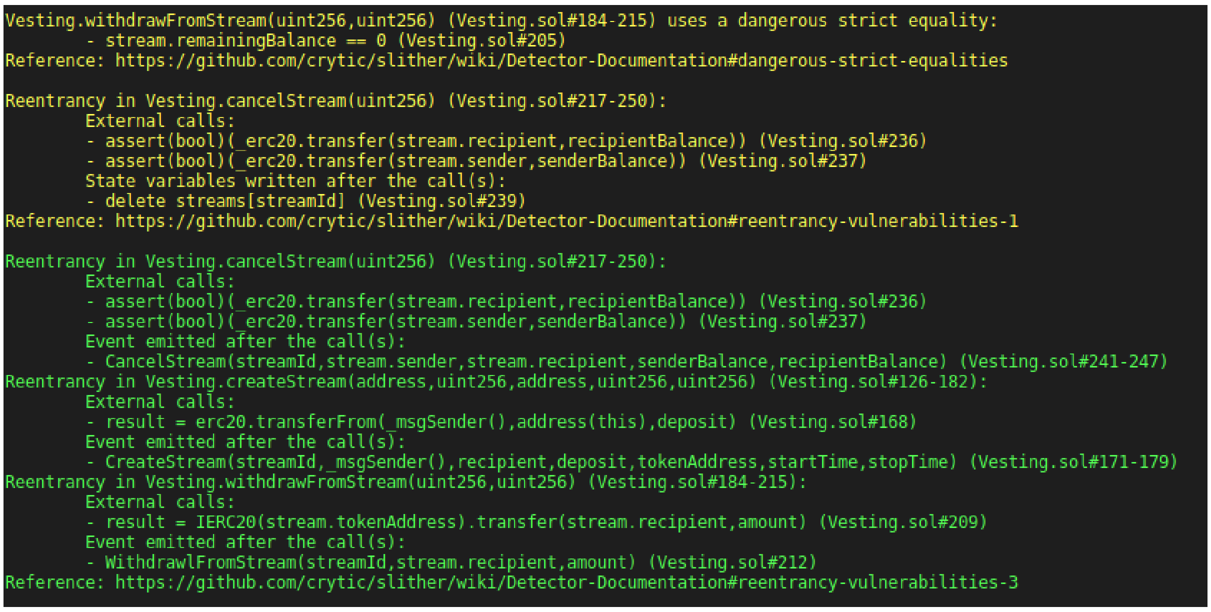
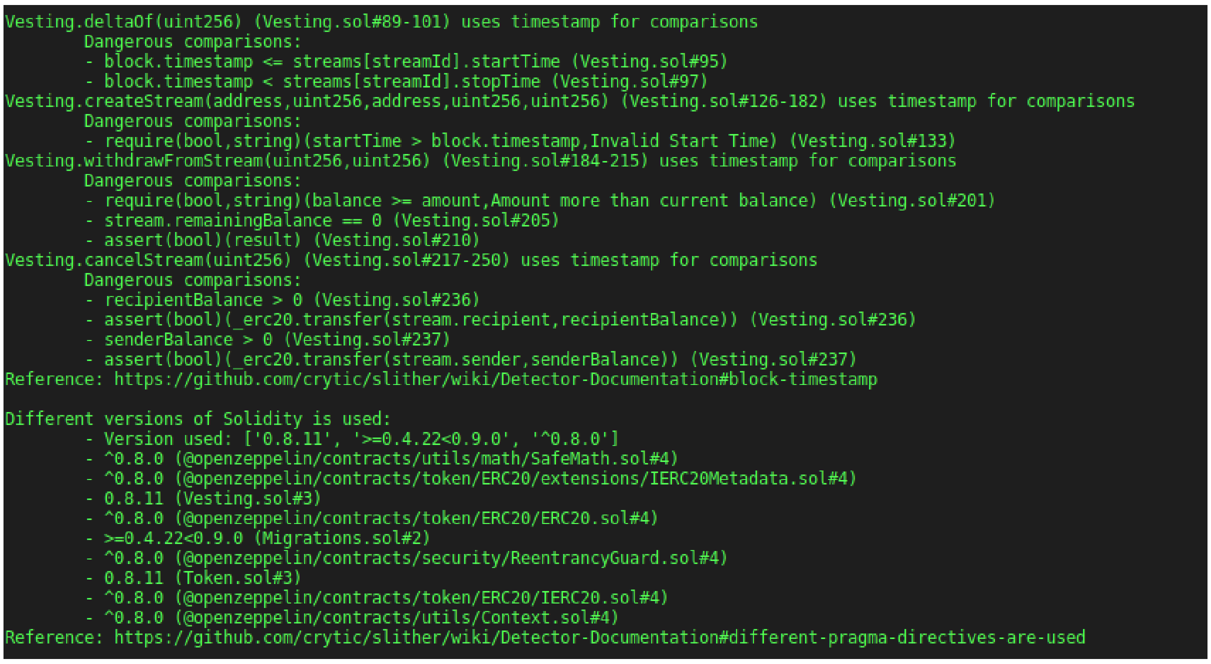
Vesting Protocol

