verichains

SECURITY AUDIT OF

LAKRIMA TOKEN SMART CONTRACT

Public Report

Mar 21, 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 Mar 21, 2022. We would like to thank the ECIO 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 Lakrima 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 issue in the contract code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Lakrima Token Smart Contract

ECIO is filled with fun and exciting gameplay elements & an immersive gaming experience with story-rich gameplay. Aside from the game itself, the ECIO ecosystem will be complemented with a Comic amongst others, which will reveal the background story of our game. Discover how the different races are fighting an intense battle for freedom & power. In a galaxy engulfed in war, you must train yourself to be the deadliest warrior.

As a first in blockchain gaming, ECIO allows you to join forces with a friend and expand your Space Crew from 3 to 6 members. Challenge other crew in the team arena or battle in survival mode to earn more rewards.

ECIO's goal is to deliver the best gaming experience for players. We focus on the entertainment factors of the gameplay which will reflect in the sustainable economic growth. Through this, ECIO aims to reach a long-term business interest for everyone involved.

1.2. Audit scope

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

The following file were made available in the course of the review:

SHA256 Sum	File
fb83d8551c52c32a370e18ecc1d202bff3e36fedebc7802c0344555a703dd6aa	Lakrima.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

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

PROPERTY	VALUE
Name	Lakrima
Symbol	LKM
Decimals	18
Total Supply	Unlimited

Table 2. The Lakrima Token Smart Contract properties

2.2. Contract codes

The Lakrima Token Smart Contract was written in Solidity language, with the required version to be 0.8.7. Almost all source codes in the Lakrima Token Smart Contract are imported from OpenZeppelin contracts.

The contract extends ERC20, ERC20Burnable, AccessControl and Pausable contracts. AccessControl allows the contract to implement role-based access control mechanisms which add token owner (contract deployer) DEFAULT_ADMIN_ROLE, PAUSER_ROLE and MINTER_ROLE roles. The user who has PAUSER_ROLE can pause/unpause the contract using Pausable contract, the users can only transfer tokens when the contract is not paused.

The contract also implements mint public function which allows MINTER_ROLE to create new tokens.

2.3. Findings

During the audit process, the audit team had identified no vulnerable issue in the contract code.

2.4. Additional notes and recommendations

2.4.1. Lakrima.sol - BPContract function INFORMATIVE

Since we do not control the logic of the BPContract, there is no guarantee that BPContract will not contain any security related issues. With the current context, in case the BPContract is

compromised, there is not yet a way to exploit the Lakrima Token Smart Contract, but we still note that here as a warning for avoiding any related issue in the future.

By the way, if having any issue, the BPContract function can be easily disabled anytime by the contract owner using the setBpEnabled function. In addition, BPContract is only used in a short time in token public sale IDO then the contract owner will disable it forever by the setBotProtectionDisableForever function.

APPENDIX

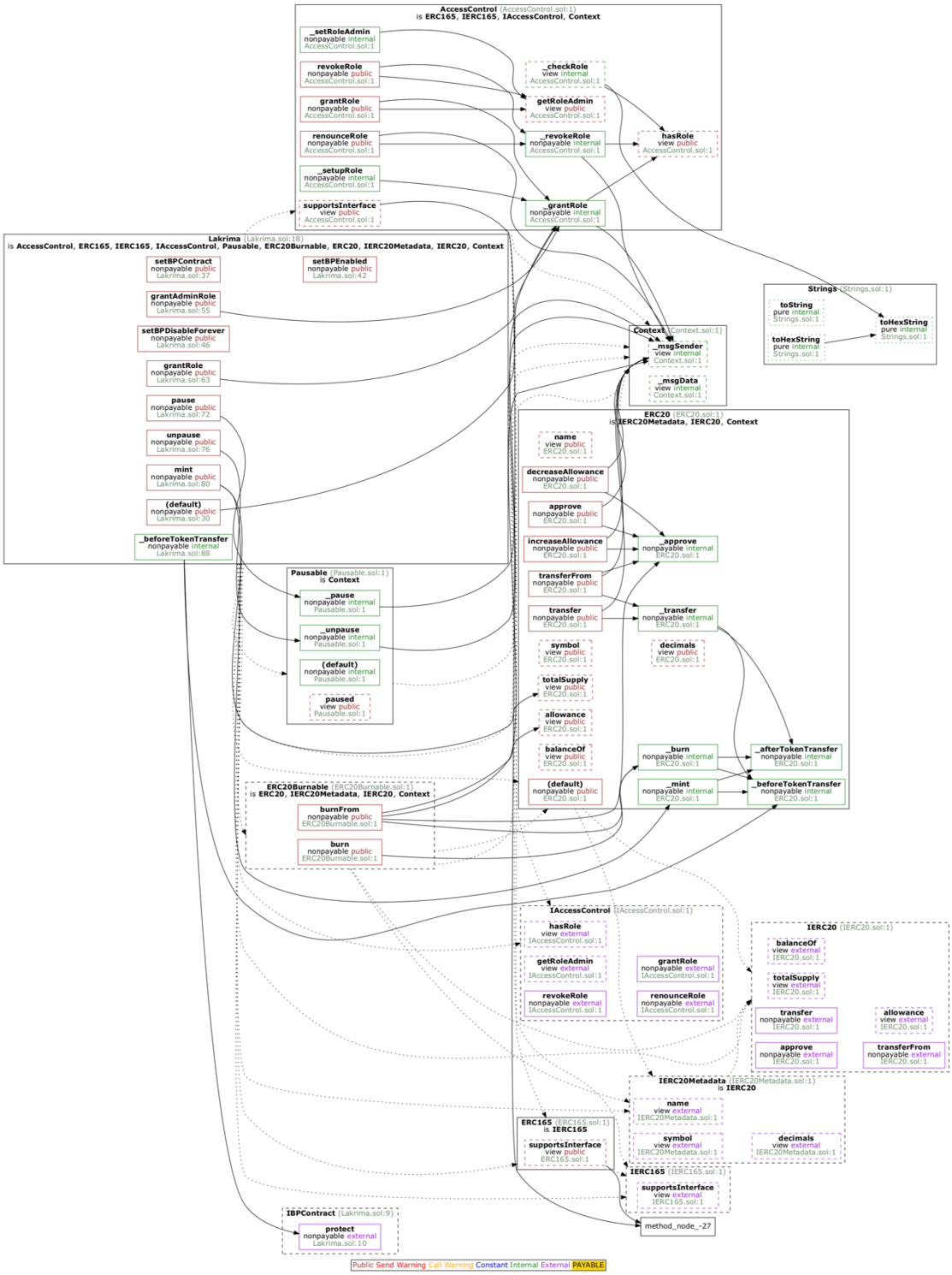

Image 1. Lakrima Token Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Jan 31, 2022*	Public Report	Verichains Lab
1.1	*Mar 21, 2022*	Public Report	Verichains Lab

Table 3. Report versions history