Assure DeFi®

THE VERIFICATION GOLD STANDARD

VERIFIED BY ASSURE DEFI ® INTEGRITY ✦ TRUST ✦ CREDIBILITY

Security Assessment

Devour

Date: 12/05/2024

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🟡 Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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

Insecure	Poorly Secured	Secured	**Well Secured**

Scope

Target Code And Revision

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

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	UNCX Token minter - ENMT.sol https://etherscan.io/token/0xe5a733681bbe6cd8c764bb8078ef8e13a576dd78
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

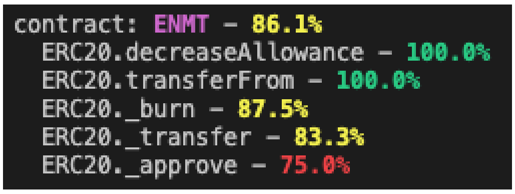
No informational severity issues were found.

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.*

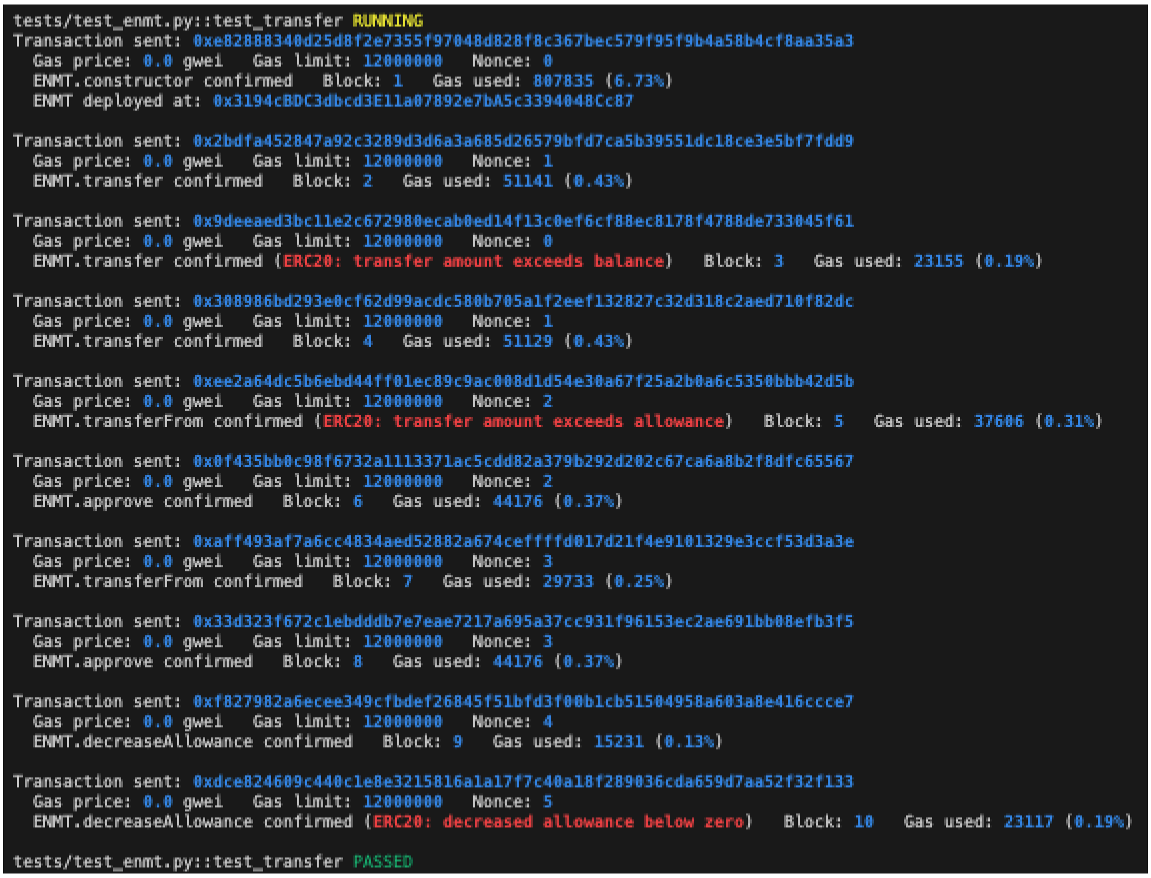
Devour staking test:

