verichains

SECURITY AUDIT OF

METADOS SMART CONTRACTS

Public Report

Feb 21, 2024

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 was prepared by Verichains Lab on Feb 21, 2024. We would like to thank the DOS Labs 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 MetaDOS Smart Contracts. 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 identified no vulnerable issues but made several recommendations regarding security best practices in the contract code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About MetaDOS Smart Contracts

MetaDOS is an upgradeable ERC-1155 token with additional functionalities such as minting, buying, redeeming, and exchanging tokens.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the MetaDOS Smart Contracts. It was conducted on commit `0e90c29675374c9af2f5a94a26e415a50d2e2f60` from git repository link: *https://github.com/DOSLabs/Smart-Contracts/tree/main/auditsV4*.

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 were 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

DOS Labs acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. DOS Labs understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, DOS Labs agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.5. Acceptance Minute

This final report served by Verichains to the DOS Labs will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the DOS Labs, the final report will be considered fully accepted by the DOS Labs without the signature.

2. AUDIT RESULT

2.1. Overview

The MetaDOS Smart Contracts was written in `Solidity` language, with the required version to be `^0.8.20`.

The MetaDos contract encapsulates the core functionalities of the MetaDos protocol, facilitating actions such as token burning, minting, purchasing, redemption, and exchange of MetaDos tokens and NFTs.

Users are granted access via off-chain administration signatures, enabling them to mint, buy, and burn MetaDos tokens and NFTs through a one-time valid signature. Additionally, users possess the capability to redeem MetaDos tokens, NFTs by burning that and minting new tokens or NFTs. Furthermore, they can exchange it.

Moreover, the contract empowers the owner to designate new bridges, signers, and operators. Operators are entrusted with the authority to blacklist users or lock their balances and NFTs, although these functionalities remain dormant within the protocol.

The MetaDos Proxy contract serves as a Transparent Upgradeable Proxy, facilitating protocol upgrades without necessitating a change in the contract's address, with the MetaDos contract constituting the underlying logic contract of the MetaDos protocol.

2.2. Findings

During the audit process, the audit team identified no vulnerable issues but made several recommendations regarding security best practices in the contract code.

2.2.1. Contract Initialization Requirement - INFORMATIVE

The MetaDos contract serves as the logic contract for a Transparent Upgradeable Proxy. It is imperative that the contract be properly initialized during deployment. This initialization process should occur only once, be invoked by the contract owner, take place post-deployment, and precede any other function calls. Additionally, it is crucial that the initialization function be executed with the correct parameters.

It's essential to note that since the transaction to initialize the contract is public, anyone can potentially call the initialization function. If the initialization function has not been called yet, an attacker could exploit this vulnerability by calling the initialization function and altering the contract's state. Therefore, strict adherence to initialization protocols is paramount to maintaining the contract's integrity and security posture.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Feb 21, 2024*	Public Report	Verichains Lab

Table 2. Report versions history