Autonomous Testing Report

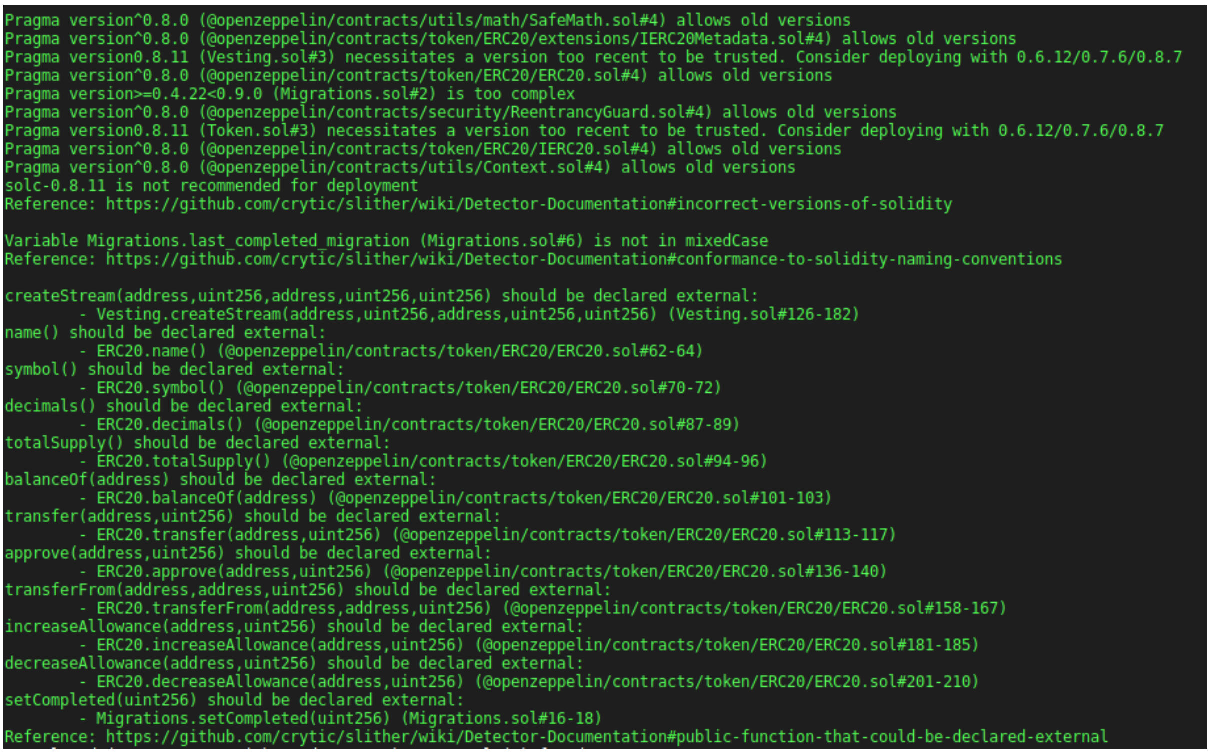
1. Slither Report



```
Vesting.withdrawFromStream(uint256,uint256) (Vesting.sol#184-215) uses a dangerous strict equality:
        - stream.remainingBalance == 0 (Vesting.sol#205)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Vesting.cancelStream(uint256) (Vesting.sol#217-250):
        External calls:
        - assert(bool)(_erc20.transfer(stream.recipient,recipientBalance)) (Vesting.sol#236)
        - assert(bool)(_erc20.transfer(stream.sender,senderBalance)) (Vesting.sol#237)
        State variables written after the call(s):
        - delete streams[streamId] (Vesting.sol#239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Reentrancy in Vesting.cancelStream(uint256) (Vesting.sol#217-250):
        External calls:
        - assert(bool)(_erc20.transfer(stream.recipient,recipientBalance)) (Vesting.sol#236)
        - assert(bool)(_erc20.transfer(stream.sender,senderBalance)) (Vesting.sol#237)
        Event emitted after the call(s):
        - CancelStream(streamId,stream.sender,stream.recipient,senderBalance,recipientBalance) (Vesting.sol#241-247)
Reentrancy in Vesting.createStream(address,uint256,address,uint256,uint256) (Vesting.sol#126-182):
        External calls:
        - result = erc20.transferFrom(_msgSender(),address(this),deposit) (Vesting.sol#168)
        Event emitted after the call(s):
        - CreateStream(streamId,_msgSender(),recipient,deposit,tokenAddress,startTime,stopTime) (Vesting.sol#171-179)
Reentrancy in Vesting.withdrawFromStream(uint256,uint256) (Vesting.sol#184-215):
        External calls:
        - result = IERC20(stream.tokenAddress).transfer(stream.recipient,amount) (Vesting.sol#209)
        Event emitted after the call(s):
        - WithdrawlFromStream(streamId,stream.recipient,amount) (Vesting.sol#212)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```



