Assure DeFi®

THE VERIFICATION GOLD STANDARD

Security Assessment

ICB Network

Date: 28/04/2024

Audit Status: PASS

Audit Edition: Advanced

ASSURE DEFI®

THE VERIFICATION GOLD STANDARD

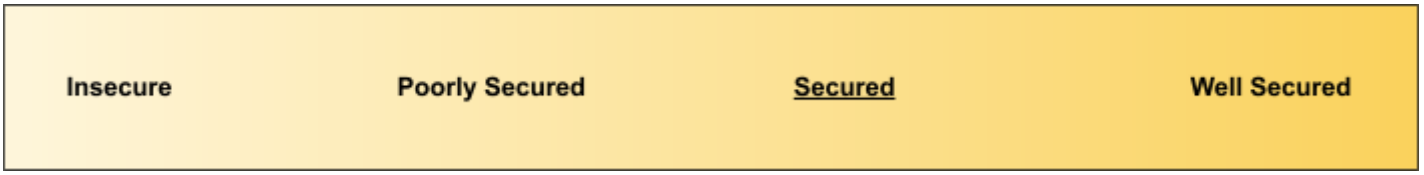
Risk Analysis

Vulnerability summary

Classification	Description
High	High-level vulnerabilities can result in the loss of assets or manipulation of data.
Medium	Medium-level vulnerabilities can be challenging to exploit, but they still have a considerable impact on smart contract execution, such as allowing public access to critical functions.
Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.
Informational	Informational vulnerabilities, code style violations, and informational statements do not affect smart contract execution and can typically be disregarded.

Executive Summary

According to the Assure assessment, the Customer's smart contract is **Secured.**



Scope

Target Code And Revision

For this audit, we performed research, investigation, and review of the ICB Network contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	ICBStaking.sol - [SHA256] *98683389c205ebbf8d4fc43537b36537f566424a445279f4baf0ed534562c004*
Audit Methodology	Static, Manual

Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

Category	Item
Code review & Functional Review	• Compiler warnings. • Race conditions and Reentrancy. Cross-function race conditions. • Possible delays in data delivery. • Oracle calls. • Front running. • Timestamp dependence. • Integer Overflow and Underflow. • DoS with Revert. • DoS with block gas limit. • Methods execution permissions. • Economy model. • Private user data leaks. • Malicious Event log. • Scoping and Declarations. • Uninitialized storage pointers. • Arithmetic accuracy. • Design Logic. • Cross-function race conditions. • Safe Zeppelin module. • Fallback function security. • Overpowered functions / Owner privileges

AUDIT OVERVIEW

HIGH

No high severity issues were found.

MEDIUM

No medium severity issues were found.

LOW

1. Ensuring Correct Timing in Smart Contract Pools

Contract: ICBStaking

Function: createPool()

Issue: The contract may behave unpredictably if _startTime for the pool is set incorrectly.

Mitigation: Implement a require() statement to ensure _startTime is greater than block.timestamp, preventing the creation of a pool with a past start time.

INFORMATIONAL

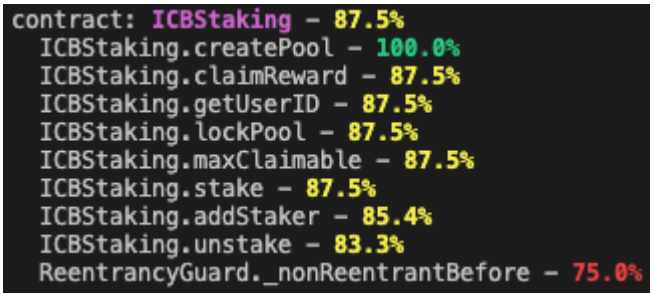
No informational severity issues were found.

