

# Smart Contract Security Audit

**AUDIT RATE TECH**

**for**

**SurvivalCoin**



## ***Audit details:***

Audited project: SurvivalCoin

Contract address: 0x7593301f91f5f24953c8aD7a32Aa300D2Ec215f1

Languages: Solidity (Smart contract)

Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Solhint, VScode, Mythril, Contract Library

Total supply: 100,000,000,000,000

Token ticker: SRV

Decimals: 9

Compiler Version: v0.8.4+commit.c7e474f2

Contract Deployer Address: 0xEb5F241dAb22A96aD8c84fA328D2b006a0B4e2f6

Optimization Enabled: Yes with 200 runs

Client contacts: SurvivalCoin team

Blockchain: Binance Smart Chain

Project website: <https://survivalcoin.io/>

The audit items and results:

(Other unknown security vulnerabilities are not included in the audit responsibility scope)

Audit Result: Passed

Audit Date: December 17, 2021

Audit Team: AUDIT RATE TECH

<https://www.auditrate.tech>

## ***Introduction***

This Audit Report mainly focuses on the overall security of SurvivalCoin Smart Contract. With this report, we have tried to ensure the reliability and correctness of their smart contract by complete and rigorous assessment of their system's architecture and the smart contract codebase.

### ***Auditing Approach and Methodologies applied***

The AUDIT RATE TECH team has performed rigorous testing of the project starting with analyzing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third-party smart contracts and libraries.

Our team then performed a formal line by line inspection of the Smart Contract to find any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

In the Unit testing Phase, we coded/conducted custom unit tests written for each function in the contract to verify that each function works as expected.

In Automated Testing, we tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was tested in collaboration of our multiple team members and this included -

- Testing the functionality of the Smart Contract to determine proper logic has been followed throughout the whole process.
- Analyzing the complexity of the code in depth and detailed, manual review of the code, lineby-line.
- Deploying the code on testnet using multiple clients to run live tests.
- Analyzing failure preparations to check how the Smart Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analyzing the security of the on-chain data.

## ***Audit Goals***

The focus of the audit was to verify that the Smart Contract System is secure, resilient and working according to the specifications. The audit activities can be grouped in the following three categories:

**Security**

Identifying security related issues within each contract and the system of contract.

**Sound Architecture**

Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.

**Code Correctness and Quality**

A full review of the contract source code. The primary areas of focus include:

- Accuracy
- Readability
- Sections of code with high complexity
- Quantity and quality of test coverage

## ***Issue Categories***

Every issue in this report was assigned a severity level from the following:

### ***High level severity issues***

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

### ***Medium level severity issues***

Issues on this level could potentially bring problems and should eventually be fixed.

### ***Low level severity issues***

Issues on this level are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

## Number of issues per severity

Critical	High	Medium	Low	Note
0	0	0	1	0

## Issues Checking Status

No	Issue description.	Checking status
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Low
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

### **Manual Audit:**

For this section the code was tested/read line by line by our developers. We also used Remix IDE's JavaScript VM and Kovan networks to test the contract functionality.

### **Automated Audit**

Remix Compiler Warnings

It throws warnings by Solidity's compiler. If it encounters any errors the contract cannot be compiled and deployed. No issues found.

### **Owner privileges**

- Transfers ownership of the contract to a new account (`newOwner`).
- Owner entitled to exclude from reward
- Owner entitled to include in reward
- Owner can renounce ownership
- Owner can lock
- Owner entitled to change tax fee
- Owner can exclude from fees
- Owner can include in Fees
- Owner set max tx amount
- Owner set marketing divisor
- Owner set num tokens sell to add to liquidity
- Owner set buyback upper limit
- Owner set marketing address
- Owner set swap and liquify enabled
- Owner set buy back enabled

### **Conclusion**

Smart contract contain 1 low severity issues!

**!** Out of gas

Issue:

includeInReward

The function includeInReward() also uses the loop for evaluating total supply. It also could be aborted with OUT\_OF\_GAS exception if there will be a long excluded addresses list.

Recommendation:

Check that the excluded array lenght is not too big.

### **Note:**

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.

### ***Implemented events***

Approval(address,address,uint256)

Transfer(address,address,uint256)

### ***Implemented functions***

marketingAddress()

transferOwnership(address)

lock(uint256)

allowance(address,address)

unlock()

transfer(address,uint256)

owner()

symbol()

decreaseAllowance(address,uint256)

balanceOf(address)

renounceOwnership()

increaseAllowance(address,uint256)

getTime()

getUnlockTime()

transferFrom(address,address,uint256)

decimals()

name()

approve(address,uint256)

totalFees()

totalSupply()

WETH()

## ***Website Audit***

Address	<a href="https://survivalcoin.io/">https://survivalcoin.io/</a>
Domain registration	1 years
Domain	Clean
Web server	Cloudflare
The server is located	United States
Server response time	0.32 sec
SSL certificate	Yes
JavaScript errors	Not found
Typos, or grammatical errors	Not found
Issues with loading elements, code, or stylesheets	Not found
Malware	Not found
Injected spam	Not found
Internal server errors	Not found
Popups	Not found
Blocking files	Not found
Mobile Friendly	Yes
Compress CSS files	Optimized
Compress JS files	Optimized
Image compression	Optimized
Visible content	Optimized
Social Media/contacts	Yes
Roadmap	Yes



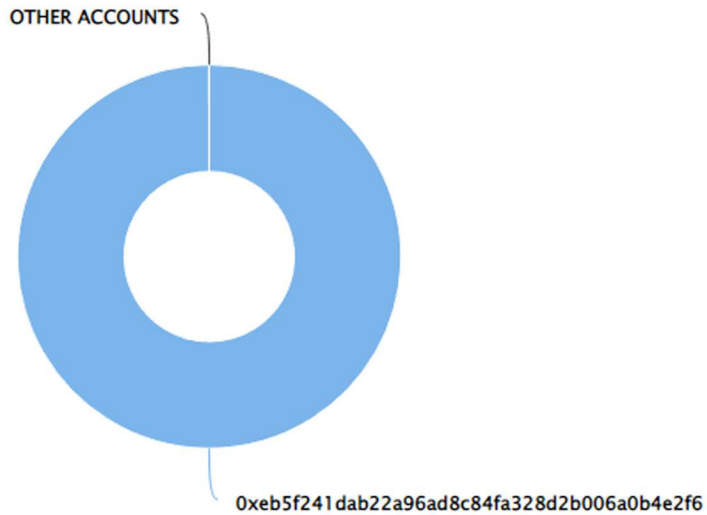
## Top Token Holders

💡 The top 100 holders collectively own 100.00% (100,000,000,000,000.00 Tokens) of SurvivalCoin

💡 Token Total Supply: 100,000,000,000,000.00 Token | Total Token Holders: 1

### SurvivalCoin Top 100 Token Holders

Source: BscScan.com



(A total of 100,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000.00 token)

At the time of the audit:

100% tokens are in the wallet 0xeb5f241dab22a96ad8c84fa328d2b006a0b4e2f6

Do your own research and ask the current owner about it

### ***KYC/Doxx***

At the time of the audit, there is no information about the conduct of KYC / Doxx

## ***THANK YOU!***