Assure DeFi®

THE VERIFICATION **GOLD STANDARD**

Security Assessment

Profit_IQ

Date: 14/04/2024

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Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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

Insecure	Poorly Secured	Secured	Well Secured

Scope

Target Code And Revision

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

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	https://github.com/raptor042/ETH_Miner/blob/master/contracts/Bank.sol Bank.sol - [Commit] *21f148879b2f4766850895dc316ac6e6c27d60cc* Fixed version - [Sepolia] *0x32FcBAC5E32749a03F5f8A18a72a270D6bde08d0* https://github.com/raptor042/ETH_Miner/blob/master/contracts/Miner.sol Miner.sol - [Commit] *21f148879b2f4766850895dc316ac6e6c27d60cc* Fixed version - [Sepolia] *0x42004d4c4F39F0D91740bA5ed7841C9a8C3D3372*
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

1. Reentrancy Attack Vulnerability in Reward Functions [Fixed ✅]

Contract: Miner

Functions: claimRewards(), withdraw()

Issue: The functions lack a reentrancy guard, which poses a high risk of reentrancy attacks where external calls can lead to unintended contract behavior.

Mitigation: Incorporate the nonReentrant modifier from OpenZeppelin contracts into these functions.

Fix: Reentrancy ward added on the contracts.

MEDIUM

2. Arbitrary Reward Setting Vulnerability in claimRewards() and withdraw() [Fixed ✅]

Contract: Miner

Functions: claimRewards(), withdraw()

Issue: The contract currently allows any reward amount to be set via the function argument. This can lead to unexpected or manipulated rewards being issued.

Mitigation: Modify the functions to calculate rewards internally using a calculateRewards() function specific to the msg.sender. This approach ensures that rewards are computed based on predefined rules, minimizing the risk of erroneous payouts.

Fix: Calculate reward function was fixed.

LOW

1. Address Zero Assignment Vulnerability Across Contract Functions [Fixed ✅]

Contract: Miner, Bank

Functions:

Miner: constructor(_bank, _wallet1, _wallet2) , changeTFeeWallet(), changePFeeWallet(), changeBank(_bank), claimRewards(), withdraw()

Bank: setAdmin(_admin)

Issue: Within both the Miner and Bank contracts, several functions are at risk of receiving and processing a zero address. This vulnerability can lead to unintended behaviors and could potentially disrupt contract management by mistakenly assigning a zero address to key contract functionalities.

Mitigation: To address this security risk, a robust validation mechanism should be implemented across all affected functions. Specifically, a require() statement should be added to verify that no zero addresses (address(0)) are used when setting or updating addresses. This should be applied in:

Miner Contract: In the constructor to ensure initial addresses are valid, and in the changeTFeeWallet(), changePFeeWallet(), and changeBank(_bank) functions to secure address updates.

Bank Contract: In the setAdmin(_admin) function to guard administrative controls.

This check, formatted as require(address_variable != address(0), "Address cannot be zero."), ensures the use of valid, non-null addresses across all critical operations, thereby reducing the risk of security lapses due to erroneous or malicious address inputs. Implementing this mitigation uniformly across related functions strengthens the contracts' defenses against common vulnerabilities associated with address handling.

Fix: 0 check controls were added.

INFORMATIONAL

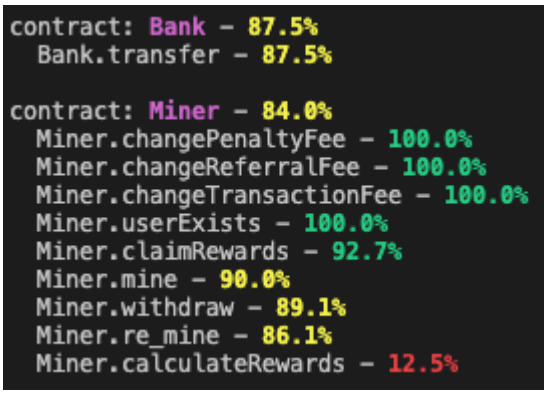
No Informational severity issues were found.