```
Vesting.deltaOf(uint256) (Vesting.sol#89-101) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp <= streams[streamId].startTime (Vesting.sol#95)
        - block.timestamp < streams[streamId].stopTime (Vesting.sol#97)
Vesting.createStream(address,uint256,address,uint256,uint256) (Vesting.sol#126-182) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(startTime > block.timestamp,Invalid Start Time) (Vesting.sol#133)
Vesting.withdrawFromStream(uint256,uint256) (Vesting.sol#184-215) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(balance >= amount,Amount more than current balance) (Vesting.sol#201)
        - stream.remainingBalance == 0 (Vesting.sol#205)
        - assert(bool)(result) (Vesting.sol#210)
Vesting.cancelStream(uint256) (Vesting.sol#217-250) uses timestamp for comparisons
        Dangerous comparisons:
        - recipientBalance > 0 (Vesting.sol#236)
        - assert(bool)(_erc20.transfer(stream.recipient,recipientBalance)) (Vesting.sol#236)
        - senderBalance > 0 (Vesting.sol#237)
        - assert(bool)(_erc20.transfer(stream.sender,senderBalance)) (Vesting.sol#237)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Different versions of Solidity is used:
        - Version used: ['0.8.11', '>=0.4.22<0.9.0', '^0.8.0']
        - ^0.8.0 (@openzeppelin/contracts/utils/math/SafeMath.sol#4)
        - ^0.8.0 (@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - 0.8.11 (Vesting.sol#3)
        - ^0.8.0 (@openzeppelin/contracts/token/ERC20/ERC20.sol#4)
        - >=0.4.22<0.9.0 (Migrations.sol#2)
        - ^0.8.0 (@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
        - 0.8.11 (Token.sol#3)
        - ^0.8.0 (@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (@openzeppelin/contracts/utils/Context.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