Coverages:

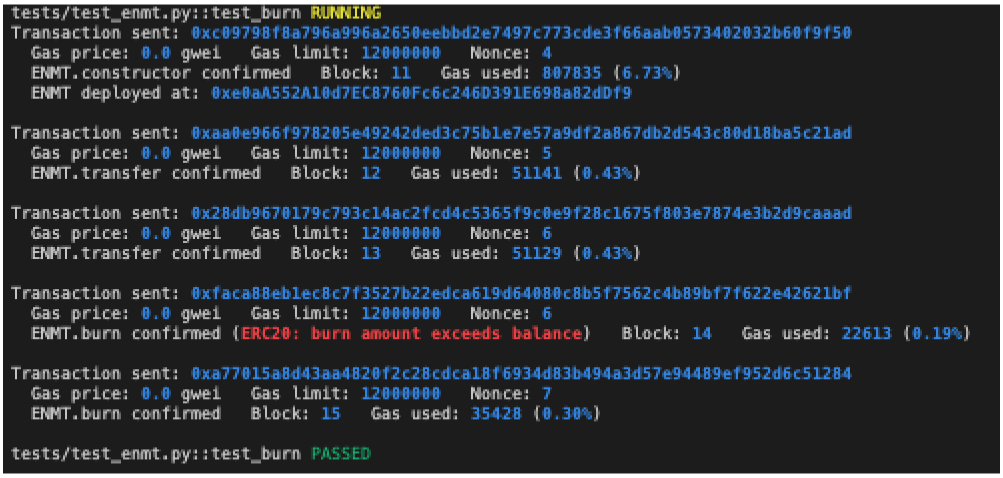


```
contract: ENMT - 86.1%
  ERC20.decreaseAllowance - 100.0%
  ERC20.transferFrom - 100.0%
  ERC20._burn - 87.5%
  ERC20._transfer - 83.3%
  ERC20._approve - 75.0%
```