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.*

Profit_IQ contracts tests:

Coverages:



Testing Profit_IQ Token:

```
tests/test_bank.py::test_set_admin RUNNING
Transaction sent: 0x32dc56a646e8778a9c962f6faf0fa0f54949b92715399eee5c7fdb75b7f444cf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Bank.constructor confirmed   Block: 1   Gas used: 262651 (2.19%)
  Bank deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x05967bac6acec557173810c78cd18dbfba4bde78a4ab6097565fd0bee4c9b810
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Bank.setAdmin confirmed (Only owner can call this function.)   Block: 2   Gas used: 22660 (0.19%)

Transaction sent: 0xb0be0c77402261444bb3ad78c15c12926e491da018197f22ca15c04e198a7f4e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  Bank.setAdmin confirmed   Block: 3   Gas used: 43409 (0.36%)

tests/test_bank.py::test_set_admin PASSED
tests/test_bank.py::test_transfer RUNNING
Transaction sent: 0xcfddfbe62270b70cdcb3e97becf544e149b5fda185affe15a238b9d53bf2a5d2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  Bank.constructor confirmed   Block: 4   Gas used: 262651 (2.19%)
  Bank deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0x8bb956197079cb3ed604555da973810fe2b1d40c961f244099fd37ec98d1bb66
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  Bank.transfer confirmed (Only admin can call this function.)   Block: 5   Gas used: 22936 (0.19%)

Transaction sent: 0xd6621b69a24ef9dcaa5d93f85d4de523965516809fc8e6d459ce2fae6eff0bb8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  Bank.setAdmin confirmed   Block: 6   Gas used: 43421 (0.36%)

Transaction sent: 0xf782cf8b45791741071b702134d60849eeda293c3e853159a02ea82e1d08b241
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  Bank.transfer confirmed (Insufficent funds.)   Block: 7   Gas used: 22943 (0.19%)

Transaction sent: 0x8a16532f16733d478c472be5ef9f16c825efb98b1e001bd1e4a8c090a63a01ab
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  Bank.transfer confirmed   Block: 9   Gas used: 31838 (0.27%)

tests/test_bank.py::test_transfer PASSED
```

```
tests/test_miner.py::test_change_fee RUNNING
Transaction sent: 0x4c679e47dd6c32ead360f51b0ed65d11ff577afd51402f406e144655e737825b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  Bank.constructor confirmed   Block: 10   Gas used: 262651 (2.19%)
  Bank deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0x07b487af0427fd9df7c32e846ad348859259a1acd408f99cf6e41ad83d9a6f33
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  Miner.constructor confirmed   Block: 11   Gas used: 1638074 (13.65%)
  Miner deployed at: 0x9E4c14403d7d9A8A782044E86a93CAE09D7B2ac9

Transaction sent: 0x71c563b25496f250eafc0fcad921438870c92ca69d4ccf76ffb21211188d464d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  Miner.changeTransactionFee confirmed (Only owner can call this function.)   Block: 12   Gas used: 22428 (0.19%)

Transaction sent: 0x948ac97b9b9ff5902b28c3c16340f34be45f82a189ba2fa67b06a08cb487fdff
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  Miner.changeTransactionFee confirmed (Transaction fee cannot exceed 5%)   Block: 13   Gas used: 22412 (0.19%)

Transaction sent: 0xa298e0d696677898c575821f5d899d73f8b0103df3f0222b95b31e82708f8ffa
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  Miner.changeTransactionFee confirmed   Block: 14   Gas used: 27305 (0.23%)

Transaction sent: 0xe8d26d53422691f7ea7ad85002fda2865e56a33324e8866b307ab55ee3588a77
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  Miner.changeReferralFee confirmed (Only owner can call this function.)   Block: 15   Gas used: 22506 (0.19%)

Transaction sent: 0x8478bfc5160f077f21cf2a0ce45b63cce2b1921b178d563fc032169884850ca7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  Miner.changeReferralFee confirmed (Referral fee cannot exceed 5%)   Block: 16   Gas used: 22487 (0.19%)

Transaction sent: 0xcab482a344181bbbacb63dcab2ecd848c4d49553e2b5469284f09e0ba7887592
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  Miner.changeReferralFee confirmed   Block: 17   Gas used: 27383 (0.23%)

