Assure DeFi®

THE VERIFICATION GOLD STANDARD

Security Assessment

Mullet

Date: 23/05/2024

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🟡 Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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

Insecure	Poorly Secured	**Secured**	Well Secured

Scope

Target Code And Revision

For this audit, we performed research, investigation, and review of the Mullet 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/moontography/mullet/blob/main/contracts/Mullet.sol File: mullet-main.zip [SHA256 Hash]: da34959452c8b09272665b5a9904edc0ecb032096b3e99286da20eecb37d9c4f
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

Contract: GameTheoryRewards

Function: paybackLastClaim()

Issue: There is a scenario where a user can call claim() and then immediately call paybackLastClaim(). If _removeShares() calls _depositRewards() and this function is called again after _removeShares(), it triggers the require(_amount > 0 && totalShares > 0, 'D0'); statement, causing a revert. This results in the user being unable to execute paybackLastClaim() due to the lack of available totalShares.

Solution: Review and adjust the internal logic of the function.

LOW

No low severity issues were found.

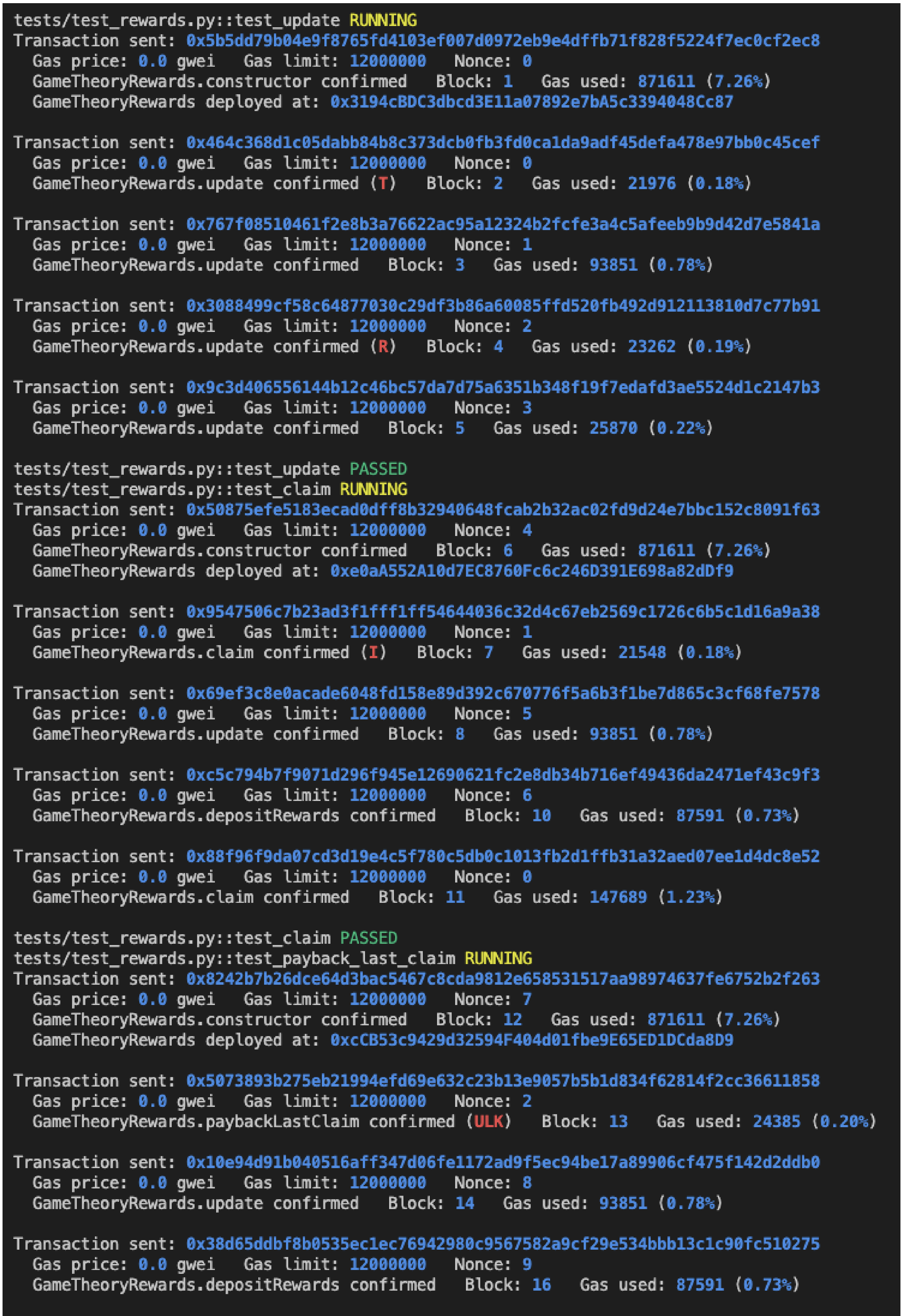
INFORMATIONAL

No Informational severity issues were found.

