

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 Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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	TokenVesting.sol [SHA256] 90798ab1dec175fee19e9801b1a3aa9c705eda8cf7003a84fdc2e2e00062b6eb
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	<ul style="list-style-type: none">• 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



No high severity issues were found.



No medium severity issues were found.



No low severity issues were found.



1. Lack of Clarity in Vesting Schedule Checks in onlyIfVestingScheduleNotRevoked()

Contract: TokenVesting

Function: onlyIfVestingScheduleNotRevoked()

Issue: The function currently lacks a clear error message when the vesting schedule is revoked.

Recommendation: Add a descriptive message to the require() statement for better clarity.

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 Token Vesting:

```
tests/test_token_vesting.py::test_create_vesting RUNNING
Transaction sent: 0x8f8090a2dd268daed3d929b8ea84e8156d714d752230792bbe5b2ebc41e08766
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
ERC20Mock.constructor confirmed Block: 1 Gas used: 523834 (4.37%)
ERC20Mock deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0xb9249e83e88bdfc102c2c9d486e81c83724924108a4f7053f33873ddb92fc28
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
TokenVesting.constructor confirmed Block: 2 Gas used: 1371828 (11.43%)
TokenVesting deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0xe42efd4251123bfa4314a56ad4ee25ab44505d1d1241f0ee8ef78ab058b8716b
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
TokenVesting.createVestingSchedule confirmed (UNAUTHORIZED) Block: 3 Gas used: 24112 (0.20%)

Transaction sent: 0xba9cecd159d04039c976b2d5aaef1fff79aace70884fc7fb5aa546ef5f2e01426
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
TokenVesting.createVestingSchedule confirmed (TokenVesting: cannot create vesting schedule because not sufficient tokens) Block: 4 Gas used: 27301 (0.23%)

Transaction sent: 0x683f8b4187b51b3d9efde09974206f2731b1fe4d31062c7b850243d989ba18aa
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
ERC20Mock.mint confirmed Block: 5 Gas used: 65821 (0.55%)

Transaction sent: 0x11913da07c39e30ee3f3d84a7186c632393a518637004bd0311b78dd9241273
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 4
TokenVesting.createVestingSchedule confirmed (TokenVesting: duration must be > 0) Block: 6 Gas used: 27270 (0.23%)

Transaction sent: 0x0657f19139be8be9308f75466f51bfde866c176a15ed86b4e637ad98963a553a5
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 5
TokenVesting.createVestingSchedule confirmed (TokenVesting: amount must be > 0) Block: 7 Gas used: 27236 (0.23%)

Transaction sent: 0x2dbd21a83c2382fc4e1b38c77c5943b472d69136d33031a5063471169b3336b1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 6
TokenVesting.createVestingSchedule confirmed (TokenVesting: beneficiary cannot be the zero address) Block: 8 Gas used: 27136 (0.23%)

Transaction sent: 0x34d601f90071a0fe3bdf8826f47227b09e1e6e4544d8f299f5d0569ec0252f
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 7
TokenVesting.createVestingSchedule confirmed (TokenVesting: slicePeriodSeconds must be >= 1) Block: 9 Gas used: 27354 (0.23%)

Transaction sent: 0x0455ede61f698c0b43f20d7b935bacbdcla4c2577f9de289fad6555164a5f23a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 8
TokenVesting.createVestingSchedule confirmed (TokenVesting: duration must be >= cliff) Block: 10 Gas used: 27416 (0.23%)

Transaction sent: 0xe51f7d696f37d45be1bac928001a6332765761115ed60e4b41b5a7ed13718e5d
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 9
TokenVesting.createVestingSchedule confirmed Block: 11 Gas used: 318839 (2.66%)

tests/test_token_vesting.py::test_create_vesting PASSED
tests/test_token_vesting.py::test_revoke RUNNING
Transaction sent: 0x3058b93f3c0e9b5358a633bcd911dbe3b309f0b26aab4d8f31d1f9e2b3041ff5
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 10
ERC20Mock.constructor confirmed Block: 12 Gas used: 523834 (4.37%)
ERC20Mock deployed at: 0xb6286fAFd0451320ad6A8143089b216C2152c025

Transaction sent: 0xd9fc8d2291420f690efadf2a030adca594efa424143a8304954cfff1358120d0
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 11
TokenVesting.constructor confirmed Block: 13 Gas used: 1371828 (11.43%)
TokenVesting deployed at: 0x7a3d735ee6873f17Dbdcab1d518604928dc10d92

Transaction sent: 0x48bf577ba89f4a1ffc736426be20dae3e82aebd72b393b03cb7a10fa4c6d4ba8
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 12
ERC20Mock.mint confirmed Block: 14 Gas used: 65821 (0.55%)

Transaction sent: 0xa07097c1ee4d67a5134aeb512f95b462e138a86e3301b55bd016c0b09fbf3496
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 13
TokenVesting.createVestingSchedule confirmed Block: 15 Gas used: 318839 (2.66%)

Transaction sent: 0xd4e3302ee065f5bf7336df3a41147f6b630e349f0644e8bb2c4bca6ee7a2ade
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
TokenVesting.revoke confirmed (UNAUTHORIZED) Block: 16 Gas used: 22860 (0.19%)

Transaction sent: 0xfbc4ee2cd21b349d421c5ec27b65b988c41cc1e589db3fde91768ae27c318901
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 14
TokenVesting.createVestingSchedule confirmed Block: 17 Gas used: 239627 (2.00%)

Transaction sent: 0x3f962ed19d00a5172a13c898494f141c194e6dba0b162553ef93f6899b9b9a06
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 15
TokenVesting.revoke confirmed (TokenVesting: vesting is not revocable) Block: 18 Gas used: 24635 (0.21%)

