AUDIT.SC

Smart Contract Audit

BRING TRUST IN YOUR PROJECT

AUDIT-SC
PARTNER
FlokiEnergy

WWW.AUDIT.SC

2021

FULL SMART CONTRACT AUDIT
SOLIDITY CHECK

Audit SC Guarantees that every smart contract that has been audited has gone through both automated Smart Contract Scanner Softwares and is manually verified by one of our highly experienced smart contract experts.

Table of Contents

AUDIT-SC

DISCLAIMER

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and AUDIT-SC and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (AUDIT-SC) owe no duty of care towards you or any other person, nor does AUDIT-SC make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided «as is», without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and AUDIT-SC hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, AUDIT-SC hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against AUDIT-SC, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

OVERVIEW

PROJECT SUMMARY

Project Name	**Floki Energy (FENG)**
Platform	Binance Smartchain
Language	Solidity

AUDIT SUMMARY

Date	03-12-2021
Audit Type	Static Analysis, Manual Review
Audit Result	**PENDING**

RISK SUMMARY

Risk Level	Total	Found	Pending	Solved	Acknowledgde	Objected
Critical	3	3	3	0	0	0
Major	1	1	1	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informative	12	12	12	0	0	0
Discussion	0	0	0	0	0	0

FINDINGS

Function Default Visibility

SWC-ID: SWC-100

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:
Functions that do not have a function visibility type specified are public by default. This can lead to a vulnerability if a developer forgot to set the visibility and a malicious user is able to make unauthorized or unintended state changes or unnecessary gas usage.

Relevance:
public functions that are never called by the contract should be declared external to save gas.

Category	Risk Level	Number of Findings	Status
SWC-100	Informative	**5**	Pending

Constable State

SWC-ID: SWC-108

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:
Functions that do not have a function visibility type specified are public by default. This can lead to a vulnerability if a developer forgot to set the visibility and a malicious user is able to make unauthorized or unintended state changes or unnecessary gas usage.

Relevance:
public functions that are never called by the contract should be declared external to save gas.

Category	Risk Level	Number of Findings	Status
SWC-108	Informative	**7**	Pending

AUDIT.SC

Integer Overflow

SWC-ID: SWC-101

Relationship:
CWE-682: Incorrect Calculation

Description:
An overflow/underflow happens when an arithmetic operation reaches the maximum or minimum size of a type. For instance if a number is stored in the uint8 type, it means that the number is stored in a 8 bits unsigned number ranging from 0 to 2^8-1. In computer programming, an integer overflow occurs when an arithmetic operation attempts to create a numeric value that is outside of the range that can be represented with a given number of bits – either larger than the maximum or lower than the minimum representable value. In the case of FlokiEnergy, the setMaxWalletPercent() function is realistically vulnerable to unexpected behavior as a result of an integer underflow or overflow.

Category	Risk Level	Number of Findings	Status
SWC-101	Major	**1**	Pending

Unauthorized Access

Relationship:
CWE-284: Improper Access Control

The constructor sets the owner as Authorized address with authorizations[_owner] = true; Upon transferring ownership, the owner address is changed, but the authorization of the previous owner is not lifted, leaving unexpected access control to the previous owner.

Remediation:
Make sure that all access control is properly managed

Category	Risk Level	Number of Findings	Status
Access Control	Critical	**1**	Pending

Reetrancy

SWC-ID: SWC-107

Relationship:
CWE-841: Improper Enforcement of behavioral Workflow

Description:
One of the major dangers of calling external contracts is that they can take over the control flow. In the reentrancy attack (a.k.a. recursive call attack), a malicious contract calls back into the calling contract before the first invocation of the function is finished. This may cause the different invocations of the function to interact in undesirable ways.

Relevance:

In the transferFrom() function, state variables are written before function calls. When interacting with an address that holds arbitrary code, the function's state writing may be called multiple times before the function body ends, resulting in lost funds.

