



verichains

*SECURITY AUDIT OF*

**INFINITY ARENA TOKEN SMART  
CONTRACT**



**Public Report**

*Jan 25, 2022*

**Verichains Lab**

[info@verichains.io](mailto:info@verichains.io)

<https://www.verichains.io>

*Driving Technology > Forward*

## ABBREVIATIONS

Name	Description
<b>Ethereum</b>	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
<b>Ether (ETH)</b>	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
<b>Smart contract</b>	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
<b>Solidity</b>	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
<b>Solc</b>	A compiler for Solidity.
<b>ERC20</b>	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 Jan 25, 2022. We would like to thank the Infinity Arena 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 Infinity Arena 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

<b>1. MANAGEMENT SUMMARY.....</b>	<b>5</b>
<b>1.1. About Infinity Arena Token Smart Contract.....</b>	<b>5</b>
<b>1.2. Audit scope .....</b>	<b>5</b>
<b>1.3. Audit methodology.....</b>	<b>5</b>
<b>1.4. Disclaimer .....</b>	<b>6</b>
<b>2. AUDIT RESULT .....</b>	<b>7</b>
<b>2.1. Overview .....</b>	<b>7</b>
<b>2.2. Contract codes.....</b>	<b>7</b>
<b>2.3. Findings .....</b>	<b>7</b>
<b>3. VERSION HISTORY .....</b>	<b>9</b>

# 1. MANAGEMENT SUMMARY

## 1.1. About Infinity Arena Token Smart Contract

Infinity Arena is a metaverse game which is built with lots of NFTs. Owning NFT you can enter the game for mining Zeronium or fighting in the battle. Owning Land and Zeronium can also help you to enter the Planet Dao to vote for how your planet will work.

InfinityArenaZeronium is an ERC20 token that Infinity Arena players can use in the game.

## 1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Infinity Arena Token Smart Contract. It was conducted on the source code provided by the Infinity Arena team.

## 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
<b>CRITICAL</b>	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
<b>HIGH</b>	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
<b>MEDIUM</b>	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
<b>LOW</b>	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 Infinity Arena Token Smart Contract (as of the report writing time).

PROPERTY	VALUE
<b>Name</b>	Infinity Arena Zeronium
<b>Symbol</b>	INAZ
<b>Decimals</b>	18
<b>Total Supply</b>	10,000,000,000 ( $\times 10^{18}$ ) Note: the number of decimals is 18, so the total representation token will be 10,000,000,000 or 10 billion.

*Table 2. The Infinity Arena Token Smart Contract properties*

### 2.2. Contract codes

The Infinity Arena Token Smart Contract was written in **Solidity** language, with the required version to be **0.8.10**.

The contract extends **ERC20**, **ERC20Burnable**, **Ownable** and **ERC20Pausable** contracts. With **Ownable**, by default, Token Owner is contract deployer but he can transfer ownership to another address at any time. **ERC20Burnable** allows token holders to destroy both their own tokens and those that they have an allowance for. Token Owner can pause/unpause contract using **ERC20Pausable** contract, user can only transfer tokens when contract is not paused. The contract implements the **batchApprove** function which allow an **user** to **approve** serveral **users** in a function call.

### 2.3. Findings

During the audit process, the audit team found no vulnerability in the given version of Infinity Arena Token Smart Contract.

