mad Monkey Guild Token Smart Contract (as of the report writing time).

PROPERTY	VALUE
Name	Mad Monkey Guild
Symbol	MMG
Decimals	18
Total Supply	88,000,000 ($x10^{18}$) Note: the number of decimals is 18, so the total representation token will be 88,000,000 or 88 million.

Table 2. The Mad Monkey Guild Token Smart Contract properties

2.4. Findings

During the audit process, the audit team found no vulnerability in the given version of Mad Monkey Guild Token Smart Contract.

APPENDIX

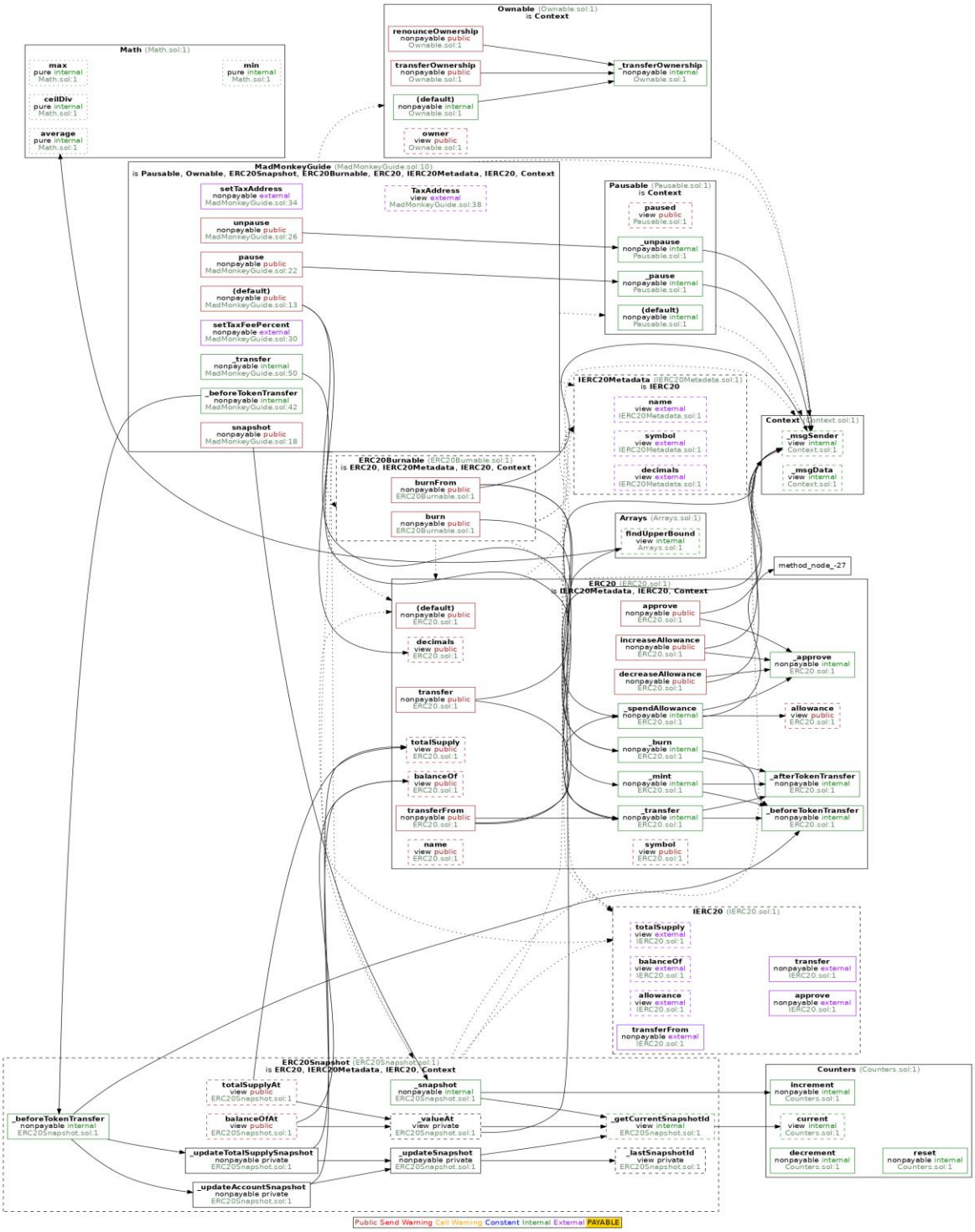


Image 1. Mad Monkey Guild Token Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Apr 15, 2021*	Public Report	Verichains Lab

Table 3. Report versions history