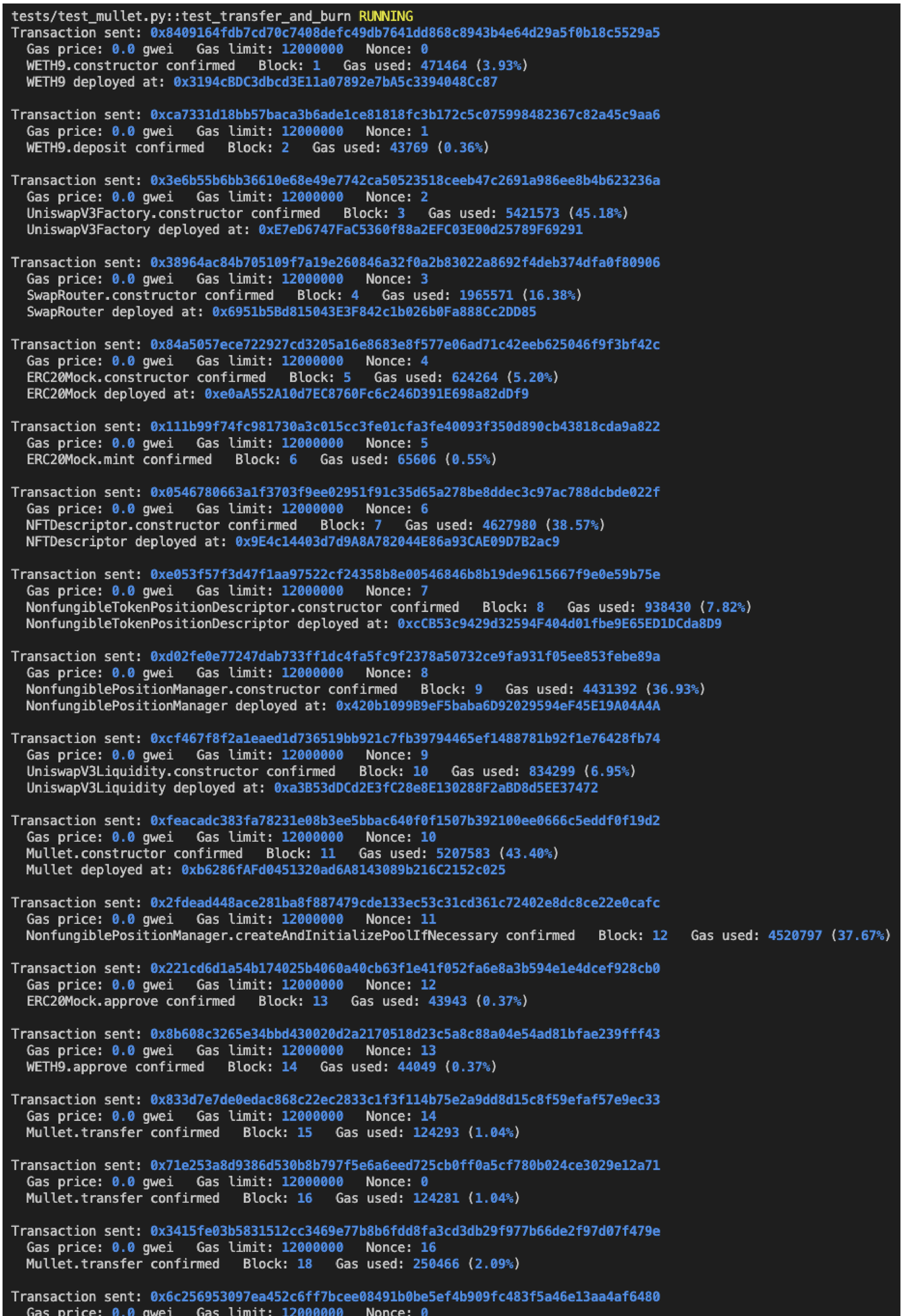
Testing coverage

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

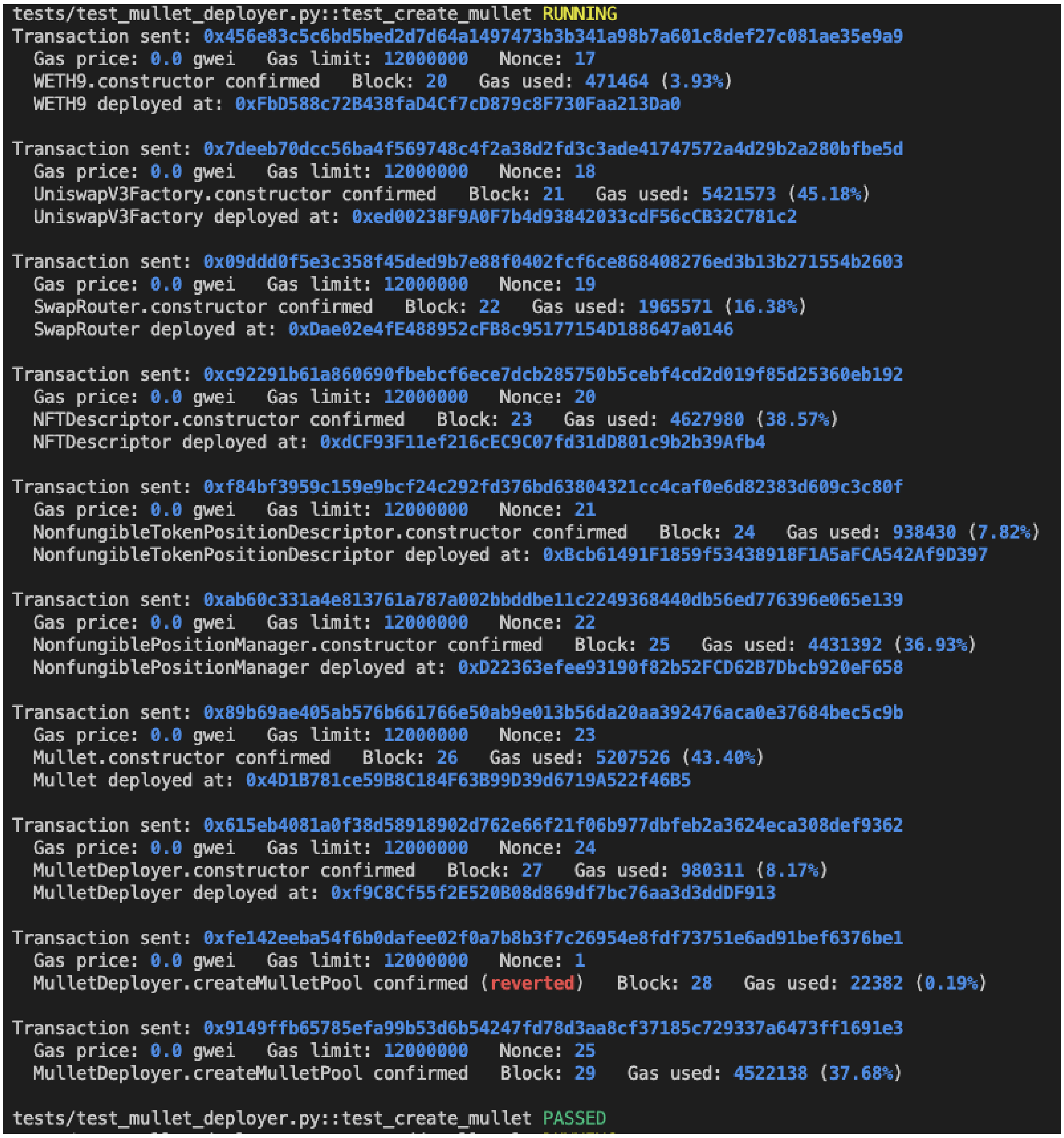
Mullet contracts tests:



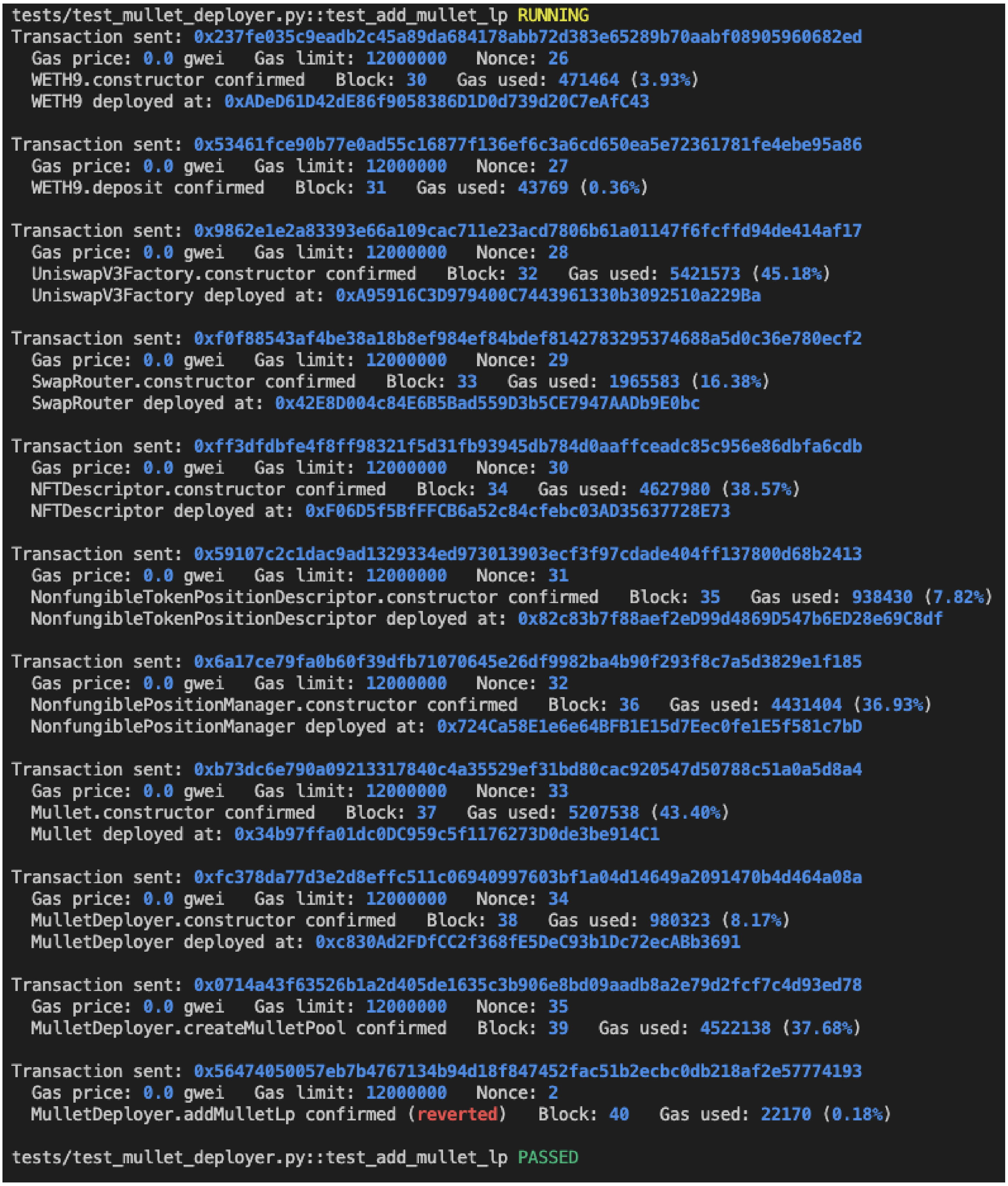
```
tests/test_rewards.py::test_update RUNNING
Transaction sent: 0x5b5dd79b04e9f8765fd4103ef007d0972eb9e4dffb71f828f5224f7ec0cf2ec8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  GameTheoryRewards.constructor confirmed   Block: 1   Gas used: 871611 (7.26%)
  GameTheoryRewards deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x464c368d1c05dabb84b8c373dcb0fb3fd0ca1da9adf45defa478e97bb0c45cef
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  GameTheoryRewards.update confirmed (T)   Block: 2   Gas used: 21976 (0.18%)

Transaction sent: 0x767f08510461f2e8b3a76622ac95a12324b2fcfe3a4c5afeeb9b9d42d7e5841a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  GameTheoryRewards.update confirmed   Block: 3   Gas used: 93851 (0.78%)

Transaction sent: 0x3088499cf58c64877030c29df3b86a60085ffd520fb492d912113810d7c77b91
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  GameTheoryRewards.update confirmed (R)   Block: 4   Gas used: 23262 (0.19%)

Transaction sent: 0x9c3d406556144b12c46bc57da7d75a6351b348f19f7edafd3ae5524d1c2147b3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  GameTheoryRewards.update confirmed   Block: 5   Gas used: 25870 (0.22%)

tests/test_rewards.py::test_update PASSED
tests/test_rewards.py::test_claim RUNNING
Transaction sent: 0x50875efe5183ecad0dff8b32940648fcab2b32ac02fd9d24e7bbc152c8091f63
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  GameTheoryRewards.constructor confirmed   Block: 6   Gas used: 871611 (7.26%)
  GameTheoryRewards deployed at: 0xe0aA552A10d7EC8760Fc6c246D391E698a82dDf9

Transaction sent: 0x9547506c7b23ad3f1fff1ff54644036c32d4c67eb2569c1726c6b5c1d16a9a38
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  GameTheoryRewards.claim confirmed (I)   Block: 7   Gas used: 21548 (0.18%)

Transaction sent: 0x69ef3c8e0acade6048fd158e89d392c670776f5a6b3f1be7d865c3cf68fe7578
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  GameTheoryRewards.update confirmed   Block: 8   Gas used: 93851 (0.78%)

Transaction sent: 0xc5c794b7f9071d296f945e12690621fc2e8db34b716ef49436da2471ef43c9f3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  GameTheoryRewards.depositRewards confirmed   Block: 10   Gas used: 87591 (0.73%)

Transaction sent: 0x88f96f9da07cd3d19e4c5f780c5db0c1013fb2d1ffb31a32aed07ee1d4dc8e52
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  GameTheoryRewards.claim confirmed   Block: 11   Gas used: 147689 (1.23%)

tests/test_rewards.py::test_claim PASSED
tests/test_rewards.py::test_payback_last_claim RUNNING
Transaction sent: 0x8242b7b26dce64d3bac5467c8cda9812e658531517aa98974637fe6752b2f263
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  GameTheoryRewards.constructor confirmed   Block: 12   Gas used: 871611 (7.26%)
  GameTheoryRewards deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0x5073893b275eb21994efd69e632c23b13e9057b5b1d834f62814f2cc36611858
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  GameTheoryRewards.paybackLastClaim confirmed (ULK)   Block: 13   Gas used: 24385 (0.20%)

Transaction sent: 0x10e94d91b040516aff347d06fe1172ad9f5ec94be17a89906cf475f142d2ddb0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  GameTheoryRewards.update confirmed   Block: 14   Gas used: 93851 (0.78%)

Transaction sent: 0x38d65ddbf8b0535ec1ec76942980c9567582a9cf29e534bbb13c1c90fc510275
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  GameTheoryRewards.depositRewards confirmed   Block: 16   Gas used: 87591 (0.73%)
```



```
