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🟡 Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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 **Well Secured.**

Insecure	**Poorly Secured**	**Secured**	**Well 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	StakeNFT.sol [SHA256] b92c1c06b13e0059e0731766b7873ba355843a9a65ab37c13ed07651c5628d73
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

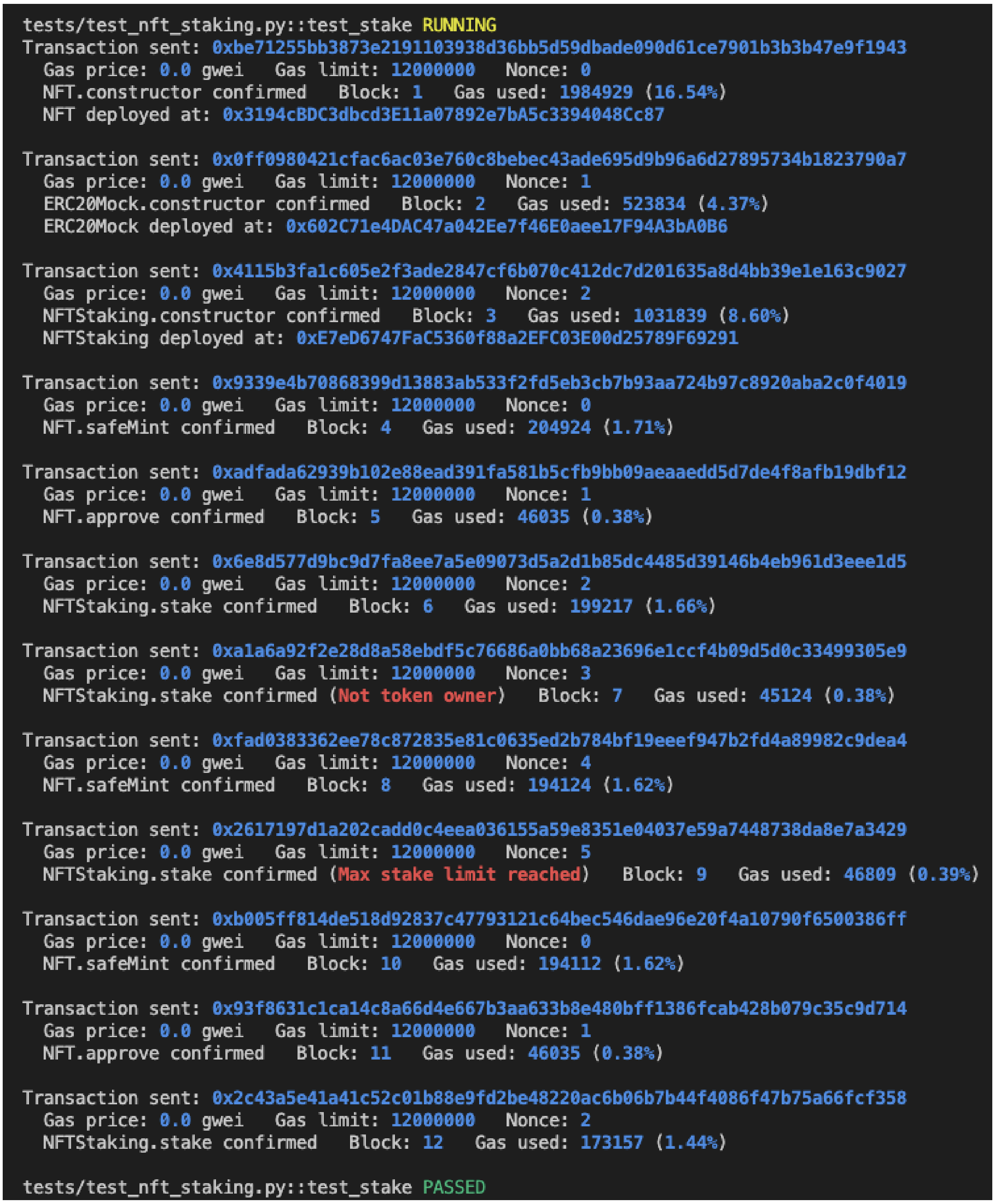
No informational severity issues were found.

