Assure DeFi®

THE VERIFICATION GOLD STANDARD

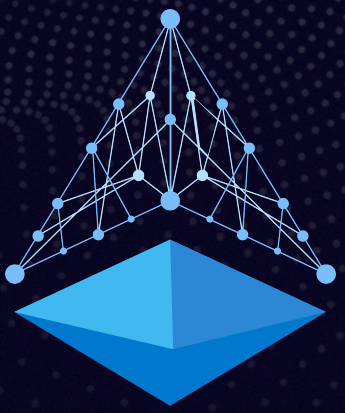
Security Assessment

EVA Token

Date: 11/09/2024

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Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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 EVA Token contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	EVA.sol - [SHA256] e102a180a55936ef67b7fa9842e345198e41df8186dccc55e901ce5795775f02
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.

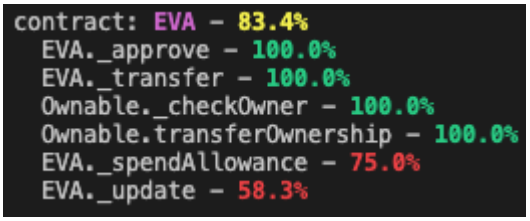
INFORMATIONAL

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.*

EVA Token contract tests:



```
tests/test_eva.py::test_transfer RUNNING
Transaction sent: 0xa58d53a083ac927bf3f418d3fc23c72543ff43c4804f35a9b3707a2774b07de8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  EVA.constructor confirmed   Block: 1   Gas used: 556372 (4.64%)
  EVA deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0xb1dba2d79c6c171e5763a48ed7a187751c1a104f811e9b1a4def8bdecd77dd87
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  EVA.transfer confirmed (reverted)   Block: 2   Gas used: 22144 (0.18%)

Transaction sent: 0xbd4806e98020104b4c709b9249cb97e3f45ac0b6e4535dfb3ac7449e3567108d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  EVA.transfer confirmed (reverted)   Block: 3   Gas used: 21951 (0.18%)

Transaction sent: 0x6e441aa311bc67e58fcd037913b97216de1d9e242562d8728e1fb2f5625071ca
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  EVA.transfer confirmed   Block: 4   Gas used: 51111 (0.43%)

Transaction sent: 0xd942d22363f60df63ad9b31fd227fc35fccc2cab8577d873719d8fd4fdbe611a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  EVA.transfer confirmed   Block: 5   Gas used: 51099 (0.43%)

tests/test_eva.py::test_transfer PASSED
tests/test_eva.py::test_transfer_from RUNNING
Transaction sent: 0xe316399382e4379a87101b9e5c01c0c490b5e53050cf554ad326c8ef2b6c14c7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  EVA.constructor confirmed   Block: 6   Gas used: 556372 (4.64%)
  EVA deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0x6e96b205eed75f0841e39115596fb527cd3417f1778251d919f7f9937c4ea866
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  EVA.transfer confirmed   Block: 7   Gas used: 51111 (0.43%)

Transaction sent: 0xb07ab75a77723736b306f0f53cccc415c046a84d1b226d6653ebbf343ebb5cae
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  EVA.approve confirmed   Block: 8   Gas used: 44259 (0.37%)

Transaction sent: 0x52e39bea670ed65a32aa9322173f2f26d8916d26933ec78ebc520038e4507e77
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  EVA.transferFrom confirmed   Block: 9   Gas used: 42941 (0.36%)

Transaction sent: 0x4f8e9f04e798d679f29b28e0cd719d856dfbee2438567732098d77934b6dd733
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  EVA.approve confirmed (reverted)   Block: 10   Gas used: 22128 (0.18%)

Transaction sent: 0xb725e583620052effa2411a25df477e9b4d0da7480efd4b113870829db9c003b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  EVA.approve confirmed (reverted)   Block: 11   Gas used: 21935 (0.18%)

Transaction sent: 0x44ac97af9f18908e084b997887cce4fca8e064781813654c68e9899015f12e8b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  EVA.renounceOwnership confirmed (reverted)   Block: 12   Gas used: 22133 (0.18%)

Transaction sent: 0xb41bd1d82ba7745c0c0594a76573ee224e48b4abca22c3d2c3451bb00064475b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  EVA.transferOwnership confirmed (reverted)   Block: 13   Gas used: 22563 (0.19%)

Transaction sent: 0x3c1a74763af900894f47904ab0cf1871c9047f5b844837272e827b970dd51f15
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  EVA.transferOwnership confirmed   Block: 14   Gas used: 30175 (0.25%)

tests/test_eva.py::test_transfer_from PASSED
```

Annexes

Testing code:

Testing_EVA Token:

```
from brownie import (
    reverts,
)


from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    get_account,
)


from scripts.deploy import (
    deploy_eva,
)


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

    token = deploy_eva(owner)

    with reverts("ERC20InvalidSender: 0x0000000000000000000000000000000000000000"):
```

```
        token.transfer(extra, 1e18, {"from": ZERO_ADDRESS})
    with reverts("ERC20InvalidReceiver: 0x0000000000000000000000000000000000000000"):
        token.transfer(ZERO_ADDRESS, 1e18, {"from": extra})

    assert token.balanceOf(owner) == 21000000e18
    tx = token.transfer(other, 1e18, {"from": owner})
    assert tx.events['Transfer'][0]['from'] == owner
    assert tx.events['Transfer'][0]['to'] == other
    assert tx.events['Transfer'][0]['value'] == 1e18

    tx = token.transfer(extra, 1e17, {"from": other})
    assert tx.events['Transfer'][0]['from'] == other
    assert tx.events['Transfer'][0]['to'] == extra
    assert tx.events['Transfer'][0]['value'] == 1e17


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

    token = deploy_eva(owner)

    token.transfer(other, 2e18, {"from": owner})
    assert token.allowance(other, extra) == 0
    token.approve(extra, 1e18, {"from": other})
    assert token.allowance(other, extra) == 1e18
```

```
tx = token.transferFrom(other, extra, 1e18, {"from": extra})
assert tx.events['Transfer'][0]['from'] == other
assert tx.events['Transfer'][0]['to'] == extra
assert tx.events['Transfer'][0]['value'] == 1e18


with reverts("ERC20InvalidApprover: 0x0000000000000000000000000000000000000000"):
    token.approve(extra, 1e18, {"from": ZERO_ADDRESS})
with reverts("ERC20InvalidSpender: 0x0000000000000000000000000000000000000000"):
    token.approve(ZERO_ADDRESS, 1e18, {"from": other})


with reverts():
    token.renounceOwnership({"from": other})
with reverts("OwnableInvalidOwner: 0x0000000000000000000000000000000000000000"):
    token.transferOwnership(ZERO_ADDRESS, {"from": owner})
token.transferOwnership(other, {"from": owner})
```

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	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 the EVA Token project, we inform you that the project has met the necessary security standards. The eva token contract is an extension of an open zeppelin erc20 token to have the highest possible security.

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 adEVA Token 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 adEVA Token, 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 serEVA Tokens 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 serEVA Tokens, 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 serEVA Tokens may access, and depend upon, multiple layers of third parties.