Testing coverage

During the testing phase, custom use cases were written to cover all the logic of contracts. *Check “Annexes” to see the testing code.*

ICB Network staking test:

Coverages:



Testing ICB Network Staking contract:

```
tests/test_icb_staking.py::test_create_pool RUNNING
Transaction sent: 0xe5f4049c8d1205fe9cf90125b7a43c2142acf14de02b79d1b844c6db81bb3968
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ICBStaking.constructor confirmed   Block: 1   Gas used: 1997576 (16.65%)
  ICBStaking deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x9dad7e5efb8d5886b3660202363239abf9055cb3df4d3ablad2f4aab71a5dfdb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ICBStaking.createPool confirmed (You are not the owner)   Block: 2   Gas used: 22962 (0.19%)

Transaction sent: 0x2a72f395e4fea7e0c9ce8fd072cdee9b68830188b652a043c6265a5feac75786
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ICBStaking.createPool confirmed (Reward cant be 0)   Block: 3   Gas used: 23063 (0.19%)

Transaction sent: 0x6d7929db68b48b4e7189dbd221f82d29700c4d65b6938623b64a53bfde736126
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  ICBStaking.createPool confirmed (Reward cant be 0)   Block: 4   Gas used: 23097 (0.19%)

Transaction sent: 0x96da1d0645fe0e46e8fe1277d90b60ba1d003d282d74ea0aa6e8ed13f117d40f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  ICBStaking.createPool confirmed (Start time cant be less than end time)   Block: 5   Gas used: 23138 (0.19%)

Transaction sent: 0x5032a1d3d4f406c3a733908b4ef0d1785df2050c7d02efe6c02e7aaeef8dc8ef
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ICBStaking.createPool confirmed   Block: 6   Gas used: 130457 (1.09%)

tests/test_icb_staking.py::test_create_pool PASSED
tests/test_icb_staking.py::test_lock_pool RUNNING
Transaction sent: 0xce02271acadfbf330ac5c5841e64114afdb06523f7553fc4e259e4c243f8fea7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  ICBStaking.constructor confirmed   Block: 7   Gas used: 1997576 (16.65%)
  ICBStaking deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0xac25b8a9f807e6deed8302f3891f72041395d32ce2b521e968856cb0faadbeec
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  ICBStaking.createPool confirmed   Block: 8   Gas used: 130457 (1.09%)

Transaction sent: 0x4bf249b398d96254962198738985a5ee7f1d6ae010843890e49ed29615d914f3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ICBStaking.lockPool confirmed (You are not the owner)   Block: 9   Gas used: 34271 (0.29%)

Transaction sent: 0x712ccc903faf43634767e9fc56943858b47a5de68adb9b14404bc03e4840c2a0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  ICBStaking.lockPool confirmed (Pool didnt started yet cant be locked before start time)   Block: 10   Gas used: 36068 (0.30%)

Transaction sent: 0x0cb4219d2c0c4cac8516db027e094cd9fa5febccf20aa1f8167d999b89777966
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  ICBStaking.lockPool confirmed   Block: 12   Gas used: 64451 (0.54%)

Transaction sent: 0xcb76a34e4935b0866bc2ce841ff703c33369ee4b5ff715e3781be05af96d9d07
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  ICBStaking.lockPool confirmed (Pool is not open)   Block: 13   Gas used: 27534 (0.23%)

tests/test_icb_staking.py::test_lock_pool PASSED
```

```
tests/test_icb_staking.py::test_stake RUNNING
Transaction sent: 0x1c5a882fe4f62a41e7d6b4406a8a43f6dd15142a894904c205bec7f55f8a2018
