verichains

SECURITY AUDIT OF

SENSPARK TOKEN

Public Report

Apr 11, 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 Apr 11, 2024. We would like to thank the Senspark 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 Senspark Token. 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 smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Senspark Token

Senspark Token within the Senspark GameFi metaverse encompasses the economic principles governing the supply, distribution, and utility of tokens within our gaming ecosystem. By integrating elements of decentralized finance (DeFi) into Senspark's platform, they aim to create a dynamic economy that incentivizes user participation, fosters community engagement, and enhances the overall gaming experience.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Senspark Token that was deployed on Polygon.

The latest version was made available in the course of the review:

FIELD	VALUE
Address Deploy	0xFe302B8666539d5046cd9aA0707bB327F5f94C22
Tx Deploy	0xd1821650585bbab35011212832d7c54e9630e11657625fba6f1aea9c46757508
Deployer	0x050F559cD756cA09FC46988B6cf19ebF01256268
Block Number	55692351

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

Senspark 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. Senspark understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, Senspark 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 Senspark will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the Senspark, the final report will be considered fully accepted by the Senspark without the signature.

2. AUDIT RESULT

2.1. Overview

The Senspark Token was written in `Solidity` language, with the required version to be `0.8.20`.

The contract makes use of the `OpenZeppelin` library's `ERC20` extension. Below is the contract's properties:

PROPERTY	VALUE
Name	Senspark
Symbol	SEN
Decimals	18
Total Supply	$10,000,000,000x10^{18}$ (It represents 10 billion tokens)

Table 2. The Senspark Token properties

For the ERC20 token, the security audit team has the following checklist of standard protocols:

Title	Status
Total Supply Consistency	Passed
Approval	Passed
Self Transfer	Passed
Transfer from/to Zero	Passed
Transfer Effective	Passed

Table 3. The standard testing for ERC20

2.2. Findings

During the audit process, the audit team found no vulnerability in the given version of Senspark Token.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Apr 11, 2024*	Public Report	Verichains Lab

Table 4. Report versions history