Contract test:



```
tests/test_enmt.py::test_transfer RUNNING
Transaction sent: 0xe82888340d25d8f2e7355f97048d828f8c367bec579f95f9b4a58b4cf8aa35a3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ENMT.constructor confirmed   Block: 1   Gas used: 807835 (6.73%)
  ENMT deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x2bdfa452847a92c3289d3d6a3a685d26579bfd7ca5b39551dc18ce3e5bf7fdd9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ENMT.transfer confirmed   Block: 2   Gas used: 51141 (0.43%)

Transaction sent: 0x9deeaed3bc11e2c672980ecab0ed14f13c0ef6cf88ec8178f4788de733045f61
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ENMT.transfer confirmed (ERC20: transfer amount exceeds balance)   Block: 3   Gas used: 23155 (0.19%)

Transaction sent: 0x308986bd293e0cf62d99acdc580b705a1f2eef132827c32d318c2aed710f82dc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ENMT.transfer confirmed   Block: 4   Gas used: 51129 (0.43%)

Transaction sent: 0xee2a64dc5b6ebd44ff01ec89c9ac008d1d54e30a67f25a2b0a6c5350bbb42d5b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  ENMT.transferFrom confirmed (ERC20: transfer amount exceeds allowance)   Block: 5   Gas used: 37606 (0.31%)

Transaction sent: 0x0f435bb0c98f6732a1113371ac5cdd82a379b292d202c67ca6a8b2f8dfc65567
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  ENMT.approve confirmed   Block: 6   Gas used: 44176 (0.37%)

Transaction sent: 0xaff493af7a6cc4834aed52882a674cefffd017d21f4e9101329e3ccf53d3a3e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  ENMT.transferFrom confirmed   Block: 7   Gas used: 29733 (0.25%)

Transaction sent: 0x33d323f672c1ebdddb7e7eae7217a695a37cc931f96153ec2ae691bb08efb3f5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  ENMT.approve confirmed   Block: 8   Gas used: 44176 (0.37%)

Transaction sent: 0xf827982a6ecee349cfbdef26845f51bfd3f00b1cb51504958a603a8e416ccce7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ENMT.decreaseAllowance confirmed   Block: 9   Gas used: 15231 (0.13%)

Transaction sent: 0xdce824609c440c1e8e3215816a1a17f7c40a18f289036cda659d7aa52f32f133
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  ENMT.decreaseAllowance confirmed (ERC20: decreased allowance below zero)   Block: 10   Gas used: 23117 (0.19%)

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



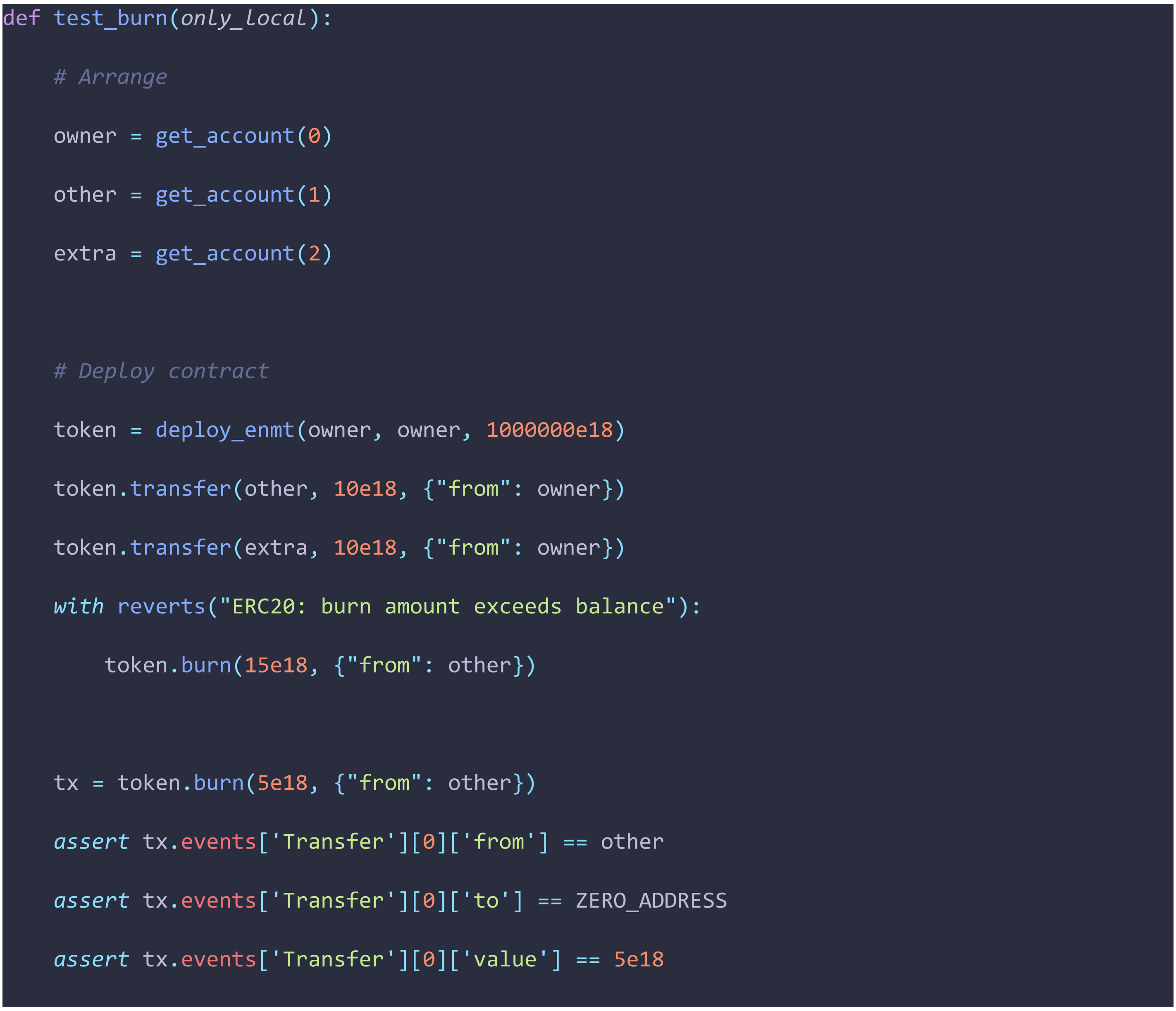
```
tests/test_enmt.py::test_burn RUNNING
Transaction sent: 0xc09798f8a796a996a2650eebbd2e7497c773cde3f66aab0573402032b60f9f50
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ENMT.constructor confirmed   Block: 11   Gas used: 807835 (6.73%)
  ENMT deployed at: 0xe0aA552A10d7EC8760Fc6c246D391E698a82dDf9

