



verichains

SECURITY AUDIT OF

KINGDOM RAIDS TOKEN SMART

CONTRACTS



Public Report

Nov 23, 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 xRP20 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 23, 2021. We would like to thank the Kingdom Raids 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 Kingdom Raids Token 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 had identified no vulnerable issues in the smart contracts code.



TABLE OF CONTENTS

| | |
|---|-----------|
| 1. MANAGEMENT SUMMARY | 5 |
| 1.1. About Kingdom Raids Token Smart Contracts | 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 |
| 2.3. Findings | 7 |
| 2.4. Additional notes and recommendations | 8 |
| 2.4.1. Unnecessary usage of SafeMath library in Solidity 0.8.0+ INFORMATIVE | 8 |
| 3. VERSION HISTORY | 10 |

1. MANAGEMENT SUMMARY

1.1. About Kingdom Raids Token Smart Contracts

Kingdom Raids is an RPG developed by a leading gaming studio, Alley Labs which has attracted more than 100 million downloads for their games. The game takes place in a fictional kingdom, “Dood Kingdom”, a land of mystery and adventure.

Kingdom Raids Token is an ERC20 token that Kingdom Raids players can use in the game.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the smart contracts of Kingdom Raids Token. It was conducted on commit [9fe6b7976fca754e245986767930f6b4a2f48754](https://github.com/kingdomraids/kr-token/commit/9fe6b7976fca754e245986767930f6b4a2f48754) from git repository <https://github.com/kingdomraids/kr-token/>.

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

The initial review was conducted in Nov 2021 and a total effort of 5 working days was dedicated to identifying and documenting security issues in the code base of Kingdom Raids Token.

Table 2 lists some properties of the audited Kingdom Raids Token Smart Contracts (as of the report writing time).

| PROPERTY | VALUE |
|---------------------|---|
| Name | Kingdom Raids Token |
| Symbol | KRS |
| Decimals | 18 |
| Total Supply | 1,000,000,000 ($\times 10^{18}$) Note: the number of decimals is 18, so the total representation token will be 1,000,000,000 or 1 billion. |

Table 2. The Kingdom Raids Token Smart Contracts properties

2.2. Contract codes

The Kingdom Raids Token Smart Contracts was written in [Solidity](#) language, with the required version to be [0.8.0](#).

The Kingdom Raids uses [KRToken](#) contract to initial [Kingdom Raids Token](#) and release the tokens by the contracts correspond to categories in tokenomics.

There are eight contracts that the Kingdom Raids uses to release the tokens. They are [Advisor](#), [CompanyReserves](#), [EcosystemFund](#), [Liquidity](#), [Marketing](#), [PrivateSale](#), [SeedSale](#) and [Team](#) contracts.

2.3. Findings

During the audit process, the audit team found no vulnerability in the given version of Kingdom Raids Token Smart Contracts.

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 do 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. We suggest using normal operators for readability and gas saving.

Currently, the methods of safemath are used in [Advisor](#), [CompanyReservers](#), [EcosystemFund](#), [Liquidity](#), [Marketing](#), [PrivateSale](#), [SeedSale](#), [Team](#) contracts.

RECOMMENDATION

We suggest changing all methods from [SafeMath](#) library to normal arithmetic operator in the contracts that we regarded above.