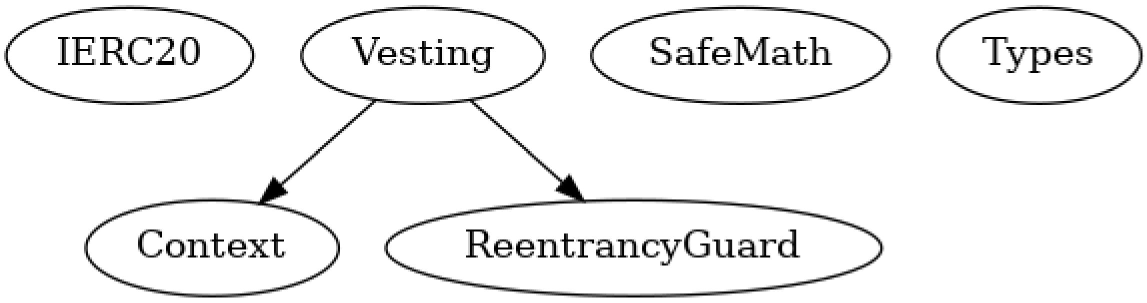


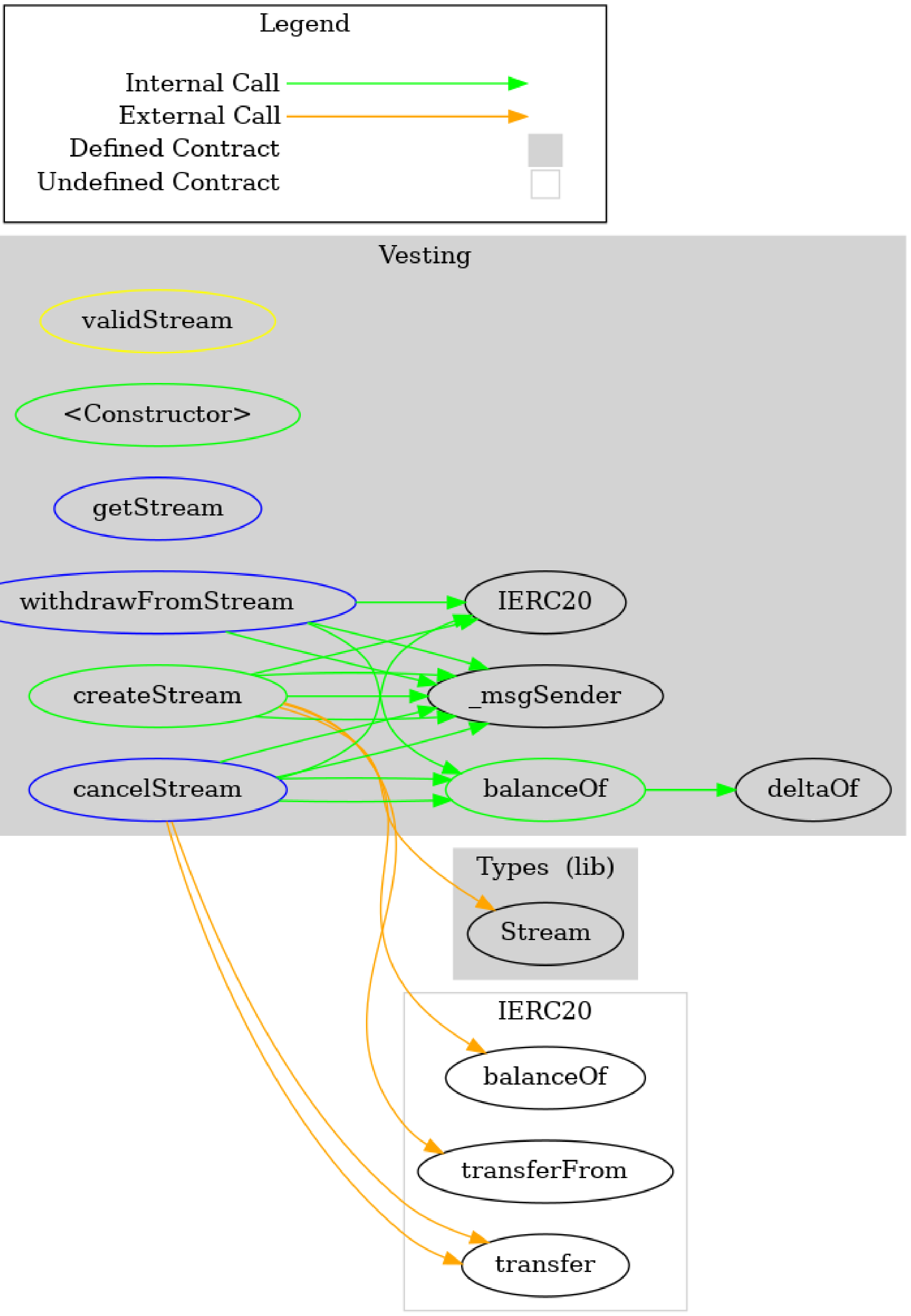
```
Pragma version^0.8.0 (@openzeppelin/contracts/utils/math/SafeMath.sol#4) allows old versions
Pragma version^0.8.0 (@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version0.8.11 (Vesting.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (@openzeppelin/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version>=0.4.22<0.9.0 (Migrations.sol#2) is too complex
Pragma version^0.8.0 (@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version0.8.11 (Token.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (@openzeppelin/contracts/utils/Context.sol#4) allows old versions
solc-0.8.11 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable Migrations.last_completed_migration (Migrations.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

createStream(address,uint256,address,uint256,uint256) should be declared external:
        - Vesting.createStream(address,uint256,address,uint256,uint256) (Vesting.sol#126-182)
name() should be declared external:
        - ERC20.name() (@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64)
symbol() should be declared external:
        - ERC20.symbol() (@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72)