Transaction sent: 0xaa0e966f978205e49242ded3c75b1e7e57a9df2a867db2d543c80d18ba5c21ad
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  ENMT.transfer confirmed   Block: 12   Gas used: 51141 (0.43%)

Transaction sent: 0x28db9670179c793c14ac2fcd4c5365f9c0e9f28c1675f803e7874e3b2d9caaad
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  ENMT.transfer confirmed   Block: 13   Gas used: 51129 (0.43%)

Transaction sent: 0xfaca88eb1ec8c7f3527b22edca619d64080c8b5f7562c4b89bf7f622e42621bf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  ENMT.burn confirmed (ERC20: burn amount exceeds balance)   Block: 14   Gas used: 22613 (0.19%)

Transaction sent: 0xa77015a8d43aa4820f2c28cdca18f6934d83b494a3d57e94489ef952d6c51284
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  ENMT.burn confirmed   Block: 15   Gas used: 35428 (0.30%)

tests/test_enmt.py::test_burn PASSED
```

Annexes

Testing code:

```python
from brownie import (
    reverts
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    get_account,
)

from scripts.deploy import (
    deploy_enmt,
)

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

    # Deploy contract
    token = deploy_enmt(owner, owner, 1000000e18)
    tx = token.transfer(other, 10e18, {"from": owner})
    assert tx.events['Transfer'][0]['from'] == owner
    assert tx.events['Transfer'][0]['to'] == other
    assert tx.events['Transfer'][0]['value'] == 10e18
```

```
with reverts("ERC20: transfer amount exceeds balance"):
    token.transfer(extra, 15e18, {"from": other})

tx = token.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

with reverts("ERC20: transfer amount exceeds allowance"):
    token.transferFrom(other, extra, 1e18, {"from": owner})

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

tx = token.approve(owner, 5e18, {"from": other})
assert tx.events['Approval'][0]['owner'] == other
assert tx.events['Approval'][0]['spender'] == owner
assert tx.events['Approval'][0]['value'] == 5e18

token.decreaseAllowance(owner, 5e18, {"from": other})

with reverts("ERC20: decreased allowance below zero"):
    token.decreaseAllowance(owner, 5e18, {"from": other})
```



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

    # Deploy contract
    token = deploy_enmt(owner, owner, 1000000e18)
    token.transfer(other, 10e18, {"from": owner})
    token.transfer(extra, 10e18, {"from": owner})
    with reverts("ERC20: burn amount exceeds balance"):
        token.burn(15e18, {"from": other})

    tx = token.burn(5e18, {"from": other})
    assert tx.events['Transfer'][0]['from'] == other
    assert tx.events['Transfer'][0]['to'] == ZERO_ADDRESS
    assert tx.events['Transfer'][0]['value'] == 5e18
```

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	**90/100**
Assure KYC	Not completed
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.

Audit PASS

Following our comprehensive security audit of the staking contract for Devour project, the audit has been successfully completed and passed with 0 issues detected.

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.