Transaction sent: 0xa4029f5e2fb7b3566a39b8c5e16859cfcaedc938426686842865fb8c0672c1dd
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 16
TokenVesting.revoke confirmed Block: 19 Gas used: 61392 (0.51%)

Transaction sent: 0x4cf9e10f3fffe70cd4104090566d8f4b7dd7ff6209766c16f529db1b44da62ae
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 17
TokenVesting.revoke confirmed (reverted) Block: 20 Gas used: 23626 (0.20%)

tests/test_token_vesting.py::test_revoke PASSED
```

```
tests/test_token_vesting.py::test_withdraw RUNNING
Transaction sent: 0x7f12c12947e2de949ad98f278a703f7fcecfc4787ff12eb3364f6add26ec36c5
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 18
ERC20Mock.constructor confirmed Block: 21 Gas used: 523834 (4.37%)
ERC20Mock deployed at: 0xed00238F9A0F7b4d93842033cdF56cCB32C781c2

Transaction sent: 0x9bc8a87e8ec108b7f1d618ad6eef2f9dc76bc068f6f7d639c3b62d0b07897f0f
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 19
TokenVesting.constructor confirmed Block: 22 Gas used: 1371816 (11.43%)
TokenVesting deployed at: 0xDae02e4fE488952cFB8c95177154D188647a0146

Transaction sent: 0x4fb752c1ebafa02d1ea95632aef67f9dcf59eaac852855c66f01901cc140d5ae
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
TokenVesting.withdraw confirmed (UNAUTHORIZED) Block: 23 Gas used: 28356 (0.24%)

Transaction sent: 0xb48249a7d9077b5be0b98b543b21a984219c5467edfa5e115c6bd6de11f58192
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 20
TokenVesting.withdraw confirmed (TokenVesting: not enough withdrawable funds) Block: 24 Gas used: 31527 (0.26%)

Transaction sent: 0x32201ac7a53edf86a2ba7a3b546e016bdb3340ad498123d0bcda23366a04a68d
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 21
ERC20Mock.mint confirmed Block: 25 Gas used: 65821 (0.55%)

Transaction sent: 0x9b15a2548a6d583e2f276747a823d5a3f57f168059d174e6f28ed4ecec8cffe3
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 22
TokenVesting.withdraw confirmed Block: 26 Gas used: 58376 (0.49%)

Transaction sent: 0xb0fd4d184d3ea49186742b506cce9460499a0975e0ecb20b315986cac7a89aab
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 23
TokenVesting.withdraw confirmed Block: 27 Gas used: 43376 (0.36%)

tests/test_token_vesting.py::test_withdraw PASSED
tests/test_token_vesting.py::test_release RUNNING
Transaction sent: 0x6963d74e753cccd05104d0ef58a4971164c19b37f3babdb42fe15f369896f351
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 24
ERC20Mock.constructor confirmed Block: 28 Gas used: 523834 (4.37%)
ERC20Mock deployed at: 0xf9C8Cf55f2E520B08d869df7bc76aa3d3ddDF913

Transaction sent: 0xd461bc0c2a685907380f5aeb5b1e91409f9fd7fd7d98bc2e2ff34846c25f01af
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 25
TokenVesting.constructor confirmed Block: 29 Gas used: 1371828 (11.43%)
TokenVesting deployed at: 0x654f70d8442EA18904FA1AD79114f7250F7E9336

Transaction sent: 0x7e7bae3b87911435b5e4408edac4d30f76ba7e35eb7c9be495cdcce4f0fe7934
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
TokenVesting.release confirmed (TokenVesting: only beneficiary and owner can release vested tokens) Block: 30 Gas used: 30259 (0.25%)

Transaction sent: 0x93f097979f310daf7546ae0ff85912760a38fa717b21b36421cc61db90b9900c
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 26
TokenVesting.release confirmed (Vested amount is 0) Block: 31 Gas used: 38896 (0.32%)

Transaction sent: 0x23a419f72729b3578ab98f621568d091851eaad8985cbe53666229558d1918d1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 27
ERC20Mock.mint confirmed Block: 32 Gas used: 65821 (0.55%)

Transaction sent: 0x8538c5650422e913d3bf7a44db2cbee5104fe8964b56ee1c706dc1c7fb3b296e
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 28
TokenVesting.createVestingSchedule confirmed Block: 33 Gas used: 318851 (2.66%)

Transaction sent: 0x99bb6a966fe737ebede0ee9a1999a7df6eca75c6e0850ec75be7fb4030feb914
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 29
TokenVesting.createVestingSchedule confirmed Block: 34 Gas used: 269627 (2.25%)

Transaction sent: 0xc45da4f80b53ef34862027623db5dfc7bce1aa6ca7376bc27dcaa40cdcd9ff9a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
TokenVesting.release confirmed (TokenVesting: only beneficiary and owner can release vested tokens) Block: 35 Gas used: 30475 (0.25%)

Transaction sent: 0xd914245caaa175ca9da38bec296375905840066ed86b8e42c5b591fc00277af4
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 30
TokenVesting.revoke confirmed Block: 36 Gas used: 61392 (0.51%)

Transaction sent: 0xfe9fd0c3878dd92d812dfd1d926d9716d2d27cdf6c99c48804fe8e25b359306ed
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
TokenVesting.release confirmed (reverted) Block: 37 Gas used: 28549 (0.24%)

Transaction sent: 0xc66335bc3f2b37fa1e585a124b3ca540db6d8807950176e9119055e562db32f2
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
TokenVesting.release confirmed (Vested amount is 0) Block: 38 Gas used: 38878 (0.32%)

Transaction sent: 0xa7409c10e59b9e10b79e86985235304b025b33516ffcebd6a640b8317cf5dc684
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
TokenVesting.release confirmed Block: 40 Gas used: 95486 (0.80%)

Transaction sent: 0x2417b01f2039afec83f5b7ba03dd9e91f9182041a650dd5bce2b16826c2e9e90
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 31
TokenVesting.createVestingSchedule confirmed Block: 41 Gas used: 258851 (2.16%)

Transaction sent: 0x8bbd13ccfe63735f69775b6353dc9c18740d3d5984ed5318280021lea64fbd33
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 32
TokenVesting.revoke confirmed Block: 43 Gas used: 131543 (1.10%)

tests/test_token_vesting.py::test_release PASSED
```