decimals() should be declared external:
        - ERC20.decimals() (@openzeppelin/contracts/token/ERC20/ERC20.sol#87-89)
totalSupply() should be declared external:
        - ERC20.totalSupply() (@openzeppelin/contracts/token/ERC20/ERC20.sol#94-96)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (@openzeppelin/contracts/token/ERC20/ERC20.sol#101-103)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (@openzeppelin/contracts/token/ERC20/ERC20.sol#113-117)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (@openzeppelin/contracts/token/ERC20/ERC20.sol#136-140)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (@openzeppelin/contracts/token/ERC20/ERC20.sol#158-167)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (@openzeppelin/contracts/token/ERC20/ERC20.sol#181-185)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (@openzeppelin/contracts/token/ERC20/ERC20.sol#201-210)
setCompleted(uint256) should be declared external:
        - Migrations.setCompleted(uint256) (Migrations.sol#16-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

2. Mythril Report

No Issue Found.

3. Surya Report





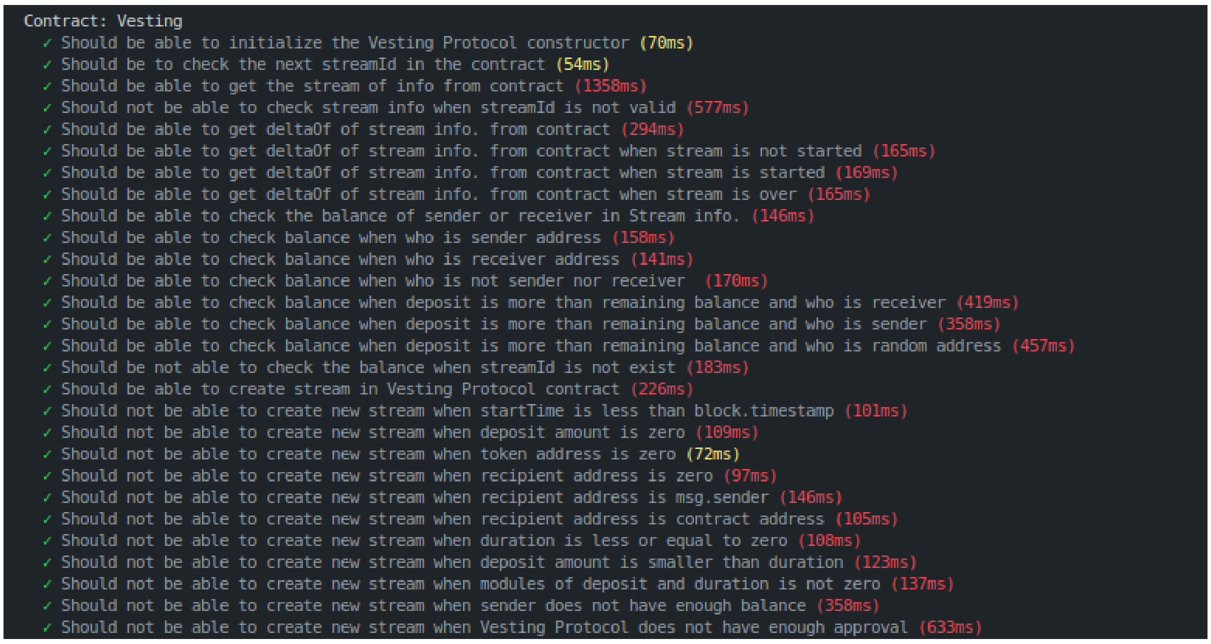
```
## Sūrya's Description Report