tests/test_mullet.py::test_transfer_and_burn RUNNING
Transaction sent: 0x8409164fdb7cd70c7408defc49db7641dd868c8943b4e64d29a5f0b18c5529a5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  WETH9.constructor confirmed   Block: 1   Gas used: 471464 (3.93%)
  WETH9 deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0xca7331d18bb57baca3b6ade1ce81818fc3b172c5c075998482367c82a45c9aa6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  WETH9.deposit confirmed   Block: 2   Gas used: 43769 (0.36%)

Transaction sent: 0x3e6b55b6bb36610e68e49e7742ca50523518ceeb47c2691a986ee8b4b623236a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  UniswapV3Factory.constructor confirmed   Block: 3   Gas used: 5421573 (45.18%)
  UniswapV3Factory deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0x38964ac84b705109f7a19e260846a32f0a2b83022a8692f4deb374dfa0f80906
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  SwapRouter.constructor confirmed   Block: 4   Gas used: 1965571 (16.38%)
  SwapRouter deployed at: 0x6951b5Bd815043E3F842c1b026b0Fa888Cc2DD85

Transaction sent: 0x84a5057ece722927cd3205a16e8683e8f577e06ad71c42eeb625046f9f3bf42c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ERC20Mock.constructor confirmed   Block: 5   Gas used: 624264 (5.20%)
  ERC20Mock deployed at: 0xe0aA552A10d7EC8760Fc6c246D391E698a82dDf9

Transaction sent: 0x111b99f74fc981730a3c015cc3fe01cfa3fe40093f350d890cb43818cda9a822
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  ERC20Mock.mint confirmed   Block: 6   Gas used: 65606 (0.55%)

Transaction sent: 0x0546780663a1f3703f9ee02951f91c35d65a278be8ddec3c97ac788dcbde022f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  NFTDescriptor.constructor confirmed   Block: 7   Gas used: 4627980 (38.57%)
  NFTDescriptor deployed at: 0x9E4c14403d7d9A8A782044E86a93CAE09D7B2ac9

Transaction sent: 0xe053f57f3d47f1aa97522cf24358b8e00546846b8b19de9615667f9e0e59b75e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  NonfungibleTokenPositionDescriptor.constructor confirmed   Block: 8   Gas used: 938430 (7.82%)
  NonfungibleTokenPositionDescriptor deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0xd02fe0e77247dab733ff1dc4fa5fc9f2378a50732ce9fa931f05ee853febe89a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  NonfungiblePositionManager.constructor confirmed   Block: 9   Gas used: 4431392 (36.93%)
  NonfungiblePositionManager deployed at: 0x420b1099B9eF5baba6D92029594eF45E19A04A4A

Transaction sent: 0xcf467f8f2a1eaed1d736519bb921c7fb39794465ef1488781b92f1e76428fb74
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  UniswapV3Liquidity.constructor confirmed   Block: 10   Gas used: 834299 (6.95%)
  UniswapV3Liquidity deployed at: 0xa3B53dDCd2E3fC28e8E130288F2aBD8d5EE37472

Transaction sent: 0xfeacadc383fa78231e08b3ee5bbac640f0f1507b392100ee0666c5eddf0f19d2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  Mullet.constructor confirmed   Block: 11   Gas used: 5207583 (43.40%)
  Mullet deployed at: 0xb6286fAFd0451320ad6A8143089b216C2152c025

Transaction sent: 0x2fdead448ace281ba8f887479cde133ec53c31cd361c72402e8dc8ce22e0cafc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  NonfungiblePositionManager.createAndInitializePoolIfNecessary confirmed   Block: 12   Gas used: 4520797 (37.67%)

Transaction sent: 0x221cd6d1a54b174025b4060a40cb63f1e41f052fa6e8a3b594e1e4dcef928cb0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  ERC20Mock.approve confirmed   Block: 13   Gas used: 43943 (0.37%)

Transaction sent: 0x8b608c3265e34bbd430020d2a2170518d23c5a8c88a04e54ad81bfae239fff43