Remediation:

Make sure all internal state changes are performed before the call is executed. This is known as the Checks-Effects-Interactions pattern. You can also use a reetrancy lock/ guard.

Category	Risk Level	Number of Findings	Status
SWC-107	Critical	**1**	Pending

Unchecked Ownership Transfer

Relationship:
CWE-284: Improper Access Control

The transferOwnership() function send the ownership to an arbitrary address without checking if the receiving address is able to receive it. This may result in permanent loss of the ownership of the token contract with no way of retrieving it.

Remediation:
Make sure that the receiving address accepts the ownership before ownership is transferred.

Category	Risk Level	Number of Findings	Status
Access Control	Critical	**1**	Pending

AUDIT DETAILS

SCW-100 Function Default Visibility

authorize(address) should be declared external

unauthorize(address) should be declared external:

transferOwnership(address) should be declared external:

tradingStatus() should be declared external:

cooldownEnabled(bool) should be declared external

SCW-108 Constable State

DividendDistributor.WBNB should be constant

DividendDistributor.dividendsPerShareAccuracyFactor should be constant

FlokiEnergy.DEAD should be constant

FlokiEnergy.ETH should be constant

FlokiEnergy.WBNB should be constant

FlokiEnergy.ZERO should be constant

FlokiEnergy.launchedAt should be constant

AUDIT RESULT

Basic Coding Bugs

1. Constructor Mismatch

o Description: Whether the contract name and its constructor are not identical to each other.

o Result: PASSED

o Severity: Critical

Ownership Takeover

o Description: Whether the set owner function is not protected.

*o Result: **FAILED***

o Severity: Critical

Redundant Fallback Function

o Description: Whether the contract has a redundant fallback function.

o Result: PASSED

o Severity: Critical

Overflows & Underflows

Description: Whether the contract has general overflow or underflow Vulnerabilities

*o Result: **FAILED***

o Severity: Critical

Reentrancy

o Description: Reentrancy is an issue when code can call back into your contract and change state, such as withdrawing ETHs.

*o Result: **FAILED***

o Severity: Critical

MONEY-Giving Bug

o Description: Whether the contract returns funds to an arbitrary address.

o Result: PASSED

o Severity: High

Blackhole

- o Description: Whether the contract locks ETH indefinitely: merely in without out.
- o Result: PASSED
- o Severity: High

Unauthorized Self-Destruct

- o Description: Whether the contract can be killed by any arbitrary address.
- o Result: PASSED
- o Severity: Medium

Revert DoS

- o Description: Whether the contractis vulnerable to DoSattack because of unexpected revert.
- o Result: PASSED
- o Severity: Medium

Unchecked External Call

- o Description: Whether the contract has any external call without checking the return value.
- o Result: PASSED
- o Severity: Medium

Gasless Send

- o Description: Whether the contractis vulnerable to gasless send.
- o Result: PASSED
- o Severity: Medium

Send Instead of Transfer

- o Description: Whether the contract uses send i
- o Result: PASSED
- o Severity: Medium

Costly Loop

- o Description: Whether the contract has any costly loop which may lead to Out-Of-Gas exception.
- o Result: PASSED
- o Severity: Medium

(Unsafe) Use of Untrusted Libraries

- o Description: Whether the contract use any suspicious libraries.
- o Result: PASSED
- o Severity: Medium

(Unsafe) Use of Predictable Variables

- o Description: Whether the contract contains any randomness variable, but its value can be predicated.
- o Result: PASSED
- o Severity: Medium

Transaction Ordering Dependence

- o Description: Whether the final state of the contract depends on the order of the transactions.
- o Result: PASSED
- o Severity: Medium

. Deprecated Uses

- o Description: Whether the contract use the deprecated tx.origin to perform the authorization.
- o Result: PASSED
- o Severity: Medium

AUDIT.SC

CONTACT US

Audit SC Guarantees that every smart contract that has been audited has gone through both automated Smart Contract Scanner Softwares and is manually verified by

E-mail: info@audit.sc

Website: www.audit.sc

AUDIT.SC