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  ICBStaking.constructor confirmed   Block: 14   Gas used: 1997576 (16.65%)
  ICBStaking deployed at: 0xb6286fAFd0451320ad6A8143089b216C2152c025

Transaction sent: 0x46fb62c612c2e654abf8e5e485d8d8523a36cfe1421a271bfb478788c26aadf2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  ICBStaking.createPool confirmed   Block: 15   Gas used: 130457 (1.09%)

Transaction sent: 0x5bd00df2e10e6e057c7f7d6212587c8384874d389a84180c4191e48080dcbcd0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  ICBStaking.stake confirmed (Staking not open)   Block: 16   Gas used: 39483 (0.33%)

Transaction sent: 0x05dbabefd439533e8d89878a5889f423ee57405478bf1d68009d474bfa6d5c83
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  ICBStaking.stake confirmed (Staking not open)   Block: 17   Gas used: 39449 (0.33%)

Transaction sent: 0x028bdc02b9c1557e3c68b06b31be9c4e9793b6249d7353bae8fd311ea4c68a91
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ICBStaking.stake confirmed (0 ICB cant be staked)   Block: 19   Gas used: 39493 (0.33%)

Transaction sent: 0x449edfec0078f05d67b24b57f73b7952189436665267d92c26bd71b23113d623
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  ICBStaking.stake confirmed   Block: 20   Gas used: 189985 (1.58%)

Transaction sent: 0x793fa40c8bbe66b7b3eeab9623f0877bcab91586d0b0b7ba6daf3274c989de7b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  ICBStaking.stake confirmed   Block: 21   Gas used: 80035 (0.67%)

Transaction sent: 0x325bfac594c917acd749aeb0f157a58aa6175405cbd77e3c7886504ad25c0d51
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ICBStaking.stake confirmed   Block: 22   Gas used: 180796 (1.51%)

tests/test_icb_staking.py::test_stake PASSED
```

```
tests/test_icb_staking.py::test_unstake RUNNING
Transaction sent: 0x0f66d911a099bd2ee16440108517bd9d45790f2b1eeca239f47049e91f51b310
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  ICBStaking.constructor confirmed   Block: 23   Gas used: 1997576 (16.65%)
  ICBStaking deployed at: 0x2c15A315610Bfa5248E4CbCbd693320e9D8E03Cc

Transaction sent: 0x0225cde2ee4702ece4cdc0eafb00c09f98d9744aedc29b7eaf5c93c51e9d7c0d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  ICBStaking.createPool confirmed   Block: 24   Gas used: 130457 (1.09%)

Transaction sent: 0x17b6d6299946a5f4f5a9f8121ac04abdf92349d5fc5992791338ec29382a3f18
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  ICBStaking.unstake confirmed (No Staking for this user in this pool)   Block: 25   Gas used: 22555 (0.19%)

Transaction sent: 0x35d90256aa069bf0d622fbfca8a361109c24930b73762c32a4acb6e49bb5c482
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  ICBStaking.stake confirmed   Block: 27   Gas used: 189985 (1.58%)

Transaction sent: 0x174e29dafb7e1f179593208d80deaf2030e9b821e730e9d227b17240745a425d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  ICBStaking.unstake confirmed (Pool is not locked)   Block: 28   Gas used: 31285 (0.26%)

Transaction sent: 0x097cb66d187a1290b64818501387fd51ecd1d477f898566365eb1f372a300287
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  ICBStaking.lockPool confirmed   Block: 29   Gas used: 64451 (0.54%)

Transaction sent: 0xb7bd2709fdd2401567a76474cd437201277edd1e04879cfadcd057bf50f877af