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  WETH9.approve confirmed   Block: 14   Gas used: 44049 (0.37%)

Transaction sent: 0x833d7e7de0edac868c22ec2833c1f3f114b75e2a9dd8d15c8f59efaf57e9ec33
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  Mullet.transfer confirmed   Block: 15   Gas used: 124293 (1.04%)

Transaction sent: 0x71e253a8d9386d530b8b797f5e6a6eed725cb0ff0a5cf780b024ce3029e12a71
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Mullet.transfer confirmed   Block: 16   Gas used: 124281 (1.04%)

Transaction sent: 0x3415fe03b5831512cc3469e77b8b6fdd8fa3cd3db29f977b66de2f97d07f479e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  Mullet.transfer confirmed   Block: 18   Gas used: 250466 (2.09%)

Transaction sent: 0x6c256953097ea452c6ff7bcee08491b0be5ef4b909fc483f5a46e13aa4af6480
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
```



```
tests/test_mullet_deployer.py::test_create_mullet RUNNING
Transaction sent: 0x456e83c5c6bd5bed2d7d64a1497473b3b341a98b7a601c8def27c081ae35e9a9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  WETH9.constructor confirmed   Block: 20   Gas used: 471464 (3.93%)
  WETH9 deployed at: 0xFbD588c72B438faD4Cf7cD879c8F730Faa213Da0

Transaction sent: 0x7deeb70dcc56ba4f569748c4f2a38d2fd3c3ade41747572a4d29b2a280bfbe5d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 18
  UniswapV3Factory.constructor confirmed   Block: 21   Gas used: 5421573 (45.18%)
  UniswapV3Factory deployed at: 0xed00238F9A0F7b4d93842033cdF56cCB32C781c2

Transaction sent: 0x09ddd0f5e3c358f45ded9b7e88f0402fcf6ce868408276ed3b13b271554b2603
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 19
  SwapRouter.constructor confirmed   Block: 22   Gas used: 1965571 (16.38%)
  SwapRouter deployed at: 0xDae02e4fE488952cFB8c95177154D188647a0146

Transaction sent: 0xc92291b61a860690fbebcf6ece7dcb285750b5cebf4cd2d019f85d25360eb192
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 20
  NFTDescriptor.constructor confirmed   Block: 23   Gas used: 4627980 (38.57%)
  NFTDescriptor deployed at: 0xdCF93F11ef216cEC9C07fd31dD801c9b2b39Afb4

Transaction sent: 0xf84bf3959c159e9bcf24c292fd376bd63804321cc4caf0e6d82383d609c3c80f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 21
  NonfungibleTokenPositionDescriptor.constructor confirmed   Block: 24   Gas used: 938430 (7.82%)
  NonfungibleTokenPositionDescriptor deployed at: 0xBcb61491F1859f53438918F1A5aFCA542Af9D397

Transaction sent: 0xab60c331a4e813761a787a002bbddbe11c2249368440db56ed776396e065e139
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 22
  NonfungiblePositionManager.constructor confirmed   Block: 25   Gas used: 4431392 (36.93%)
  NonfungiblePositionManager deployed at: 0xD22363efee93190f82b52FCD62B7Dbcb920eF658

Transaction sent: 0x89b69ae405ab576b661766e50ab9e013b56da20aa392476aca0e37684bec5c9b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 23
  Mullet.constructor confirmed   Block: 26   Gas used: 5207526 (43.40%)
  Mullet deployed at: 0x4D1B781ce59B8C184F63B99D39d6719A522f46B5

Transaction sent: 0x615eb4081a0f38d58918902d762e66f21f06b977dbfeb2a3624eca308def9362
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  MulletDeployer.constructor confirmed   Block: 27   Gas used: 980311 (8.17%)
  MulletDeployer deployed at: 0xf9C8Cf55f2E520B08d869df7bc76aa3d3ddDF913

Transaction sent: 0xfe142eeba54f6b0dafee02f0a7b8b3f7c26954e8fdf73751e6ad91bef6376be1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  MulletDeployer.createMulletPool confirmed (reverted)   Block: 28   Gas used: 22382 (0.19%)