Annexes

Testing code:

Test Token Vesting:

```
from brownie import (

    reverts,

)

from scripts.helpful_scripts import (

    ZERO_ADDRESS,

    DAY_TIMESTAMP,

    increase_timestamp,

    get_timestamp,

    get_account,

)

from scripts.deploy import (

    deploy_erc,

    deploy_token_vesting

)

def test_create_vesting(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    token = deploy_erc(owner, "test", "test")
```

```

token_vesting = deploy_token_vesting(owner, token.address)

with reverts("UNAUTHORIZED"):

    token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),
        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,
        {"from": other})

    with reverts("TokenVesting: cannot create vesting schedule because not sufficient
tokens"):

        token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),
            DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,
            {"from": owner})

token.mint(token_vesting.address, 10e18)

with reverts("TokenVesting: duration must be > 0"):

    token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),
        DAY_TIMESTAMP * 10, 0, DAY_TIMESTAMP, True, 1e18,
        {"from": owner})

with reverts("TokenVesting: amount must be > 0"):

    token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),
        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 0,
        {"from": owner})

with reverts("TokenVesting: beneficiary cannot be the zero address"):

    token_vesting.createVestingSchedule(b"name", ZERO_ADDRESS, get_timestamp(),
        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,
        {"from": owner})

with reverts("TokenVesting: slicePeriodSeconds must be >= 1"):

    token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),
        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, 0, True, 1e18,