Transaction sent: 0xcb415d96aeeedc27d2915068fc9a121ae972353081a0d081c0d125c3c2cf549a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  Miner.changePenaltyFee confirmed (Only owner can call this function.)   Block: 18   Gas used: 22439 (0.19%)

Transaction sent: 0x45b3571f0d95482d80c11bf4e80f401aa08a874d7c915eef39e7a88e3dfb83ff
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  Miner.changePenaltyFee confirmed (Penalty fee cannot exceed 50%)   Block: 19   Gas used: 22420 (0.19%)

Transaction sent: 0x06a20c20a6e3ba9ae51bfff99623bb6023407b6e209a90fd5339b099fb320bef
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  Miner.changePenaltyFee confirmed   Block: 20   Gas used: 27316 (0.23%)

tests/test_miner.py::test_change_fee PASSED
```

```
tests/test_miner.py::test_mine RUNNING
Transaction sent: 0x9155c70a643474285132c4bcf5fabf377cdcfca320e02eeb833d609ab6ad4723
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  Bank.constructor confirmed   Block: 21   Gas used: 262651 (2.19%)
  Bank deployed at: 0xe692Cf21B12e0B2717C4bF647F9768Fa58861c8b

Transaction sent: 0x22d5842f88320fd055c3b75a269603faee7b93318e28748476430dd233f01a0d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  Miner.constructor confirmed   Block: 22   Gas used: 1638074 (13.65%)
  Miner deployed at: 0xe65A7a341978d59d40d30FC23F5014FACB4f575A

Transaction sent: 0xc7161b31d071456a030abc542cc27f82f7b6c8245ce9fba8d317466336d29cdc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  Miner.mine confirmed (Insufficent deposit amount.)   Block: 23   Gas used: 23445 (0.20%)

Transaction sent: 0xe9de2c7672ef44455568c877acdd67613e02c14c1e99372705ab149bafca9d5c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  Miner.mine confirmed   Block: 24   Gas used: 241319 (2.01%)

Transaction sent: 0x4e60eec0946b48c9e26a3793c36c2307b2487ac007b90df117975ac4624d9a66
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  Miner.mine confirmed   Block: 25   Gas used: 57045 (0.48%)

Transaction sent: 0xf73e93e94268740e30573e09478783531070e437ff8130d38806327ef52dbb77
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Miner.mine confirmed   Block: 26   Gas used: 309361 (2.58%)

Transaction sent: 0xe531b09d4f28416bc92bb608aa6439c0a89fe5eb31430cb2df9dd102a7f193fa
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Miner.re_mine confirmed (No user account detected.)   Block: 27   Gas used: 27560 (0.23%)

Transaction sent: 0xe2a6a302c66bfe3b1998c9969715c8a3117e784062e76708ea64e36de6ddb571
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  Miner.re_mine confirmed   Block: 28   Gas used: 36785 (0.31%)

Transaction sent: 0x26eac7244e853608383c736e1dda20c96b49db87ae2a78d8fcd89f9499d50012
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  Miner.re_mine confirmed   Block: 30   Gas used: 50195 (0.42%)

tests/test_miner.py::test_mine PASSED
```

```
tests/test_miner.py::test_claim_rewards RUNNING
Transaction sent: 0xd2fbecd713b1c84479dc09f36a8a98d558890e378e9cbfd93df75fb552d8c976
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  Bank.constructor confirmed   Block: 31   Gas used: 262651 (2.19%)
  Bank deployed at: 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d

Transaction sent: 0x85e52fbf98a50817d94e07c674a52db033cf69745904230f80998dfc50308e7e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  Miner.constructor confirmed   Block: 32   Gas used: 1638074 (13.65%)
  Miner deployed at: 0x26f15335BB1C6a4C0B660eDd694a0555A9F1cce3

Transaction sent: 0xe5bd58f3ac07102ee3ed68e677bbd97e340bc1e5dc97ede1c9dd9ef0b3079143
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  Miner.mine confirmed   Block: 33   Gas used: 241319 (2.01%)

Transaction sent: 0xc609acb20f9b1b95e426c6a021c26da890d13b5fd7773a56d828591de21f13d6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  Miner.claimRewards confirmed (No user account detected.)   Block: 34   Gas used: 26095 (0.22%)

Transaction sent: 0x314d61cf9626594edb72adbe6fc71d0b8c1ed19523411f2462a5b14698d16fbd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  Miner.claimRewards confirmed (Only admin can call this function.)   Block: 35   Gas used: 35064 (0.29%)