# Report for Infinity Arena

## Security Audit – Infinity Arena Token Smart Contract

Version: 1.0 – Public Report

Date: Jan 25, 2022



### APPENDIX

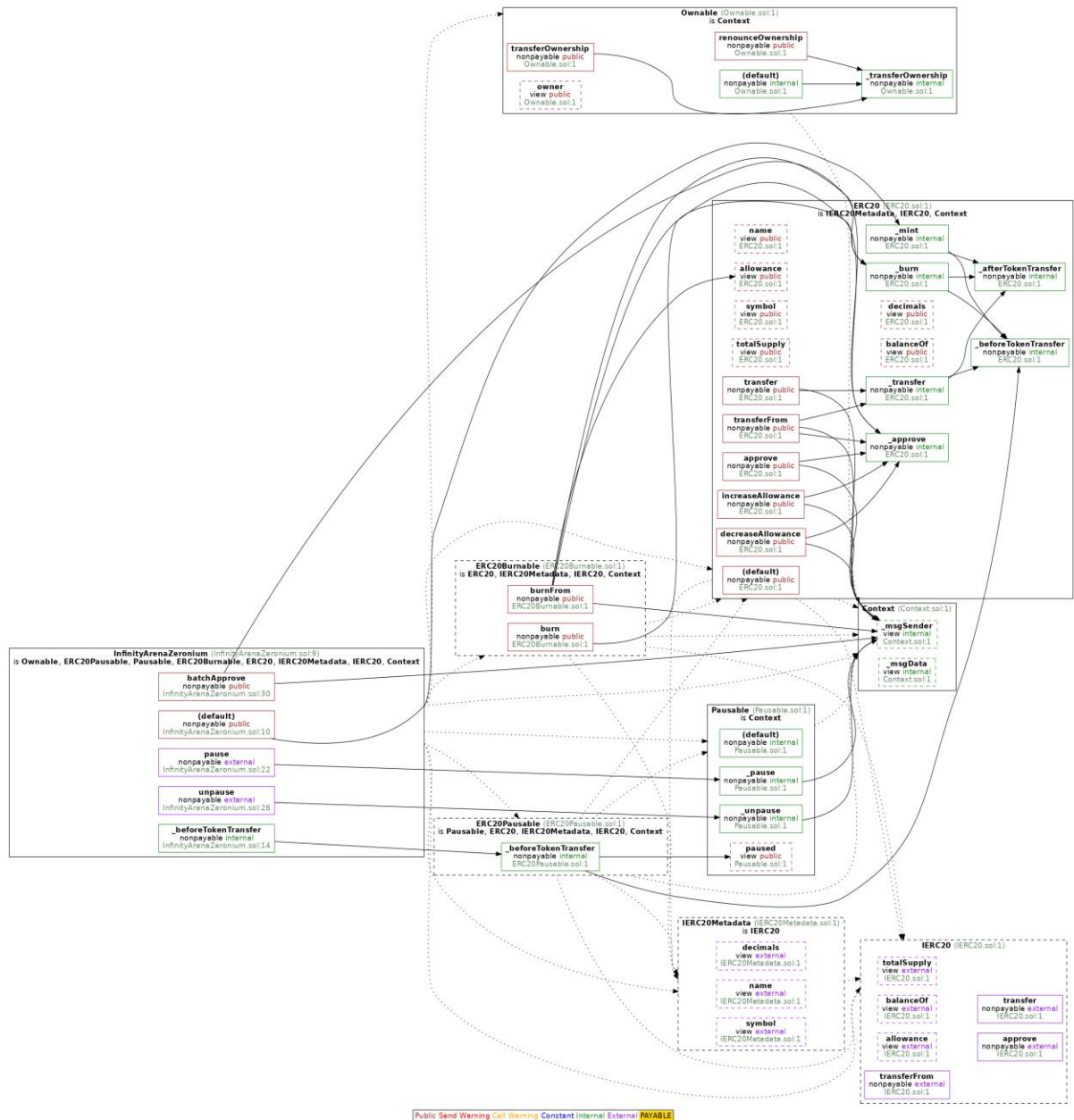


Image 1. Infinity Arena Token Smart Contract call graph



## Report for Infinity Arena

### Security Audit – Infinity Arena Token Smart Contract

Version: 1.0 – Public Report

Date: Jan 25, 2022



## 3. VERSION HISTORY

Version	Date	Status/Change	Created by
<b>1.0</b>	<i>Jan 25, 2021</i>	Public Report	Verichains Lab

*Table 3. Report versions history*