verichains

SECURITY AUDIT OF

KINGDOM RAIDS NFT SMART CONTRACTS

Public Report

Nov 29, 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 29, 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 NFT 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

1.1. About Kingdom Raids NFT 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.

The Kingdom Raids NFT Smart Contracts are the contracts which is responsible for managing the hero tokens 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 9cbd2e910cecbca5ce1e62372d255fbc4afa4c6b from git repository *https://github.com/kingdomraids/kr-nft/*.

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 NFT contracts.

2.2. Contract codes

The Kingdom Raids NFT Smart Contracts was written in Solidity language, with the required version to be 0.8.0.

There are two main contracts in the audit scope. They are Hero contract and Summon contract.

The Hero contract is an ERC721 contract which is responsible for all main activities about hero token - the unique tokens in Kingdom Raids. The suppyLimit of contract is 10000 that means there are only 10000 hero summoned in the game. To easily mint a new hero following the logic game, the Hero contract approves MINTER_ROLE for the Summon contract.

The Summon contract is responsible for summoning hero in the game. All data that users use to call the summon function will be verified through signature. For each valid summon function call, the Summon contract will call to the Hero contract for minting a hero token.

2.3. Findings

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

2.4. Additional notes and recommendations

2.4.1. Unused ERC721URIStorage abstract contract INFORMATIVE

ERC721URIStorage is an abstract contract which allows contract to set URI for specific tokenID through _setTokenURI internal function. But no contract which inherits the ERC721URIStorage abstract contract implements the function uses _setTokenURI internal function.

RECOMMENDATION

We suggest removing ERC721URIStorage abstract contract in ExtendERC721.sol file and all statements which interact with this contract.

UPDATES

- *2021-11-29*: This issue has been acknowledged and fixed by the Kingdom Raids team in commit 5fddac09e066101b6e9aa73864ec6ebcc5d0f7ef.

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

In Hero contract and Summon contract, the Safemath library is only used for covering uint256 that is an unnecessary usage. All safe math usages in the contract are for overflow checking, solidity 0.8.0+ already do that by default.

RECOMMENDATION

We suggest removing SafeMath from those contract for gas-saving and readability(including the codes which uses SafeMath for uint256)

UPDATES

- *2021-11-29*: This issue has been acknowledged and fixed by the Kingdom Raids team in commit 5fddac09e066101b6e9aa73864ec6ebcc5d0f7ef.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*2021-11-24*	Public Report	Verichains Lab
1.1	*2021-11-29*	Public Report	Verichains Lab

Table 2. Report versions history