Transaction sent: 0x9149ffb65785efa99b53d6b54247fd78d3aa8cf37185c729337a6473ff1691e3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  MulletDeployer.createMulletPool confirmed   Block: 29   Gas used: 4522138 (37.68%)

tests/test_mullet_deployer.py::test_create_mullet PASSED
```



```
tests/test_mullet_deployer.py::test_add_mullet_lp RUNNING
Transaction sent: 0x237fe035c9eadb2c45a89da684178abb72d383e65289b70aabf08905960682ed
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 26
  WETH9.constructor confirmed   Block: 30   Gas used: 471464 (3.93%)
  WETH9 deployed at: 0xADeD61D42dE86f9058386D1D0d739d20C7eAfC43

Transaction sent: 0x53461fce90b77e0ad55c16877f136ef6c3a6cd650ea5e72361781fe4ebe95a86
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 27
  WETH9.deposit confirmed   Block: 31   Gas used: 43769 (0.36%)

Transaction sent: 0x9862e1e2a83393e66a109cac711e23acd7806b61a01147f6fcffd94de414af17
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 28
  UniswapV3Factory.constructor confirmed   Block: 32   Gas used: 5421573 (45.18%)
  UniswapV3Factory deployed at: 0xA95916C3D979400C7443961330b3092510a229Ba

Transaction sent: 0xf0f88543af4be38a18b8ef984ef84bdef8142783295374688a5d0c36e780ecf2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 29
  SwapRouter.constructor confirmed   Block: 33   Gas used: 1965583 (16.38%)
  SwapRouter deployed at: 0x42E8D004c84E6B5Bad559D3b5CE7947AADb9E0bc

Transaction sent: 0xff3dfdbfe4f8ff98321f5d31fb93945db784d0aaffceadc85c956e86dbfa6cdb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 30
  NFTDescriptor.constructor confirmed   Block: 34   Gas used: 4627980 (38.57%)
  NFTDescriptor deployed at: 0xF06D5f5BfFFCB6a52c84cfebc03AD35637728E73

Transaction sent: 0x59107c2c1dac9ad1329334ed973013903ecf3f97cdade404ff137800d68b2413
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 31
  NonfungibleTokenPositionDescriptor.constructor confirmed   Block: 35   Gas used: 938430 (7.82%)
  NonfungibleTokenPositionDescriptor deployed at: 0x82c83b7f88aef2eD99d4869D547b6ED28e69C8df

Transaction sent: 0x6a17ce79fa0b60f39dfb71070645e26df9982ba4b90f293f8c7a5d3829e1f185
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 32
  NonfungiblePositionManager.constructor confirmed   Block: 36   Gas used: 4431404 (36.93%)
  NonfungiblePositionManager deployed at: 0x724Ca58E1e6e64BFB1E15d7Eec0fe1E5f581c7bD

Transaction sent: 0xb73dc6e790a09213317840c4a35529ef31bd80cac920547d50788c51a0a5d8a4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 33
  Mullet.constructor confirmed   Block: 37   Gas used: 5207538 (43.40%)
  Mullet deployed at: 0x34b97ffa01dc0DC959c5f1176273D0de3be914C1

Transaction sent: 0xfc378da77d3e2d8effc511c06940997603bf1a04d14649a2091470b4d464a08a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 34
  MulletDeployer.constructor confirmed   Block: 38   Gas used: 980323 (8.17%)
  MulletDeployer deployed at: 0xc830Ad2FDfCC2f368fE5DeC93b1Dc72ecABb3691

Transaction sent: 0x0714a43f63526b1a2d405de1635c3b906e8bd09aadb8a2e79d2fcf7c4d93ed78
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 35
  MulletDeployer.createMulletPool confirmed   Block: 39   Gas used: 4522138 (37.68%)

Transaction sent: 0x56474050057eb7b4767134b94d18f847452fac51b2ecbc0db218af2e57774193
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  MulletDeployer.addMulletLp confirmed (reverted)   Block: 40   Gas used: 22170 (0.18%)

tests/test_mullet_deployer.py::test_add_mullet_lp PASSED
```