Transaction sent: 0xfabece6c9115e8d3816b341387c12b102a6562c8bbf990ecdd107716b14f639d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  Bank.setAdmin confirmed   Block: 36   Gas used: 43421 (0.36%)

Transaction sent: 0x9d6893ba5519da2ace7e283a0eb2735ddcc5778ca7365581d278759ba3b3ff13
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  Miner.claimRewards confirmed   Block: 37   Gas used: 103934 (0.87%)

Transaction sent: 0x609e75529a8a9f72d4e053bb4d4e271aa097ad9b7b861821f9e0ed9e11651afc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  Miner.claimRewards confirmed   Block: 39   Gas used: 62908 (0.52%)

Transaction sent: 0x34a18be87ffc9896ee39572e24ad0c9e0683094f27641ae6f344bff0507e6d21
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  Miner.mine confirmed   Block: 40   Gas used: 309361 (2.58%)

Transaction sent: 0x8aa2eb2579fb6ac4245b1c19e8d9dd12dd36aac932cb976497e4a5342a35d04b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  Miner.claimRewards confirmed   Block: 41   Gas used: 129317 (1.08%)

Transaction sent: 0x5314c6fcc56e35e2784568314ff97f0aad6af77324c3514e2b813ef948e85b94
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  Miner.claimRewards confirmed   Block: 43   Gas used: 73301 (0.61%)

tests/test_miner.py::test_claim_rewards PASSED
```

```
tests/test_miner.py::test_withdraw RUNNING
Transaction sent: 0x9c3b84a2f000f5943db32c4e0683076b533caaea9594d1871ae943236e64540e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 18
  Bank.constructor confirmed   Block: 44   Gas used: 262651 (2.19%)
  Bank deployed at: 0xed00238F9A0F7b4d93842033cdF56cCB32C781c2

Transaction sent: 0x9cffd1c1f16cb932f2de62ab4eb1f4c8bba950a65b6494b29c945616582ad2ca
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 19
  Miner.constructor confirmed   Block: 45   Gas used: 1638062 (13.65%)
  Miner deployed at: 0xDae02e4fE488952cFB8c95177154D188647a0146

Transaction sent: 0x93ba84332c42990b5e45c2727b80beb85b496ac878da848d327ce3a7da16fba6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  Miner.mine confirmed   Block: 46   Gas used: 241319 (2.01%)

Transaction sent: 0xf0342a9c5b7397bacb599feaf7677db17809c6bcc6e107792c7bcb945c7357ee
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  Miner.mine confirmed   Block: 47   Gas used: 309361 (2.58%)

Transaction sent: 0xa4e8595e8599e30b43fb389e6b4763ae8dc06cb7983403d1df7d520dc46cc57d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  Miner.withdraw confirmed (No user account detected.)   Block: 48   Gas used: 28719 (0.24%)

Transaction sent: 0x659d3a183e9ac1106e9c9d9ad53c737478785fbec549cf369a49137f7552d791
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  Miner.withdraw confirmed (Only admin can call this function.)   Block: 49   Gas used: 37871 (0.32%)

Transaction sent: 0x1eaa59a129e3f66e0dbcd072a801fe5abe125bc8c374a187e84710f9ad3e154e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 20
  Bank.setAdmin confirmed   Block: 50   Gas used: 43421 (0.36%)

Transaction sent: 0x3ab9e613db0af1ee552578ee2070538bdfe0d3329fbc2fbf9cf1808d45d4c003
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  Miner.withdraw confirmed   Block: 51   Gas used: 66704 (0.56%)

Transaction sent: 0x5b797045dbf9d1f936ae75764359d0296fff0c6bc7ae4453bbcd554bd7100172
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  Miner.withdraw confirmed   Block: 53   Gas used: 52074 (0.43%)

Transaction sent: 0xc3e3d46769eeaaa5dd69caa732cf0a9f951a1142b466c91aa64c3d9c58c2951
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Miner.mine confirmed   Block: 54   Gas used: 210772 (1.76%)

Transaction sent: 0xa3d15fd0d7c3d48e9ff33031cf2e9eb18469d132cd594eb061206c871358732d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  Miner.withdraw confirmed   Block: 55   Gas used: 58357 (0.49%)

Transaction sent: 0xb0699459704dd9634fec9ecc7dabb2b5b3cae5422e53d8265f23ca6f6fec34d0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  Miner.withdraw confirmed   Block: 57   Gas used: 58247 (0.49%)

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

Annexes

Testing code:

`Miner.py:`