```

```

        {"from": owner})

with reverts("TokenVesting: duration must be >= cliff"):

    token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),

        DAY_TIMESTAMP * 15, DAY_TIMESTAMP * 10, DAY_TIMESTAMP, True, 1e18,

        {"from": owner})

tx = token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),

    DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,

    {"from": owner})

vesting_schedule_id = tx.events['VestingScheduleCreated'][0]['vestingScheduleId']

assert tx.events['VestingScheduleCreated'][0]['beneficiary'] == extra

assert tx.events['VestingScheduleCreated'][0]['amount'] == 1e18

def test_revoke(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    token = deploy_erc(owner, "test", "test")

    token_vesting = deploy_token_vesting(owner, token.address)

    token.mint(token_vesting.address, 10e18)

    tx = token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),

        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,

        {"from": owner})

    vesting_schedule_id = tx.events['VestingScheduleCreated'][0]['vestingScheduleId']

    with reverts("UNAUTHORIZED"):

        token_vesting.revoke(vesting_schedule_id, {"from": other})

```

```

tx = token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),
    DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, False, 1e18,
    {"from": owner})

vesting_schedule_id_not_revoke =
tx.events['VestingScheduleCreated'][0]['vestingScheduleId']

with reverts("TokenVesting: vesting is not revocable"):
    token_vesting.revoke(vesting_schedule_id_not_revoke, {"from": owner})

token_vesting.revoke(vesting_schedule_id, {"from": owner})

with reverts(""):
    token_vesting.revoke(vesting_schedule_id, {"from": owner})

def test_withdraw(only_local):
    # Arrange

    owner = get_account(0)
    other = get_account(1)

    token = deploy_erc(owner, "test", "test")
    token_vesting = deploy_token_vesting(owner, token.address)

    with reverts("UNAUTHORIZED"):
        token_vesting.withdraw(1e18, {"from": other})

    with reverts("TokenVesting: not enough withdrawable funds"):
        token_vesting.withdraw(1e18, {"from": owner})

    token.mint(token_vesting.address, 10e18)

    tx = token_vesting.withdraw(1e18, {"from": owner})

    assert tx.events['Transfer'][0]['from'] == token_vesting.address

```

```

assert tx.events['Transfer'][0]['to'] == owner

assert tx.events['Transfer'][0]['value'] == 1e18

tx = token_vesting.withdraw(2e18, {"from": owner})

assert tx.events['Transfer'][0]['from'] == token_vesting.address

assert tx.events['Transfer'][0]['to'] == owner

assert tx.events['Transfer'][0]['value'] == 2e18

```

```

def test_release(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    token = deploy_erc(owner, "test", "test")

    token_vesting = deploy_token_vesting(owner, token.address)

    with reverts("TokenVesting: only beneficiary and owner can release vested tokens"):

        token_vesting.release(b'random_vesting', {"from": extra})

    with reverts("Vested amount is 0"):

        token_vesting.release(b'random_vesting', {"from": owner})

    token.mint(token_vesting.address, 10e18)

    tx = token_vesting.createVestingSchedule(b"name", other, get_timestamp(),

        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,

        {"from": owner})

    vesting_schedule_id = tx.events['VestingScheduleCreated'][0]['vestingScheduleId']

    tx = token_vesting.createVestingSchedule(b"name", extra, get_timestamp(),

```



```

        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, False, 1e18,

        {"from": owner})

    vesting_schedule_id_not_revoke =
tx.events['VestingScheduleCreated'][0]['vestingScheduleId']

    with reverts("TokenVesting: only beneficiary and owner can release vested tokens"):

        token_vesting.release(vesting_schedule_id, {"from": extra})

    token_vesting.revoke(vesting_schedule_id, {"from": owner})

    with reverts(""):

        token_vesting.release(vesting_schedule_id, {"from": other})

    with reverts("Vested amount is 0"):

        token_vesting.release(vesting_schedule_id_not_revoke, {"from": extra})

    increase_timestamp(DAY_TIMESTAMP * 15)

    tx = token_vesting.release(vesting_schedule_id_not_revoke, {"from": extra})

    assert tx.events['Transfer'][0]['from'] == token_vesting.address
    assert tx.events['Transfer'][0]['to'] == extra
    assert tx.events['Transfer'][0]['value'] > 0
    assert tx.events['Transfer'][0]['value'] < 1e18

    tx = token_vesting.createVestingSchedule(b"name", other, get_timestamp(),
        DAY_TIMESTAMP * 10, DAY_TIMESTAMP * 15, DAY_TIMESTAMP, True, 1e18,

        {"from": owner})

    vesting_schedule_id = tx.events['VestingScheduleCreated'][0]['vestingScheduleId']

    increase_timestamp(DAY_TIMESTAMP * 15)

    tx = token_vesting.revoke(vesting_schedule_id, {"from": owner})

    assert tx.events['Transfer'][0]['from'] == token_vesting.address

```





```
assert tx.events['Transfer'][0]['to'] == other

assert tx.events['Transfer'][0]['value'] > 0

assert tx.events['Transfer'][0]['value'] < 1e18
```

Technical Findings Summary

Findings

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

Assessment Results

Score Results

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