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  ICBStaking.unstake confirmed (Staking period not over)   Block: 30   Gas used: 56691 (0.47%)

Transaction sent: 0x91f41dbca2167591835dd412535477e052ce204a053b27ec337c4e0656222492
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  ICBStaking.unstake confirmed   Block: 32   Gas used: 123692 (1.03%)

tests/test_icb_staking.py::test_unstake PASSED
```

```
tests/test_icb_staking.py::test_claim_reward RUNNING
Transaction sent: 0x9440e1a7b5320998bc3cf9e71255305151db784a62127e73a49c7801c47c27be
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  ICBStaking.constructor confirmed   Block: 33   Gas used: 1997576 (16.65%)
  ICBStaking deployed at: 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d

Transaction sent: 0x1879f780ba24bff28055e67dbde89ac2b10d7b660d362367fa4b67e1d0260ad4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  ICBStaking.createPool confirmed   Block: 34   Gas used: 130457 (1.09%)

Transaction sent: 0xce843247afa9e4baeed95ee84d5882f0ca3be7cf3cc1fc615f5246f5226c26ed
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  ICBStaking.claimReward confirmed (No Staking for this user in this pool)   Block: 35   Gas used: 28385 (0.24%)

Transaction sent: 0xadfae97cb9df09bcf9d6a8efceb4e363ea2a5c6657db3b2e17f075d0e4a5fea2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  ICBStaking.stake confirmed   Block: 37   Gas used: 189985 (1.58%)

Transaction sent: 0x424bbc035ef593f2608a19be643ba2b4254e06e7d4c4320e8100993f75ea6de7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ICBStaking.stake confirmed   Block: 38   Gas used: 180796 (1.51%)

Transaction sent: 0xec8014d84dbfe817fe82b02e04b123a1f7a4b9a03ac43aa11f56a4180379e6db
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ICBStaking.stake confirmed   Block: 39   Gas used: 201608 (1.68%)

Transaction sent: 0x6b993dea75c0d97b32a9ed9095a7f5d6165885dc185caee8e1659ffe76aa52f2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ICBStaking.stake confirmed   Block: 40   Gas used: 222419 (1.85%)

Transaction sent: 0x671020e95e9ef641cc14888f4180a041fa38beb0d90f45428abe444951911b6d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ICBStaking.stake confirmed   Block: 41   Gas used: 243230 (2.03%)

Transaction sent: 0xd7ef49ede04fe76de4455b6660192757ff38fabdde248dcbe0b9e809e7486250
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  ICBStaking.claimReward confirmed (Pool is not locked)   Block: 42   Gas used: 98081 (0.82%)

Transaction sent: 0xcb85dcf697687a09c183af998662da2e909ad59c527a377493b8e75d3201d02e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  ICBStaking.lockPool confirmed   Block: 43   Gas used: 64451 (0.54%)

Transaction sent: 0x4b78a016dc6bed338a6bd26c1c2237723e2d030c6c70fccd2bc83e3e1f7d6afc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  ICBStaking.claimReward confirmed (No reward available to claim)   Block: 44   Gas used: 140270 (1.17%)

Transaction sent: 0xd803b0bdafe49bdb433397450f486d4069231272a759ef514f59da3138eac603
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ICBStaking.claimReward confirmed   Block: 46   Gas used: 204914 (1.71%)

Transaction sent: 0x691266970aa1649968729fde7aa369e4010aefc932e472c0b03f38fec163f084
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  ICBStaking.unstake confirmed   Block: 47   Gas used: 189524 (1.58%)

tests/test_icb_staking.py::test_claim_reward PASSED
```