### Files Description Table
```

```
|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| contracts/Vesting.sol | 8921fd5e5b64d3079b9decad2f60d6d01a729cdd |


### Contracts Description Table


|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:
---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **Types** | Library |  |||
||||||
| **Vesting** | Implementation | Context, ReentrancyGuard |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | getStream | External ❗ |   | validStream |
| └ | deltaOf | Public ❗ |   | validStream |
| └ | balanceOf | Public ❗ |   | validStream |
| └ | createStream | Public ❗ | 🛑  | nonReentrant |
| └ | withdrawFromStream | External ❗ | 🛑  | nonReentrant validStream |
| └ | cancelStream | External ❗ | 🛑  | nonReentrant validStream |


### Legend

|  Symbol  |  Meaning  |
|:--------:|-----------|
|    🛑    | Function can modify state |
|    💵    | Function is payable |
```

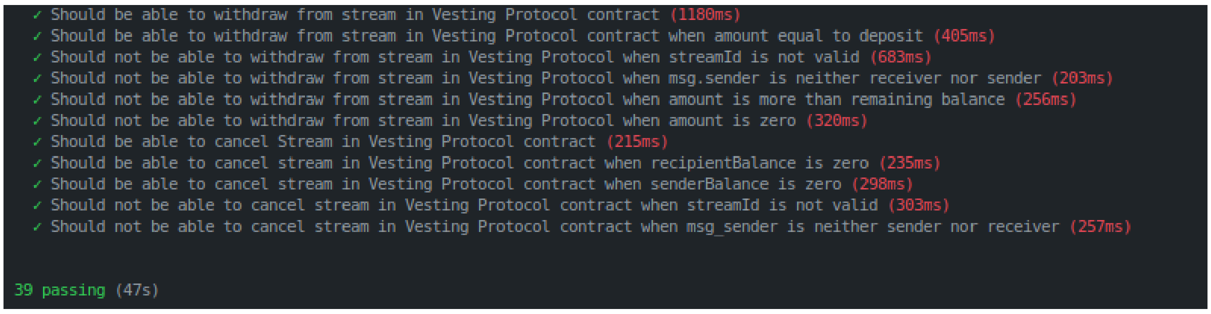
4. Solhint Report

No Issue Found.

5. Unit Test Case

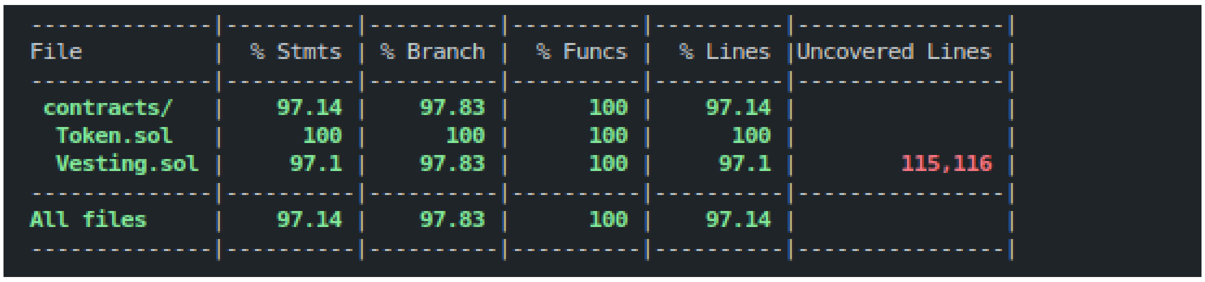