Report for Kingdom Raids

Security Audit – Kingdom Raids Token Smart Contracts

Version: 1.0 – Public Report

Date: Nov 23, 2021



APPENDIX

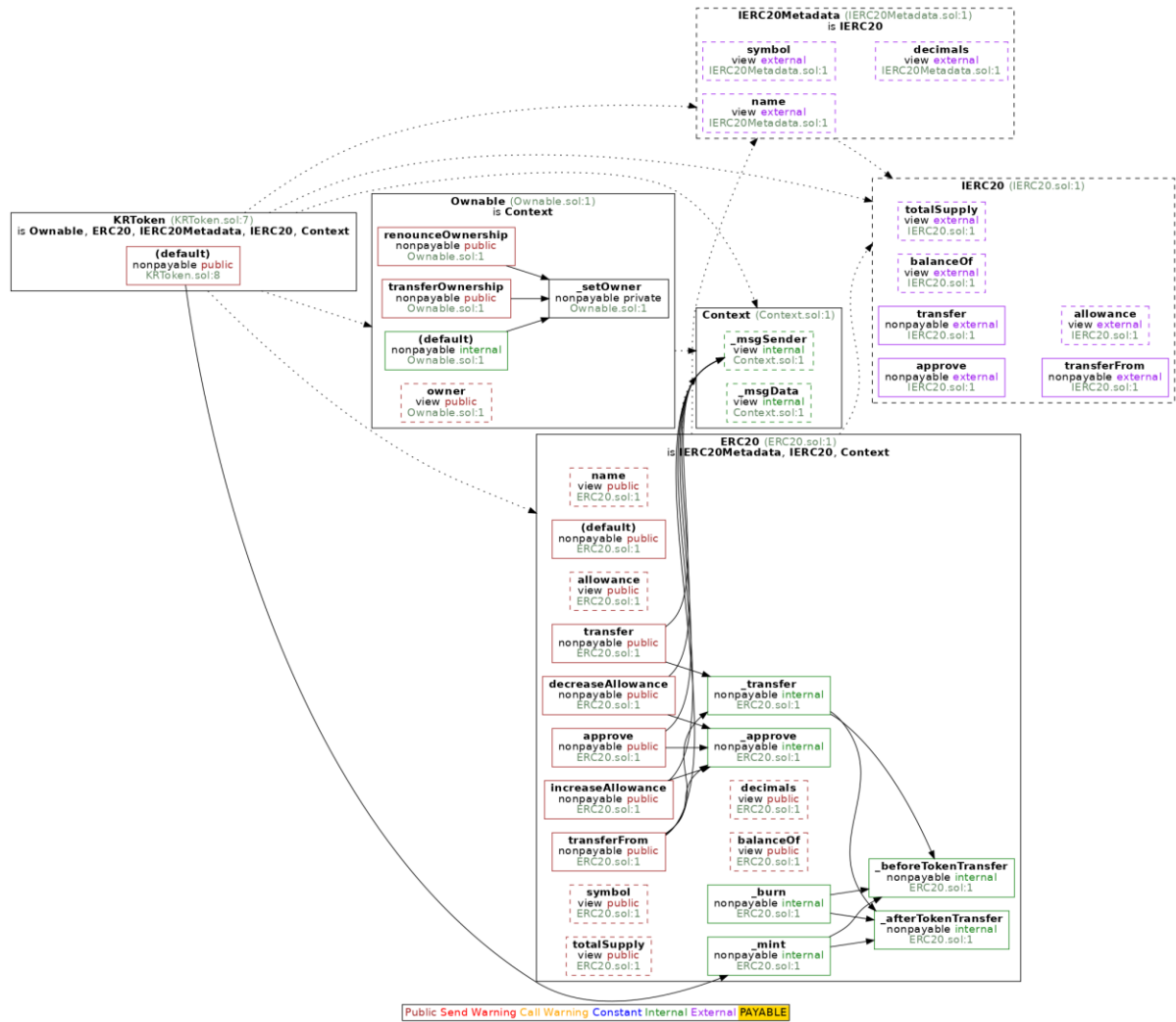


Image 1. Kingdom Raids Token Smart Contracts call graph

Report for Kingdom Raids

Security Audit – Kingdom Raids Token Smart Contracts

Version: 1.0 – Public Report

Date: Nov 23, 2021



3. VERSION HISTORY

| Version | Date | Status/Change | Created by |
|---------|------------|---------------|----------------|
| 1.0 | 2021-11-23 | Public Report | Verichains Lab |

Table 3. Report versions history