```
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_bank,
    deploy_miner
)


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

```
    t_wallet = get_account(7)
    p_wallet = get_account(8)

    bank = deploy_bank(owner)
    miner = deploy_miner(owner, 1, 1, 1, 1, 1,
                         bank.address, t_wallet, p_wallet)

    with reverts("Only owner can call this function."):
        miner.changeTransactionFee(2, {"from": other})
    with reverts("Transaction fee cannot exceed 5%"):
        miner.changeTransactionFee(6, {"from": owner})

    assert miner.transaction_fee() == 1
    miner.changeTransactionFee(2, {"from": owner})
    assert miner.transaction_fee() == 2


    with reverts("Only owner can call this function."):
        miner.changeReferralFee(2, {"from": other})
    with reverts("Referral fee cannot exceed 5%"):
        miner.changeReferralFee(6, {"from": owner})

    assert miner.referral_fee() == 1
    miner.changeReferralFee(2, {"from": owner})
    assert miner.referral_fee() == 2

    with reverts("Only owner can call this function."):
```

```
        miner.changePenaltyFee(2, {"from": other})
    with reverts("Penalty fee cannot exceed 50%"):
        miner.changePenaltyFee(55, {"from": owner})

    assert miner.penalty_fee() == 1
    miner.changePenaltyFee(25, {"from": owner})
    assert miner.penalty_fee() == 25

def test_mine(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    referee = get_account(3)

    t_wallet = get_account(7)
    p_wallet = get_account(8)

    bank = deploy_bank(owner)
    miner = deploy_miner(owner, 1, 1, 1, 1, 1,
                         bank.address, t_wallet, p_wallet)

    with reverts("Insufficent deposit amount."):
        miner.mine(ZERO_ADDRESS, {"from": other, "value": 1e15})

    assert t_wallet.balance() == 1000e18
    value = 1e18
    t_fee = (value * 0.01)
```

```
    tx = miner.mine(ZERO_ADDRESS, {"from": other, "value": value})
    assert tx.events['User_Created'][0]['user'] == other
    assert tx.events['Mine'][0]['user'] == other
    assert tx.events['Mine'][0]['amount'] == 1e18 - t_fee
    assert t_wallet.balance() == 1000e18 + t_fee

    tx = miner.mine(ZERO_ADDRESS, {"from": other, "value": value})
    assert tx.events['Mine'][0]['user'] == other
    assert tx.events['Mine'][0]['amount'] == 1e18 - t_fee

    tx = miner.mine(referee, {"from": extra, "value": value})
    assert tx.events['Mine'][0]['user'] == extra
    assert tx.events['Mine'][0]['amount'] == 1e18 - t_fee

    with reverts("No user account detected."):
        miner.re_mine({"from": referee})

    tx = miner.re_mine({"from": extra})
    assert tx.events['ReMine'][0]['user'] == extra
    increase_timestamp(DAY_TIMESTAMP * 10)
    tx = miner.re_mine({"from": extra})
    assert tx.events['ReMine'][0]['user'] == extra

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