Testing coverage

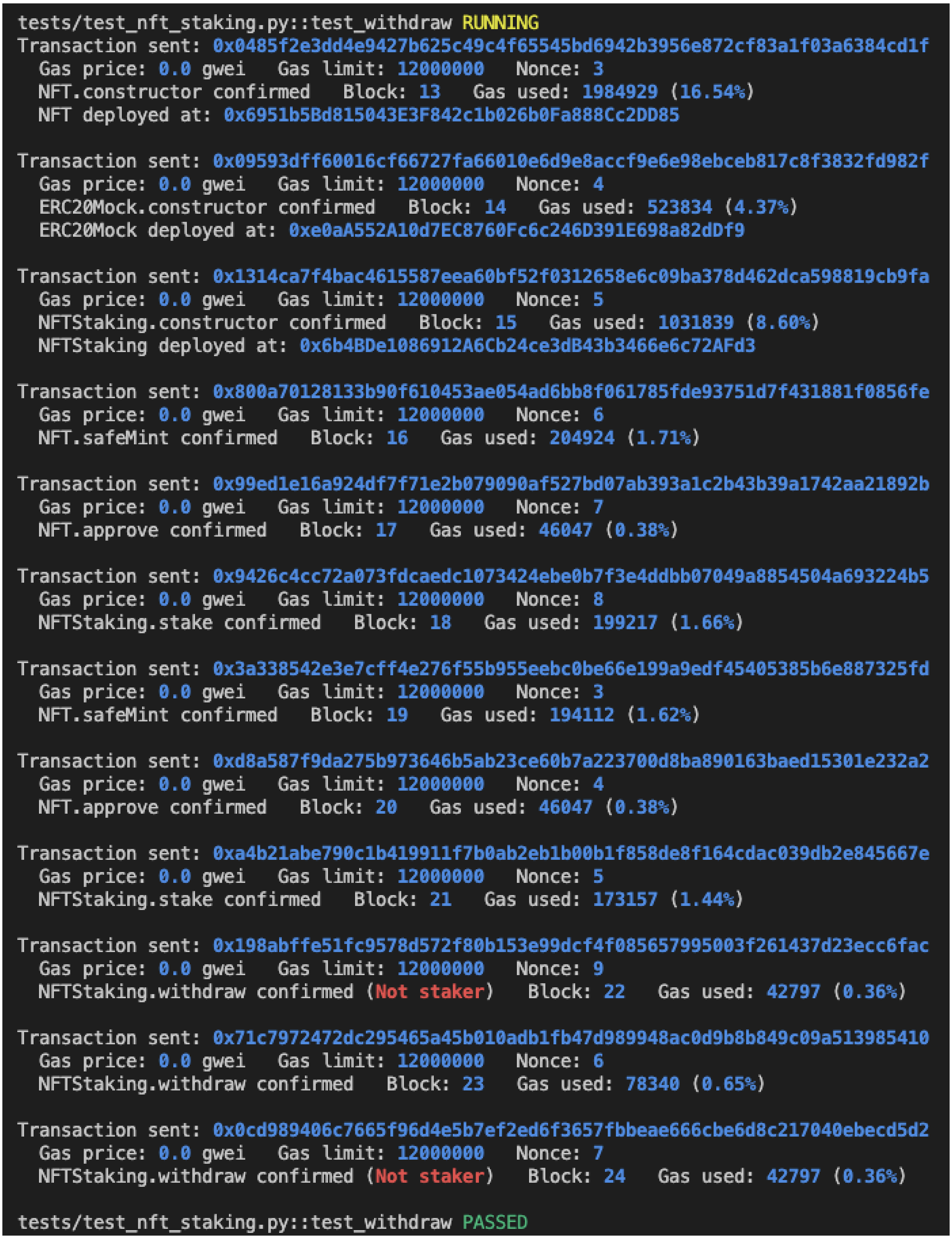
During the testing phase, custom use cases were written to cover all the logic of contracts. *Check “Annexes” to see the testing code.*

