Assure DeFi®

THE VERIFICATION GOLD STANDARD

Security Assessment

Drops

Date: 01/04/2024

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Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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 Drops 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/DropsERC/Drops/blob/main/contracts/Drops%20Token/Drops.sol Drops Token contract - [Commit] *21f148879b2f4766850895dc316ac6e6c27d60cc* *Deployed [Eth Network] - 0xa562912e1328eea987e04c2650efb5703757850c*
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

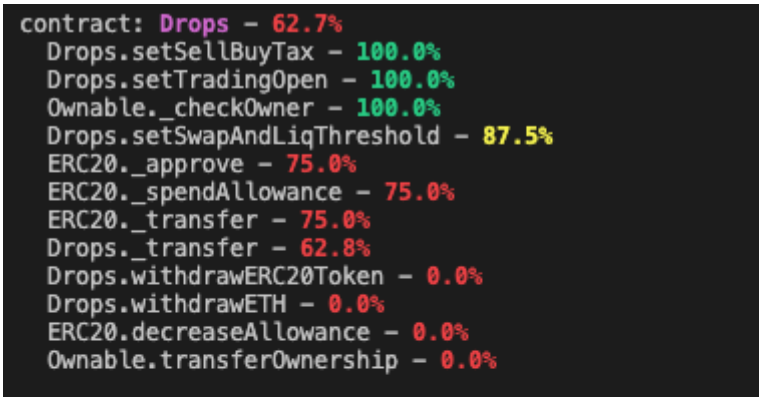
No Informational severity issues were found.

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.*

Drops contracts tests:

Coverages:



Testing Drops Token:

```
tests/test_drops.py::test_transfer RUNNING
Transaction sent: 0xc6f639b9e0e494a06751e98b9dee54bbff532b155eacd6b0ec69bf826ce2c654
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  UniswapV2Factory.constructor confirmed   Block: 1   Gas used: 2412730 (20.11%)
  UniswapV2Factory deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0xf0c4bfff822f620ae1e8103c528eff7aa5f9883de2121d54628d6992df45b4b0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  WETH9.constructor confirmed   Block: 2   Gas used: 476546 (3.97%)
  WETH9 deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0x31f542bf423b85ec64e51189ddb2e32012ead1f5145ba7726b5da371327c0721
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  UniswapV2Router02.constructor confirmed   Block: 3   Gas used: 3895430 (32.46%)
  UniswapV2Router02 deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0xad245bdf9782dbcbdfefcf91cf200225d30b9b3b03ba63d8b2f35bfed16aa94a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  Drops.constructor confirmed   Block: 4   Gas used: 4146930 (34.56%)
  Drops deployed at: 0x6951b5Bd815043E3F842c1b026b0Fa888Cc2DD85

Transaction sent: 0x34a6103a2c76298c83214bdf9461a853abaca4a1c45a5b149a94f28baf18e5d4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  Drops.transfer confirmed (Trading Closed)   Block: 5   Gas used: 24826 (0.21%)

Transaction sent: 0xd5cc7b3f1faf561b29ebce8184614a83b98e19053d5fa45f887d9921abfedf6b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Drops.setTradingOpen confirmed (Ownable: caller is not the owner)   Block: 6   Gas used: 22263 (0.19%)

Transaction sent: 0xc7bdef520f50fc2a213bf7c174c3bc708af492cad8a847918525c84cf0dbf09e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  Drops.setTradingOpen confirmed   Block: 7   Gas used: 44880 (0.37%)

Transaction sent: 0x736b558e97e6ab89e737533afc361344bff43d07859368aa94b8bd961ea9f328
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  Drops.setTradingOpen confirmed (Trading already open)   Block: 8   Gas used: 23101 (0.19%)

Transaction sent: 0x845b3ff4bf4eca33dd4616d82a802b3265c488f84c2ba24ed32a1e95963a31ca
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  Drops.transfer confirmed (Transfer amount must be greater than 0)   Block: 9   Gas used: 22105 (0.18%)

Transaction sent: 0x69296965cc0e801fd276624f7b2efe3b25e1077a7715c7f41f06f8441ffd0baf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  Drops.transfer confirmed   Block: 10   Gas used: 59942 (0.50%)

Transaction sent: 0xc5e526b623e36a11f194ad6d90cae2f5b38dbef4e05853a23efdb8f525f3220f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  Drops.transfer confirmed   Block: 11   Gas used: 87152 (0.73%)

Transaction sent: 0xb53b2d9d34ce4735f3c734bfcec55e95b8cbfb447e3c41478dda59281825d902
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  Drops.setSwapAndLiqThreshold confirmed (Ownable: caller is not the owner)   Block: 12   Gas used: 22539 (0.19%)

Transaction sent: 0x10c69286f10cf7d2c2e42995a433e66fc3ed6316ac47d3bdcabd4e719d59d897
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  Drops.setSwapAndLiqThreshold confirmed (SnL Threshold must be within the allowed range)   Block: 13   Gas used: 23671 (0.20%)

Transaction sent: 0x9c8b059b864e541c693797968cf376680775a2160144707dd617fbbb4bfb4d66
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  Drops.setSwapAndLiqThreshold confirmed   Block: 14   Gas used: 29652 (0.25%)
```

Annexes

Testing code:

`Drops_Marketplace.py:`

```
from brownie import (
    reverts,
)

from scripts.helpful_scripts import (
    get_account,
)

from scripts.deploy import (
    deploy_factory,
    deploy_router,
    deploy_weth,
    deploy_drops,
)

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

    factory = deploy_factory(owner, fee_wallet)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    # buySellTax
    # 1000  = 10%
    # 100 = 1%
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, fee_wallet)
    # Supply: 10000000e18

    with reverts("Trading Closed"):
        drops.transfer(other, 1e18, {"from": owner})
    with reverts("Ownable: caller is not the owner"):
        drops.setTradingOpen({"from": other})
    drops.setTradingOpen({"from": owner})
    with reverts("Trading already open"):
        drops.setTradingOpen({"from": owner})
    with reverts("Transfer amount must be greater than 0"):
        drops.transfer(other, 0, {"from": owner})
```

```
    tx = drops.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

    # BUY
    tx = drops.transfer(drops.uniswapV2Pair(), 1e18, {"from": other})
    assert tx.events['Transfer'][1]['from'] == other
    assert tx.events['Transfer'][1]['to'] == drops.uniswapV2Pair()
    assert tx.events['Transfer'][1]['value'] == 0.9e18

    with reverts("Ownable: caller is not the owner"):
        drops.setSwapAndLiqThreshold(1e18, {"from": other})
    with reverts("SnL Threshold must be within the allowed range"):
        drops.setSwapAndLiqThreshold(1e18, {"from": owner})
    drops.setSwapAndLiqThreshold(1e21, {"from": owner}) # 100 ethers

    # NO FEE
    tx = drops.transfer(drops.uniswapV2Pair(), 1e18, {"from": owner})
    assert tx.events['Transfer'][0]['from'] == owner
    assert tx.events['Transfer'][0]['to'] == drops.uniswapV2Pair()
    assert tx.events['Transfer'][0]['value'] == 1e18

    with reverts("Ownable: caller is not the owner"):
        drops.setSellBuyTax(10, 10, {"from": other})
    with reverts("You can only lower fees"):
        drops.setSellBuyTax(10, 10, {"from": owner})
    tx = drops.setSellBuyTax(5, 5, {"from": owner})
    assert tx.events['TaxUpdated'][0]['taxPercent'] == 1000
```

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
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 Drops project, we inform you that the project has met the necessary security standards and no issues were identified.

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.