Annexes

Testing code:

Testing_Mullet.py:

```python
from brownie import (

    reverts,

)


from scripts.helpful_scripts import (

    ZERO_ADDRESS,

    get_account,

    calculate_sqrt_price_x96

)


from scripts.deploy import (

    deploy_erc,

    deploy_mullet,

    # dependencies

    deploy_weth,

    deploy_uniswap_v3_factory,

    deploy_swap_router,

    deploy_position_manager,

    deploy_position_descriptor,

    deploy_uniswap_v3_liquidity

)


def test_transfer_and_burn(only_local):

    # Arrange
    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)

    random_endpoint = get_account(3)


    # Deployment weth, router and factory

    weth = deploy_weth(owner)

    weth.deposit({"from": owner, "value": 10e18})

    factory = deploy_uniswap_v3_factory(owner)

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


    # Deploy mock token

    liquidity_token = deploy_erc(owner, "Liquidity", "LIQ")

    liquidity_token.mint(owner, 100000e18)


    # Deploy PositionManager

    label = 0x4c4142454c

    position_descriptor = deploy_position_descriptor(owner, weth.address, label)

    position_manager = deploy_position_manager(owner, factory.address, weth.address,
position_descriptor.address)


    # Uniswap V3 Liquidity

    liquidity = deploy_uniswap_v3_liquidity(owner, weth.address, liquidity_token.address,
position_manager.address)


    mullet = deploy_mullet(

        owner, weth.address, position_manager.address,

        factory.address, router.address, random_endpoint

    )
```

```python
    # Create and initialize pool

    fee = 500

    sqrt_price_X96 = calculate_sqrt_price_x96(token0 = 1, token1= 10000) # 10000 LIQ - 1
WETH

    tx = position_manager.createAndInitializePoolIfNecessary(

        weth.address, liquidity_token.address,

        fee, sqrt_price_X96, {"from": owner})

    pool = tx.events['PoolCreated'][0]['pool']

    tick_spacing = tx.events['PoolCreated'][0]['tickSpacing']

    tick = tx.events['Initialize'][0]['tick']

    # Mint

    liquidity_token.approve(liquidity.address, 10000e18, {"from": owner})

    weth.approve(liquidity.address, 1e18, {"from": owner})


    tx = mullet.transfer(other, 5e18, {"from": owner})

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

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

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

    assert tx.events['RemoveShares'][0]['user'] == owner

    assert tx.events['RemoveShares'][0]['amount'] == 5e18

    assert tx.events['AddShares'][0]['user'] == other

    assert tx.events['AddShares'][0]['amount'] == 5e18


    tx = mullet.transfer(extra, 1e18, {"from": other})

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

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

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

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

    assert tx.events['AddShares'][0]['user'] == extra

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


    owner.transfer(mullet, "1 ether")


    tx = mullet.transfer(other, 1e18, {"from": owner})

    assert tx.events['DepositRewards'][0]['user'] == mullet.address

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


    tx = mullet.burn(1e18, {"from": extra})

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

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

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

    assert tx.events['RemoveShares'][0]['user'] == extra

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

Test rewards:

```
from brownie import (

    reverts,

)


from scripts.helpful_scripts import (

    ZERO_ADDRESS,

    DAY_TIMESTAMP,

    get_account,

    increase_timestamp
```

```python
)


from scripts.deploy import (

    deploy_rewards

)


def test_update(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    extra = get_account(2)


    rewards = deploy_rewards(owner, DAY_TIMESTAMP, 3 * DAY_TIMESTAMP)

    with reverts("T"):

        rewards.update(ZERO_ADDRESS, 1e18, False, {"from": other})

    tx = rewards.update(extra, 1e18, False, {"from": owner})

    assert tx.events['AddShares'][0]['user'] == extra

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


    with reverts("R"):

        rewards.update(extra, 2e18, True, {"from": owner})


    tx = rewards.update(extra, 1e18, True, {"from": owner})

    assert tx.events['RemoveShares'][0]['user'] == extra

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


def test_claim(only_local):

    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    rewards = deploy_rewards(owner, DAY_TIMESTAMP, 3 * DAY_TIMESTAMP)
    with reverts("I"):
        rewards.claim({"from": other})
    rewards.update(extra, 1e18, False, {"from": owner})

    increase_timestamp(DAY_TIMESTAMP * 8)
    rewards.depositRewards({"from": owner, "value": 1e17})

    tx = rewards.claim({"from": extra})
    assert tx.events['DistributeReward'][0]['user'] == extra
    assert tx.events['DistributeReward'][0]['amount'] == 99999999999999999

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

    rewards = deploy_rewards(owner, DAY_TIMESTAMP, 3 * DAY_TIMESTAMP)
    with reverts("ULK"):
        rewards.paybackLastClaim({"from": other})
    rewards.update(extra, 1e18, False, {"from": owner})
    increase_timestamp(DAY_TIMESTAMP * 8)
    rewards.depositRewards({"from": owner, "value": 1e17})
    rewards.claim({"from": extra})


    with reverts("E"):

        rewards.paybackLastClaim({"from": extra})

    with reverts(""):

        rewards.paybackLastClaim({"from": extra, "value": 999999999999999999})

    #assert tx.events['DepositRewards'][0]['user'] == extra

    #assert tx.events['DepositRewards'][0]['amountTokens'] == 1e17
```

Test mullet deployer:

```
from brownie import (

    reverts,

)


from scripts.helpful_scripts import (

    ZERO_ADDRESS,

    get_account,

    calculate_sqrt_price_x96

)


from scripts.deploy import (

    deploy_mullet,

    deploy_mullet_deployer,

    # dependencies

    deploy_weth,

    deploy_uniswap_v3_factory,

    deploy_swap_router,

    deploy_position_manager,
```

```python
    deploy_position_descriptor,
)


def test_create_mullet(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    random_endpont = get_account(2)

    # Deployment weth, router and factory
    weth = deploy_weth(owner)
    factory = deploy_uniswap_v3_factory(owner)
    router = deploy_swap_router(owner, factory.address, weth.address)

    # Deploy PositionManager
    label = 0x4c4142454c
    position_descriptor = deploy_position_descriptor(owner, weth.address, label)
    position_manager = deploy_position_manager(owner, factory.address, weth.address,
position_descriptor.address)

    mu = deploy_mullet(
        owner, weth.address, position_manager.address,
        factory.address, router.address, random_endpont
    )

    mullet = deploy_mullet_deployer(
        owner, weth.address, position_manager.address,
        factory.address, mu.address
    )


    sqrt_price_X96 = calculate_sqrt_price_x96(token0 = 1, token1= 10000)


    with reverts():

        mullet.createMulletPool(sqrt_price_X96, {"from": other})


    tx = mullet.createMulletPool(sqrt_price_X96, {"from": owner})

    assert tx.events['PoolCreated'] is not None

    assert tx.events['Initialize'] is not None


def test_add_mullet_lp(only_local):

    # Arrange

    owner = get_account(0)

    other = get_account(1)

    random_endpont = get_account(2)


    # Deployment weth, router and factory

    weth = deploy_weth(owner)

    weth.deposit({"from": owner, "value": 10e18})

    factory = deploy_uniswap_v3_factory(owner)

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


    # Deploy PositionManager

    label = 0x4c4142454c

    position_descriptor = deploy_position_descriptor(owner, weth.address, label)

    position_manager = deploy_position_manager(owner, factory.address, weth.address,
position_descriptor.address)
```

```python
mu = deploy_mullet(
    owner, weth.address, position_manager.address,
    factory.address, router.address, random_endpont
)

mullet = deploy_mullet_deployer(
    owner, weth.address, position_manager.address,
    factory.address, mu.address
)

sqrt_price_X96 = calculate_sqrt_price_x96(token0 = 1, token1= 10000)
mullet.createMulletPool(sqrt_price_X96, {"from": owner})
with reverts():
    mullet.addMulletLp({"from": other})
```

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
🔴 High	0					
🟠 Medium	1					
🟡 Low	0					
🟢 Informational	0					

Assessment Results

Score Results

Review	Score
Global Score	**85/100**
Assure KYC	Pending
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.

<u>Audit PASS</u>

Following our comprehensive security audit of the token contract for Mullet project, we inform you that the project has met the necessary security standards. In any case, we recommend fixing the medium vulnerability so that the contract has the expected behavior.

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.