Coincreate contract tests:

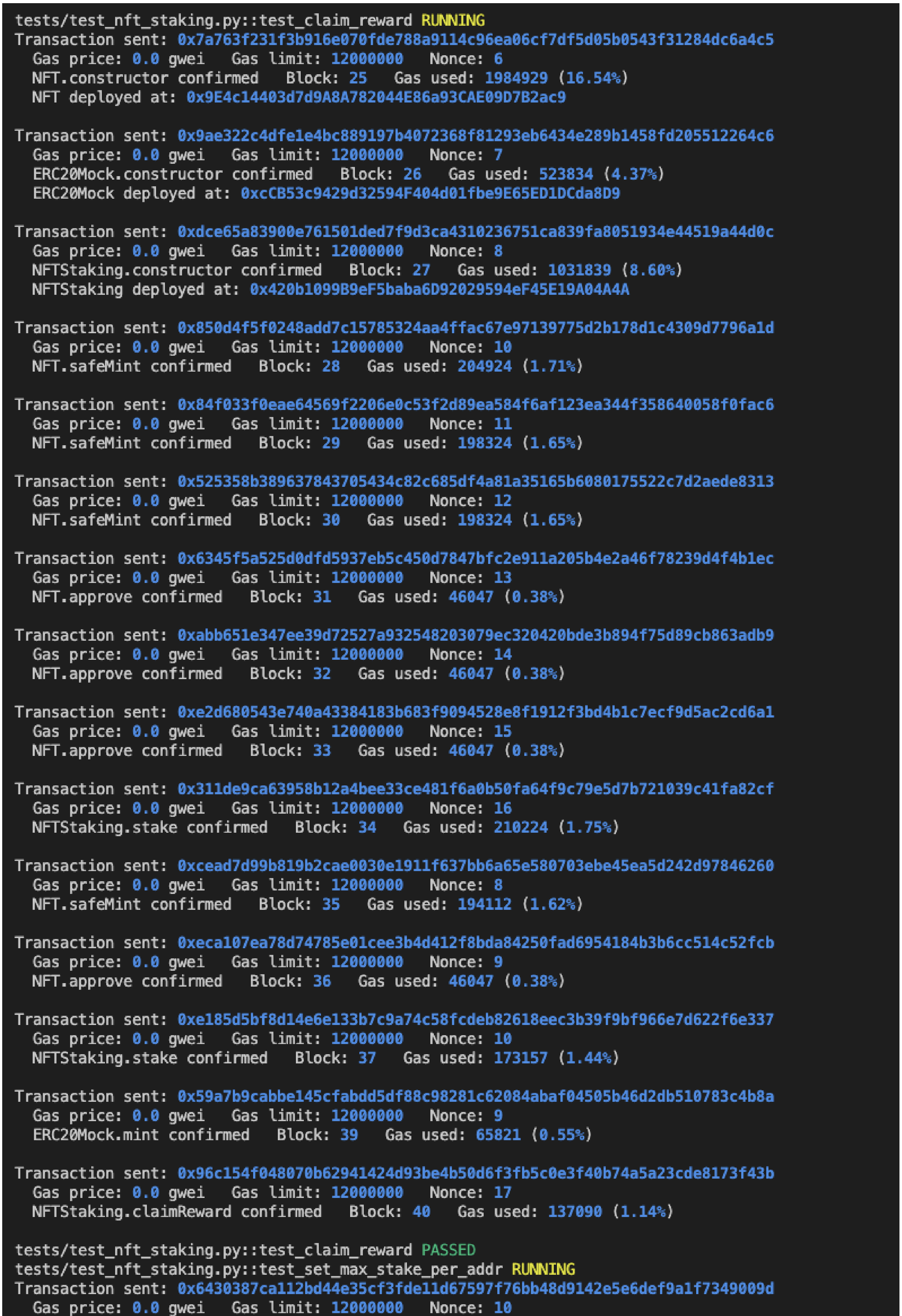
Test NFT Staking:



```
tests/test_nft_staking.py::test_stake RUNNING
Transaction sent: 0xbe71255bb3873e2191103938d36bb5d59dbade090d61ce7901b3b3b47e9f1943
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  NFT.constructor confirmed   Block: 1   Gas used: 1984929 (16.54%)
  NFT deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x0ff0980421cfac6ac03e760c8bebec43ade695d9b96a6d27895734b1823790a7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ERC20Mock.constructor confirmed   Block: 2   Gas used: 523834 (4.37%)
  ERC20Mock deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0x4115b3fa1c605e2f3ade2847cf6b070c412dc7d201635a8d4bb39e1e163c9027
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  NFTStaking.constructor confirmed   Block: 3   Gas used: 1031839 (8.60%)
  NFTStaking deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0x9339e4b70868399d13883ab533f2fd5eb3cb7b93aa724b97c8920aba2c0f4019
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  NFT.safeMint confirmed   Block: 4   Gas used: 204924 (1.71%)

Transaction sent: 0xadfada62939b102e88ead391fa581b5cfb9bb09aeaaedd5d7de4f8afb19dbf12
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  NFT.approve confirmed   Block: 5   Gas used: 46035 (0.38%)

Transaction sent: 0x6e8d577d9bc9d7fa8ee7a5e09073d5a2d1b85dc4485d39146b4eb961d3eee1d5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  NFTStaking.stake confirmed   Block: 6   Gas used: 199217 (1.66%)

