



Smart Contract Audit

FOR

Neira

DATED : 24 Sep 24'



AUDIT SUMMARY

Project name – Neira

Date: 24 Sep, 2024

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	1	2	1
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:

A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :

The code has undergone static analysis using Slither.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/address/0x0bbda504ac971418be0e5dc41ffa72db7225ec0a#code>



Token Information

Token Address:

0x49c294f4aa65C512ba17739cA1085F044C83ac98

Name: Neira

Symbol: Neira

Decimals: 18

Network: BscScan

Token Type: BEP-20

Owner: 0x2A442BdEB27C5e6E901fE8F6380b918F1Ce7A57d

Deployer:

0x2A442BdEB27C5e6E901fE8F6380b918F1Ce7A57d

Token Supply: 420690000000

Checksum: 8401cc1e8d877bef5854b883b04898d4

Testnet version:

<https://testnet.bscscan.com/address/0x0bbda504ac971418be0e5dc41ffa72db7225ec0a#code>



TOKEN OVERVIEW

Buy Fee: 5%

Sell Fee: 5%

Transfer Fee: 0-0%

Fee Privilege: No

Ownership: Owned

Minting: No

Max Tx: No

Blacklist: No

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
 - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
 - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
 - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
 - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-

VULNERABILITY CHECKLIST

- | | |
|------------------------------------|-------------------------------|
| ✓ Return values of low-level calls | ✓ Gasless Send |
| ✓ Private modifier | ✓ Using block.timestamp |
| ✓ Multiple Sends | ✓ Re-entrancy |
| ✓ Using Suicide | ✓ Tautology or contradiction |
| ✓ Gas Limitand Loops | ✓ Timestamp Dependence |
| ✓ Address hardcoded | ✓ Revert/require functions |
| ✓ Exception Disorder | ✓ Use of tx.origin |
| ✓ Using inline assembly | ✓ Integer overflow/underflow |
| ✓ Divide before multiply | ✓ Dangerous strict equalities |
| ✓ Missing Zero Address Validation | ✓ Using SHA3 |
| ✓ Compiler version not fixed | ✓ Using throw |
-

INHERITANCE TREE





POINTS TO NOTE

- The owner can renounce and transfer ownership.
- The owner can clear the stuck balance.
- The owner can set the SwapBack Settings.
- The owner can set the IsFeeExempt.



FUNCTIONAL TESTING

1- Approve (passed):

<https://testnet.bscscan.com/tx/0xfe47cc750bd93f4eaacf900306b3300d46acf0330a2180e95cf9c4f5cae60626>

2- Clear Stuck Balance (passed):

<https://testnet.bscscan.com/tx/0x357f7753e5cf7f17d2d72a9b65d1acd117a5d43a3f4df8c42277190cfcc6625b>

3- Approve Max (passed):

<https://testnet.bscscan.com/tx/0xef5a6e7f2db08f5f3f6ab43606fd784ea73b4eae4b5d3fcb8bfa3151c3242599>

4- Transfer (passed):

<https://testnet.bscscan.com/tx/0xbdcceb0a378de84c85bdaf396973dc78b0034d8e238b6d6cf02fed713408c516>



CLASSIFICATION OF RISK

Severity

Description

◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity

Found

◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	1
◆ Low-Risk	2
◆ Gas Optimization / Suggestions	1

MANUAL TESTING

Centralization – Liquidity is added to EOA.

Severity: Medium

function: addLiquidityETH

Status: Open

Overview:

Liquidity is added to EOA. The autoLiquidityReceiver may drain it.

```
if(amountToLiquify > 0) {  
    router.addLiquidityETH{value: amountBNBLiquidity}(  
        address(this),  
        amountToLiquify,  
        0,  
        0,  
        autoLiquidityReceiver,  
        block.timestamp  
    );  
}
```

Suggestion:

It is suggested that the address should not be zero or dead.

MANUAL TESTING

Centralization – Missing Events

Severity: Low

Subject: Missing Events

Status: Open

Overview:

They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setSwapBackSettings(bool _enabled, uint256 _amountS,  
uint256 _amountL, bool _alternate) external onlyOwner {  
    alternateSwaps = _alternate;  
    claimingFees = _enabled;  
    smallSwapThreshold = _amountS;  
    largeSwapThreshold = _amountL;  
    swapThreshold = smallSwapThreshold;  
}
```

```
function setIsFeeExempt(address holder, bool exempt) external  
onlyOwner {  
    isFeeExempt[holder] = exempt;  
}
```

Suggestion:

Emit an event for critical changes.



MANUAL TESTING

Centralization – Remove the safe math library.

Severity: **Low**

Status: **Open**

Line Number: **84-121**

Overview:

The Safe Math library is no longer needed for Solidity version 0.8 and above. This is because Solidity 0.8 includes checked arithmetic operations by default. All of Safe Math's methods are now inherited into Solidity programming.



MANUAL TESTING

Optimization

Severity: Optimization

Subject: Remove unused code.

Status: Open

Overview:

Unused variables are allowed in Solidity, and they do not pose a direct security issue. It is the best practice, though, to avoid them.

```
function _msgData() internal view returns (bytes memory) {  
    this; // silence state mutability warning without generating bytecode - see  
    https://github.com/ethereum/solidity/issues/2691  
    return msg.data;  
}
```

```
function checkTxLimit(address sender, uint256 amount) internal view {  
    require(amount <= _maxTxAmount || isTxLimitExempt[sender], "TX Limit  
Exceeded");  
}
```

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



ABOUT AUDITACE

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