```
    another = get_account(3)
    referee = get_account(4)

    t_wallet = get_account(7)
    p_wallet = get_account(8)

    bank = deploy_bank(owner)
    miner = deploy_miner(owner, 1, 1, 1, 1, 1,
                         bank.address, t_wallet, p_wallet)

    miner.mine(ZERO_ADDRESS, {"from": other, "value": 1e18})
    with reverts("No user account detected."):
        miner.claimRewards(10e18, {"from": extra})

    with reverts("Only admin can call this function."):
        miner.claimRewards(10e18, {"from": other})
    bank.setAdmin(miner.address, {"from": owner})

    tx = miner.claimRewards(1e16, {"from": other})
    assert tx.events['Claim'][0]['user'] == other
    assert tx.events['Claim'][0]['amount'] == 0.0099e18
    increase_timestamp(DAY_TIMESTAMP * 2)
    tx = miner.claimRewards(1e16, {"from": other})
    assert tx.events['Claim'][0]['user'] == other
    assert tx.events['Claim'][0]['amount'] == 0.0099e18

    miner.mine(referee, {"from": another, "value": 1e18})
    tx = miner.claimRewards(1e16, {"from": another})
```

```
    assert tx.events['Claim'][0]['user'] == another
    assert tx.events['Claim'][0]['amount'] == 0.0098e18

    increase_timestamp(DAY_TIMESTAMP * 2)
    tx = miner.claimRewards(1e16, {"from": another})
    assert tx.events['Claim'][0]['user'] == another
    assert tx.events['Claim'][0]['amount'] == 0.0098e18

def test_withdraw(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    another = get_account(3)
    referee = get_account(4)

    t_wallet = get_account(7)
    p_wallet = get_account(8)

    bank = deploy_bank(owner)
    miner = deploy_miner(owner, 1, 1, 1, 1, 1,
                         bank.address, t_wallet, p_wallet)

    miner.mine(ZERO_ADDRESS, {"from": other, "value": 1e18})
    miner.mine(referee, {"from": another, "value": 2e18})

    with reverts("No user account detected."):
        miner.withdraw(10e18, {"from": extra})
```

```
    with reverts("Only admin can call this function."):
        miner.withdraw(10e18, {"from": other})
    bank.setAdmin(miner.address, {"from": owner})

    miner.withdraw(0, {"from": other})
    increase_timestamp(DAY_TIMESTAMP * 2)
    tx = miner.withdraw(0, {"from": other})
    assert tx.events['Withdraw'][0]['user'] == other
    assert tx.events['Withdraw'][0]['amount'] == 0

    miner.mine(ZERO_ADDRESS, {"from": referee, "value": 10e18})
    miner.withdraw(0, {"from": another})
    increase_timestamp(DAY_TIMESTAMP * 2)
    tx = miner.withdraw(0, {"from": another})
    assert tx.events['Withdraw'][0]['user'] == another
    assert tx.events['Withdraw'][0]['amount'] == 0
```

Bank.py:

```
from brownie import (
    reverts,
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    get_account,
)
```

```
from scripts.deploy import (
    deploy_bank
)

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

    bank = deploy_bank(owner)

    with reverts("Only owner can call this function."):
        bank.setAdmin(extra, {"from": other})

    assert bank.admin() == ZERO_ADDRESS
    bank.setAdmin(extra, {"from": owner})
    assert bank.admin() == extra

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

    bank = deploy_bank(owner)
```

```
    with reverts("Only admin can call this function."):
        bank.transfer(extra, 1e18, {"from": other})
    bank.setAdmin(other, {"from": owner})
    with reverts("Insufficent funds."):
        bank.transfer(extra, 1e18, {"from": other})
    owner.transfer(bank.address, "2 ether")
    tx = bank.transfer(extra, 1e18, {"from": other})
    assert tx.events['Transfer'][0]['user'] == extra
    assert tx.events['Transfer'][0]['amount'] == 1e18
```

Technical Findings Summary

Findings

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

Assessment Results

Score Results

Review	Score
Audit Score	**90/100**
Assure KYC	Pending
Audit Score	95/100

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 Profit_IQ project, We regret to inform you that the project initially did not meet the required security standards due to identified vulnerabilities within the contract functions. However, the development team have successfully addressed and resolved all issues, including those classified as medium and high risk, ensuring the smart contacts are secure and ready for deployment.

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.