Transaction sent: 0xa1a6a92f2e28d8a58ebdf5c76686a0bb68a23696e1ccf4b09d5d0c33499305e9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  NFTStaking.stake confirmed (Not token owner)   Block: 7   Gas used: 45124 (0.38%)

Transaction sent: 0xfad0383362ee78c872835e81c0635ed2b784bf19eeef947b2fd4a89982c9dea4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  NFT.safeMint confirmed   Block: 8   Gas used: 194124 (1.62%)

Transaction sent: 0x2617197d1a202cadd0c4eea036155a59e8351e04037e59a7448738da8e7a3429
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  NFTStaking.stake confirmed (Max stake limit reached)   Block: 9   Gas used: 46809 (0.39%)

Transaction sent: 0xb005ff814de518d92837c47793121c64bec546dae96e20f4a10790f6500386ff
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  NFT.safeMint confirmed   Block: 10   Gas used: 194112 (1.62%)

Transaction sent: 0x93f8631c1ca14c8a66d4e667b3aa633b8e480bff1386fcab428b079c35c9d714
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  NFT.approve confirmed   Block: 11   Gas used: 46035 (0.38%)

Transaction sent: 0x2c43a5e41a41c52c01b88e9fd2be48220ac6b06b7b44f4086f47b75a66fcf358
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  NFTStaking.stake confirmed   Block: 12   Gas used: 173157 (1.44%)

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



