Assure DeFi®

THE VERIFICATION **GOLD STANDARD**

Security Assessment

Coincreate

Date: 01/11/2024

Audit Status: PASS

Audit Edition: Advanced

ASSURE DEFI®

THE VERIFICATION **GOLD STANDARD**

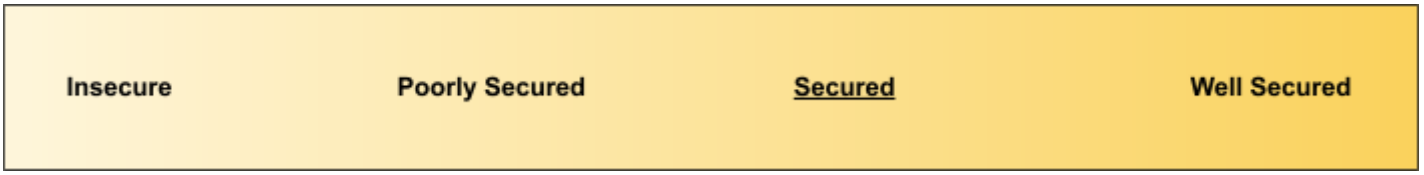
Risk Analysis

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Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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 Coincreate contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	Btoken.sol [SHA256]: b9bf669a6f60a2f418009b39d7356ea50b71ac3257c7891debfbb6f1f24a08df
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

No low severity issues were found.

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.*

Coincreate contract tests:

Test_btoken:

```
tests/test_btoken.py::test_burn RUNNING
Transaction sent: 0x90d13fc1fc76df28fb70464b27d2dbb5cbfebc7339429bfb23815e722afed7ec
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  BToken.constructor confirmed   Block: 1   Gas used: 1554390 (12.95%)
  BToken deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x3f6b404f113699ef1cc831d8b4e48ecb7ce1182f940e07672601c925108bf6f4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  BToken.burn confirmed (Burn amount must be greater than zero)   Block: 2   Gas used: 21601 (0.18%)

Transaction sent: 0xed7f3694d0cb1aa114464fc003ae9400e00ec4a64840133ecc2fd2fa341aeae6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  BToken.burn confirmed (Insufficient balance)   Block: 3   Gas used: 24609 (0.21%)

Transaction sent: 0xb1d7e257d24c31603fb184f489082c9ced612e9c26c9f1f8c1327bb2fc19f7a8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  BToken.transfer confirmed   Block: 4   Gas used: 53120 (0.44%)

Transaction sent: 0x7d34a45139cb45a98e9ff2d9064bdd9f05eb8801e3318016543cb485c9c5f154
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  BToken.burn confirmed   Block: 5   Gas used: 43883 (0.37%)

tests/test_btoken.py::test_burn PASSED
tests/test_btoken.py::test_set_trusted_forwarder RUNNING
Transaction sent: 0xc4c052aeeb07f5f9a5e5aeba2a8e99b5199dbf7d0e45ea3a28d32b17c08e0f01
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  BToken.constructor confirmed   Block: 6   Gas used: 1554390 (12.95%)
  BToken deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0xde81e14a61c3a35e392ced13e16fc528d00e2aa24d0b83e3b6b11d03a5ad0cb6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  BToken.setTrustedForwarder confirmed (reverted)   Block: 7   Gas used: 26804 (0.22%)

Transaction sent: 0x0fbc62aae212d929b130e1d2a60f6a5661a9044e8876b2b7f5d40e948dae2ede
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  BToken.setTrustedForwarder confirmed (Trusted Forwarder address cannot be the zero address)   Block: 8   Gas used: 24632 (0.21%)

Transaction sent: 0xed5589f18b244636fb47ad33b319f2f449afc4ebd7e32a046e49b5fb2d92db52
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  BToken.setTrustedForwarder confirmed   Block: 9   Gas used: 32614 (0.27%)

tests/test_btoken.py::test_set_trusted_forwarder PASSED
tests/test_btoken.py::test_set_token_uri RUNNING
Transaction sent: 0x68d6a220afeb90d87b29a2c90a9de80fdfe10cc98f435bfb0beed35d9ada4b0e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  BToken.constructor confirmed   Block: 10   Gas used: 1554390 (12.95%)
  BToken deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0x434ebb09d87390e3cd3cbdd52d9bb9847bef9be153f1241fa768a7eccc203e09
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  BToken.setTokenURI confirmed (reverted)   Block: 11   Gas used: 27211 (0.23%)

Transaction sent: 0x0a5840e44e8dc58427f7e92478d12bledefaabad2dc99244300c95920c0204fd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  BToken.setTokenURI confirmed   Block: 12   Gas used: 34048 (0.28%)

tests/test_btoken.py::test_set_token_uri PASSED
```

Annexes

Testing code:

Test btoken:

```
from brownie import (
    reverts,
)


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


from scripts.deploy import (
    deploy_btoken
)


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


    btoken = deploy_btoken(owner, extra, 100000e18, owner)
```

```
    with reverts("Burn amount must be greater than zero"):
        btoken.burn(0, {"from": other})
    with reverts("Insufficient balance"):
        btoken.burn(1e18, {"from": other})
    btoken.transfer(other, 5e18, {"from": owner})
    tx = btoken.burn(1e18, {"from": other})
    assert tx.events['TokensBurned'][0]['burner'] == other
    assert tx.events['TokensBurned'][0]['amount'] == 1e18


def test_set_trusted_forwarder(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    new_forwarder = get_account(3)
    diff_owner = get_account(4)


    btoken = deploy_btoken(owner, extra, 100000e18, diff_owner)
    with reverts():
        btoken.setTrustedForwarder(new_forwarder, {"from": other})
    with reverts("Trusted Forwarder address cannot be the zero address"):
        btoken.setTrustedForwarder(ZERO_ADDRESS, {"from": owner})
    tx = btoken.setTrustedForwarder(new_forwarder, {"from": owner})
    assert tx.events['TrustedForwarderUpdated'][0]['newTrustedForwarder'] == new_forwarder


def test_set_token_uri(only_local):
    # Arrange
    owner = get_account(0)
```

```
    other = get_account(1)

    extra = get_account(2)

    old_token_uri = "token_uri"

    new_token_uri = "new_uri"



    btoken = deploy_btoken(owner, extra, 100000e18, owner)

    with reverts():

        btoken.setTokenURI(new_token_uri, {"from": other})

    assert btoken.tokenURI() == old_token_uri

    tx = btoken.setTokenURI(new_token_uri, {"from": owner})

    assert tx.events['TokenURISet'][0]['tokenURI'] == new_token_uri

    assert btoken.tokenURI() == new_token_uri
```

Technical Findings Summary

Findings

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

Assessment Results

Score Results

Review	Score
Global Score	**95/100**
Assure KYC	https://assuredefi.com/projects/coincreate
Audit Score	90/100

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 token contract for the Coincreate project, we inform you that the contract has met the necessary security standards.

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 adCoincreate 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 adCoincreate, 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 serCoincreates 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 serCoincreates, 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 serCoincreates may access, and depend upon, multiple layers of third parties.