```
Contract: Vesting
  ✓ Should be able to initialize the Vesting Protocol constructor (70ms)
  ✓ Should be to check the next streamId in the contract (54ms)
  ✓ Should be able to get the stream of info from contract (1358ms)
  ✓ Should not be able to check stream info when streamId is not valid (577ms)
  ✓ Should be able to get deltaOf of stream info. from contract (294ms)
  ✓ Should be able to get deltaOf of stream info. from contract when stream is not started (165ms)
  ✓ Should be able to get deltaOf of stream info. from contract when stream is started (169ms)
  ✓ Should be able to get deltaOf of stream info. from contract when stream is over (165ms)
  ✓ Should be able to check the balance of sender or receiver in Stream info. (146ms)
  ✓ Should be able to check balance when who is sender address (158ms)
  ✓ Should be able to check balance when who is receiver address (141ms)
  ✓ Should be able to check balance when who is not sender nor receiver  (170ms)
  ✓ Should be able to check balance when deposit is more than remaining balance and who is receiver (419ms)
  ✓ Should be able to check balance when deposit is more than remaining balance and who is sender (358ms)
  ✓ Should be able to check balance when deposit is more than remaining balance and who is random address (457ms)
  ✓ Should be not able to check the balance when streamId is not exist (183ms)
  ✓ Should be able to create stream in Vesting Protocol contract (226ms)
  ✓ Should not be able to create new stream when startTime is less than block.timestamp (101ms)
  ✓ Should not be able to create new stream when deposit amount is zero (109ms)
  ✓ Should not be able to create new stream when token address is zero (72ms)
  ✓ Should not be able to create new stream when recipient address is zero (97ms)
  ✓ Should not be able to create new stream when recipient address is msg.sender (146ms)
  ✓ Should not be able to create new stream when recipient address is contract address (105ms)
  ✓ Should not be able to create new stream when duration is less or equal to zero (108ms)
  ✓ Should not be able to create new stream when deposit amount is smaller than duration (123ms)
  ✓ Should not be able to create new stream when modules of deposit and duration is not zero (137ms)
  ✓ Should not be able to create new stream when sender does not have enough balance (358ms)
  ✓ Should not be able to create new stream when Vesting Protocol does not have enough approval (633ms)
```

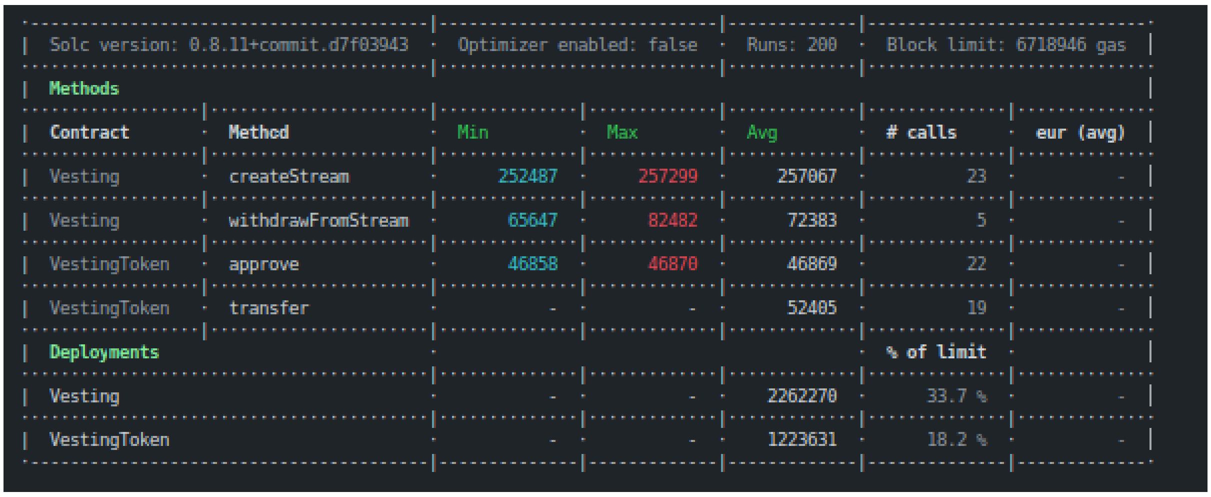