```
tests/test_nft_staking.py::test_withdraw RUNNING
Transaction sent: 0x0485f2e3dd4e9427b625c49c4f65545bd6942b3956e872cf83a1f03a6384cd1f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  NFT.constructor confirmed   Block: 13   Gas used: 1984929 (16.54%)
  NFT deployed at: 0x6951b5Bd815043E3F842c1b026b0Fa888Cc2DD85

Transaction sent: 0x09593dff60016cf66727fa66010e6d9e8accf9e6e98ebceb817c8f3832fd982f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ERC20Mock.constructor confirmed   Block: 14   Gas used: 523834 (4.37%)
  ERC20Mock deployed at: 0xe0aA552A10d7EC8760Fc6c246D391E698a82dDf9

Transaction sent: 0x1314ca7f4bac4615587eea60bf52f0312658e6c09ba378d462dca598819cb9fa
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  NFTStaking.constructor confirmed   Block: 15   Gas used: 1031839 (8.60%)
  NFTStaking deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0x800a70128133b90f610453ae054ad6bb8f061785fde93751d7f431881f0856fe
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  NFT.safeMint confirmed   Block: 16   Gas used: 204924 (1.71%)

Transaction sent: 0x99ed1e16a924df7f71e2b079090af527bd07ab393a1c2b43b39a1742aa21892b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  NFT.approve confirmed   Block: 17   Gas used: 46047 (0.38%)

Transaction sent: 0x9426c4cc72a073fdcaedc1073424ebe0b7f3e4ddbb07049a8854504a693224b5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  NFTStaking.stake confirmed   Block: 18   Gas used: 199217 (1.66%)

Transaction sent: 0x3a338542e3e7cff4e276f55b955eebc0be66e199a9edf45405385b6e887325fd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  NFT.safeMint confirmed   Block: 19   Gas used: 194112 (1.62%)

Transaction sent: 0xd8a587f9da275b973646b5ab23ce60b7a223700d8ba890163baed15301e232a2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  NFT.approve confirmed   Block: 20   Gas used: 46047 (0.38%)

Transaction sent: 0xa4b21abe790c1b419911f7b0ab2eb1b00b1f858de8f164cdac039db2e845667e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  NFTStaking.stake confirmed   Block: 21   Gas used: 173157 (1.44%)

Transaction sent: 0x198abffe51fc9578d572f80b153e99dcf4f085657995003f261437d23ecc6fac
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  NFTStaking.withdraw confirmed (Not staker)   Block: 22   Gas used: 42797 (0.36%)

Transaction sent: 0x71c7972472dc295465a45b010adb1fb47d989948ac0d9b8b849c09a513985410
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  NFTStaking.withdraw confirmed   Block: 23   Gas used: 78340 (0.65%)

Transaction sent: 0x0cd989406c7665f96d4e5b7ef2ed6f3657fbbeae666cbe6d8c217040ebecd5d2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  NFTStaking.withdraw confirmed (Not staker)   Block: 24   Gas used: 42797 (0.36%)

tests/test_nft_staking.py::test_withdraw PASSED
```



```
tests/test_nft_staking.py::test_claim_reward RUNNING
Transaction sent: 0x7a763f231f3b916e070fde788a9114c96ea06cf7df5d05b0543f31284dc6a4c5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  NFT.constructor confirmed   Block: 25   Gas used: 1984929 (16.54%)
  NFT deployed at: 0x9E4c14403d7d9A8A782044E86a93CAE09D7B2ac9

Transaction sent: 0x9ae322c4dfe1e4bc889197b4072368f81293eb6434e289b1458fd205512264c6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  ERC20Mock.constructor confirmed   Block: 26   Gas used: 523834 (4.37%)
  ERC20Mock deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0xdce65a83900e761501ded7f9d3ca4310236751ca839fa8051934e44519a44d0c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  NFTStaking.constructor confirmed   Block: 27   Gas used: 1031839 (8.60%)
  NFTStaking deployed at: 0x420b1099B9eF5baba6D92029594eF45E19A04A4A

Transaction sent: 0x850d4f5f0248add7c15785324aa4ffac67e97139775d2b178d1c4309d7796a1d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  NFT.safeMint confirmed   Block: 28   Gas used: 204924 (1.71%)

Transaction sent: 0x84f033f0eae64569f2206e0c53f2d89ea584f6af123ea344f358640058f0fac6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  NFT.safeMint confirmed   Block: 29   Gas used: 198324 (1.65%)

Transaction sent: 0x525358b389637843705434c82c685df4a81a35165b6080175522c7d2aede8313
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  NFT.safeMint confirmed   Block: 30   Gas used: 198324 (1.65%)