Annexes

Testing code:

`icb_staking.py:`

```
from brownie import (
    reverts
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    DAY_TIMESTAMP,
    get_account,
    increase_timestamp,
    get_timestamp,
)

from scripts.deploy import (
    deploy_staking
)

def test_create_pool(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy contracts
    staking = deploy_staking(owner)
    with reverts("You are not the owner"):
        staking.createPool(0, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
other})
    with reverts("Reward cant be 0"):
        staking.createPool(0, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
owner})
    with reverts("Reward cant be 0"):
        staking.createPool(10, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
owner})
    with reverts("Start time cant be less than end time"):
        staking.createPool(10, get_timestamp(5), get_timestamp(1), DAY_TIMESTAMP, {"from":
owner, "value": 10e18})

    tx = staking.createPool(10, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
```

```
owner, "value": 10e18})
    assert tx.events['PoolCreated'][0]['_poolid'] == 0
    assert tx.events['PoolCreated'][0]['_totalReward'] == 10e18

def test_lock_pool(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)

    # Deploy contracts
    staking = deploy_staking(owner)
    staking.createPool(10, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
owner, "value": 10e18})
    with reverts("You are not the owner"):
        staking.lockPool(1, {"from": other})
    with reverts("Pool didnt started yet cant be locked before start time"):
        staking.lockPool(0, {"from": owner})
    increase_timestamp(2 * DAY_TIMESTAMP)
    staking.lockPool(0, {"from": owner})
    with reverts("Pool is not open"):
        staking.lockPool(0, {"from": owner})

def test_stake(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy contracts
    staking = deploy_staking(owner)
    staking.createPool(10, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
owner, "value": 10e18})
    with reverts("Staking not open"):
        staking.stake(1, {"from": other})
    with reverts("Staking not open"):
        staking.stake(0, {"from": other})
    increase_timestamp(2 * DAY_TIMESTAMP)
    with reverts("0 ICB cant be staked"):
        staking.stake(0, {"from": other})
    tx = staking.stake(0, {"from": other, "value": 1e18})
    assert tx.events['Staked'][0]['user'] == other
    assert tx.events['Staked'][0]['amount'] == 1e18
    tx = staking.stake(0, {"from": other, "value": 1e18})
    assert tx.events['Staked'][0]['user'] == other
    assert tx.events['Staked'][0]['amount'] == 1e18
```

```
    tx = staking.stake(0, {"from": extra, "value": 1e18})
    assert tx.events['Staked'][0]['user'] == extra
    assert tx.events['Staked'][0]['amount'] == 1e18

def test_unstake(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)

    # Deploy contracts
    staking = deploy_staking(owner)
    staking.createPool(10, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
owner, "value": 10e18})
    with reverts("No Staking for this user in this pool"):
        staking.unstake(0, {"from": other})
    increase_timestamp(2 * DAY_TIMESTAMP)
    staking.stake(0, {"from": other, "value": 1e18})
    with reverts("Pool is not locked"):
        staking.unstake(0, {"from": other})
    staking.lockPool(0, {"from": owner})
    with reverts("Staking period not over"):
        staking.unstake(0, {"from": other})
    increase_timestamp(6 * DAY_TIMESTAMP)
    tx = staking.unstake(0, {"from": other})
    assert tx.events['UnStake'][0]['user'] == other
    assert tx.events['UnStake'][0]['amount'] == 1e18

def test_claim_reward(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    staker_blue = get_account(3)
    staker_red = get_account(4)
    staker_brown = get_account(5)

    # Deploy contracts
    staking = deploy_staking(owner)
    staking.createPool(10, get_timestamp(1), get_timestamp(5), DAY_TIMESTAMP, {"from":
owner, "value": 10e18})
    with reverts("No Staking for this user in this pool"):
        staking.claimReward(0, {"from": other})
    increase_timestamp(2 * DAY_TIMESTAMP)
    # Differents stakers
```

```
    staking.stake(0, {"from": other, "value": 1e18})
    staking.stake(0, {"from": extra, "value": 4e18})
    staking.stake(0, {"from": staker_blue, "value": 2e18})
    staking.stake(0, {"from": staker_red, "value": 3e18})
    staking.stake(0, {"from": staker_brown, "value": 6e18})
    with reverts("Pool is not locked"):
        staking.claimReward(0, {"from": other})
    staking.lockPool(0, {"from": owner})
    with reverts("No reward available to claim"):
        staking.claimReward(0, {"from": other})

    increase_timestamp(30 * DAY_TIMESTAMP)
    tx = staking.claimReward(0, {"from": staker_brown})
    assert tx.events['Withdrawn'][0]['user'] == staker_brown
    assert tx.events['Withdrawn'][0]['amount'] == 6e18
    assert tx.events['Withdrawn'][0]['reward'] == 3.75e18
    assert tx.events['Withdrawn'][0]['poolId'] == 0

    tx = staking.unstake(0, {"from": other})
    assert tx.events['UnStake'][0]['user'] == other
    assert tx.events['UnStake'][0]['amount'] == 1e18
    assert tx.events['UnStake'][0]['poolId'] == 0
```

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
High	0					
Medium	0					
Low	1					
Informational	0					

Assessment Results

Score Results

Review	Score
Global Score	**90/100**
Assure KYC	https://assuredefi.com/projects/icb-network/
Audit Score	85/100

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below. The Global Score is a combination of the evaluations obtained between having or not having KYC and the type of contract audited together with its manual audit.

Audit PASS

Following our comprehensive security audit of the staking contract for ICB Network project, the audit has been successfully completed and passed. However, we recommend reviewing and addressing the low vulnerability reported to ensure the robustness and security of the smart contract.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocating for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any person reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audit completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.