Transaction sent: 0x6345f5a525d0dfd5937eb5c450d7847bfc2e911a205b4e2a46f78239d4f4b1ec
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  NFT.approve confirmed   Block: 31   Gas used: 46047 (0.38%)

Transaction sent: 0xabb651e347ee39d72527a932548203079ec320420bde3b894f75d89cb863adb9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  NFT.approve confirmed   Block: 32   Gas used: 46047 (0.38%)

Transaction sent: 0xe2d680543e740a43384183b683f9094528e8f1912f3bd4b1c7ecf9d5ac2cd6a1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  NFT.approve confirmed   Block: 33   Gas used: 46047 (0.38%)

Transaction sent: 0x311de9ca63958b12a4bee33ce481f6a0b50fa64f9c79e5d7b721039c41fa82cf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  NFTStaking.stake confirmed   Block: 34   Gas used: 210224 (1.75%)

Transaction sent: 0xcead7d99b819b2cae0030e1911f637bb6a65e580703ebe45ea5d242d97846260
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  NFT.safeMint confirmed   Block: 35   Gas used: 194112 (1.62%)

Transaction sent: 0xeca107ea78d74785e01cee3b4d412f8bda84250fad6954184b3b6cc514c52fcb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  NFT.approve confirmed   Block: 36   Gas used: 46047 (0.38%)

Transaction sent: 0xe185d5bf8d14e6e133b7c9a74c58fcdeb82618eec3b39f9bf966e7d622f6e337
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  NFTStaking.stake confirmed   Block: 37   Gas used: 173157 (1.44%)

Transaction sent: 0x59a7b9cabbe145cfabdd5df88c98281c62084abaf04505b46d2db510783c4b8a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  ERC20Mock.mint confirmed   Block: 39   Gas used: 65821 (0.55%)

Transaction sent: 0x96c154f048070b62941424d93be4b50d6f3fb5c0e3f40b74a5a23cde8173f43b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  NFTStaking.claimReward confirmed   Block: 40   Gas used: 137090 (1.14%)

tests/test_nft_staking.py::test_claim_reward PASSED
tests/test_nft_staking.py::test_set_max_stake_per_addr RUNNING
Transaction sent: 0x6430387ca112bd44e35cf3fde11d67597f76bb48d9142e5e6def9a1f7349009d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
```

Annexes

Testing code:

Test Stoken:

```
from brownie import (

    reverts,

)


from scripts.helpful_scripts import (

    ZERO_ADDRESS,

    DAY_TIMESTAMP,

    increase_timestamp,

    get_account,

)


from scripts.deploy import (

    deploy_weth,

    deploy_factory,

    deploy_router,

    deploy_liquidity,

    deploy_erc,

    deploy_nft,

    deploy_stake_nft,

    deploy_stoken

)


def test_constructor(only_local):

    # Arrange
    owner = get_account(0)

    admin_wallet = get_account(8)


    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)

    router = deploy_router(owner, factory.address, weth.address)


    with reverts("Admin wallet cannot be zero address"):

        deploy_stoken(owner, 1000e18, 10, 10, 10e18, 5e18, 10e18, router.address,

                      ZERO_ADDRESS, admin_wallet, False, 10e18)

    with reverts("Owner wallet cannot be zero address"):

        deploy_stoken(owner, 1000e18, 10, 10, 10e18, 5e18, 10e18, router.address,

                      admin_wallet, ZERO_ADDRESS, False, 10e18)

    with reverts("Cannot set LpFee more then 10%"):

        deploy_stoken(owner, 1000e18, 20, 10, 10e18, 5e18, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)

    with reverts("Cannot set adminFee more then 30%"):

        deploy_stoken(owner, 1000e18, 10, 40, 10e18, 5e18, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)

    with reverts("Invalid router address"):

        deploy_stoken(owner, 1000e18, 10, 10, 10e18, 5e18, 10e18, ZERO_ADDRESS,

                      admin_wallet, admin_wallet, False, 10e18)

    with reverts("Max wallet amount must be at least 0.5% of total supply"):

        deploy_stoken(owner, 1000e18, 10, 10, 10e18, 0, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)

    with reverts("Max transaction amount cannot be less than 0.1% of total supply"):

        deploy_stoken(owner, 1000e18, 10, 10, 0, 500e18, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)
```

```python
    deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,
                  admin_wallet, admin_wallet, False, 10e18)


def test_set_token_uri(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    admin_wallet = get_account(8)

    weth = deploy_weth(owner)
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)
    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,
                  admin_wallet, admin_wallet, False, 10e18)

    with reverts("Ownable: caller is not the owner"):
        stoken.setTokenURI("some_uri", {"from": other})
    assert stoken.tokenURI() == "some_uri"
    stoken.setTokenURI("new_token_uri", {"from": admin_wallet})
    assert stoken.tokenURI() == "new_token_uri"

def test_set_max_wallet_amount(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    admin_wallet = get_account(8)
    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)

    router = deploy_router(owner, factory.address, weth.address)

    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)


    with reverts("Ownable: caller is not the owner"):

        stoken.setMaxWalletAmount(501e18, {"from": other})

    with reverts("Max wallet amount must be at least 0.5% of total supply"):

        stoken.setMaxWalletAmount(1e18, {"from": admin_wallet})

    assert stoken._maxWalletAmount() == 500e18

    stoken.setMaxWalletAmount(501e18, {"from": admin_wallet})

    assert stoken._maxWalletAmount() == 501e18


def test_set_max_wallet_amount(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    admin_wallet = get_account(8)


    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)

    router = deploy_router(owner, factory.address, weth.address)

    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)


    with reverts("Ownable: caller is not the owner"):
```

```python
        stoken.setAdminFeeThreshold(20e18, {"from": other})

    assert stoken.adminFeeThreshold() == 10e18

    stoken.setAdminFeeThreshold(20e18, {"from": admin_wallet})

    assert stoken.adminFeeThreshold() == 20e18


def test_set_liq_fee_percent(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    admin_wallet = get_account(8)


    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)

    router = deploy_router(owner, factory.address, weth.address)

    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,

                      admin_wallet, admin_wallet, False, 10e18)


    with reverts("Ownable: caller is not the owner"):

        stoken.setLiquidityFeePercent(5, {"from": other})

    with reverts("Cannot exceed the max liquidity fee"):

        stoken.setLiquidityFeePercent(15, {"from": admin_wallet})

    assert stoken._liquidityFee() == 10

    stoken.setLiquidityFeePercent(5, {"from": admin_wallet})

    assert stoken._liquidityFee() == 5


def test_set_admin_fee_perc(only_local):

    # Arrange
    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    admin_wallet = get_account(8)


    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)

    router = deploy_router(owner, factory.address, weth.address)

    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,

                  admin_wallet, admin_wallet, False, 10e18)


    with reverts("Ownable: caller is not the owner"):

        stoken.setAdminFeePercent(5, {"from": other})

    with reverts("Cannot exceed the max admin fee"):

        stoken.setAdminFeePercent(15, {"from": admin_wallet})

    assert stoken._adminFee() == 10

    stoken.setAdminFeePercent(5, {"from": admin_wallet})

    assert stoken._adminFee() == 5


def test_set_admin_wallet(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    admin_wallet = get_account(8)


    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)
    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,
                           admin_wallet, admin_wallet, False, 10e18)

    with reverts("Ownable: caller is not the owner"):
        stoken.setAdminWallet(extra, {"from": other})
    with reverts("Admin wallet cannot be zero address"):
        stoken.setAdminWallet(ZERO_ADDRESS, {"from": admin_wallet})
    assert stoken._adminWallet() == admin_wallet
    stoken.setAdminWallet(extra, {"from": admin_wallet})
    assert stoken._adminWallet() == extra

def test_set_max_tx_amount(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    admin_wallet = get_account(8)

    weth = deploy_weth(owner)
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)
    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,
                           admin_wallet, admin_wallet, False, 10e18)

    with reverts("Ownable: caller is not the owner"):
        stoken.setMaxTxAmount(50e18, {"from": other})
    with reverts("Max transaction amount cannot be less than 0.1% of total supply"):
```

```python
        stoken.setMaxTxAmount(0.5e18, {"from": admin_wallet})
    assert stoken._maxTxAmount() == 100e18
    stoken.setMaxTxAmount(50e18, {"from": admin_wallet})
    assert stoken._maxTxAmount() == 50e18


def test_set_max_tx_amount(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    admin_wallet = get_account(8)

    weth = deploy_weth(owner)
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)
    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,
                          admin_wallet, admin_wallet, False, 10e18)

    with reverts("Ownable: caller is not the owner"):
        stoken.setMaxTxAmount(50e18, {"from": other})
    with reverts("Max transaction amount cannot be less than 0.1% of total supply"):
        stoken.setMaxTxAmount(0.5e18, {"from": admin_wallet})
    assert stoken._maxTxAmount() == 100e18
    stoken.setMaxTxAmount(50e18, {"from": admin_wallet})
    assert stoken._maxTxAmount() == 50e18


def test_transfer(only_local):
    # Arrange
    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    admin_wallet = get_account(8)


    weth = deploy_weth(owner)

    factory = deploy_factory(owner, owner)

    router = deploy_router(owner, factory.address, weth.address)

    stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,

                     admin_wallet, admin_wallet, False, 10e18)


    pair = stoken.uniswapV2Pair()


    with reverts("ERC20: transfer from the zero address"):

        stoken.transfer(other, 1e18, {"from": ZERO_ADDRESS})

    with reverts("ERC20: transfer to the zero address"):

        stoken.transfer(ZERO_ADDRESS, 1e18, {"from": other})

    with reverts("Transfer amount must be greater than zero"):

        stoken.transfer(extra, 0, {"from": other})

    with reverts("Insufficient balance"):

        stoken.transfer(extra, 1e18, {"from": other})

    with reverts("Max wallet amount exceeded"):

        stoken.transfer(extra, 600e18, {"from": other})


    tx = stoken.transfer(other, 5e18, {"from": admin_wallet})

    assert tx.events['Transfer'][0]['from'] == admin_wallet

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

    assert tx.events['Transfer'][0]['value'] == 5e18
```

```python
tx = stoken.transfer(extra, 2e18, {"from": other})
assert tx.events['Transfer'][0]['from'] == other
assert tx.events['Transfer'][0]['to'] == extra
assert tx.events['Transfer'][0]['value'] == 2e18

# 10% lp fee + 10% adm fee
tx = stoken.transfer(pair, 1e18, {"from": other})
assert tx.events['Transfer'][0]['from'] == other
assert tx.events['Transfer'][0]['to'] == pair
assert tx.events['Transfer'][0]['value'] == 0.8e18

tx = stoken.transfer(other, 0.8e18, {"from": pair})
assert tx.events['Transfer'][0]['from'] == pair
assert tx.events['Transfer'][0]['to'] == other
assert tx.events['Transfer'][0]['value'] == 0.64e18

stoken = deploy_stoken(owner, 1000e18, 10, 10, 100e18, 500e18, 10e18, router.address,
                  admin_wallet, admin_wallet, True, 10e18)

pair = stoken.uniswapV2Pair()
tx = stoken.transfer(other, 5e18, {"from": admin_wallet})
assert tx.events['Transfer'][0]['from'] == admin_wallet
assert tx.events['Transfer'][0]['to'] == other
assert tx.events['Transfer'][0]['value'] == 5e18

# 10% lp fee + 10% adm fee
tx = stoken.transfer(extra, 2e18, {"from": other})
```

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

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

    assert tx.events['Transfer'][0]['value'] == 1.6e18
```

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.