```
  ✓ Should be able to withdraw from stream in Vesting Protocol contract (1180ms)
  ✓ Should be able to withdraw from stream in Vesting Protocol contract when amount equal to deposit (405ms)
  ✓ Should not be able to withdraw from stream in Vesting Protocol when streamId is not valid (683ms)
  ✓ Should not be able to withdraw from stream in Vesting Protocol when msg.sender is neither receiver nor sender (203ms)
  ✓ Should not be able to withdraw from stream in Vesting Protocol when amount is more than remaining balance (256ms)
  ✓ Should not be able to withdraw from stream in Vesting Protocol when amount is zero (320ms)
  ✓ Should be able to cancel Stream in Vesting Protocol contract (215ms)
  ✓ Should be able to cancel stream in Vesting Protocol contract when recipientBalance is zero (235ms)
  ✓ Should be able to cancel stream in Vesting Protocol contract when senderBalance is zero (298ms)
  ✓ Should not be able to cancel stream in Vesting Protocol contract when streamId is not valid (303ms)
  ✓ Should not be able to cancel stream in Vesting Protocol contract when msg_sender is neither sender nor receiver (257ms)


39 passing (47s)
```

6. Solidity Coverage



```
--------------|----------|----------|----------|----------|----------------|
File          |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
--------------|----------|----------|----------|----------|----------------|
 contracts/   |    97.14 |    97.83 |      100 |    97.14 |                |
  Token.sol   |      100 |      100 |      100 |      100 |                |
  Vesting.sol |     97.1 |    97.83 |      100 |     97.1 |        115,116 |
--------------|----------|----------|----------|----------|----------------|
All files     |    97.14 |    97.83 |      100 |    97.14 |                |
--------------|----------|----------|----------|----------|----------------|
```



```
·-----------------------------------------|---------------------------|-------------|------------------------------·
|  Solc version: 0.8.11+commit.d7f03943  ·  Optimizer enabled: false  ·  Runs: 200  ·  Block limit: 6718946 gas  │
··········································|···························|·············|·······························
|  Methods                                                                                                         │
··················|·······················|··············|·············|·············|··············|··············
|  Contract       ·  Method               ·  Min         ·  Max        ·  Avg        ·  # calls     ·  eur (avg)  │
··················|·······················|··············|·············|·············|··············|··············
|  Vesting        ·  createStream         ·      252487  ·     257299  ·     257067  ·          23  ·          -  │
··················|·······················|··············|·············|·············|··············|··············
|  Vesting        ·  withdrawFromStream   ·       65647  ·      82482  ·      72383  ·           5  ·          -  │
··················|·······················|··············|·············|·············|··············|··············
|  VestingToken   ·  approve              ·       46858  ·      46870  ·      46869  ·          22  ·          -  │
··················|·······················|··············|·············|·············|··············|··············
|  VestingToken   ·  transfer             ·           -  ·          -  ·      52405  ·          19  ·          -  │
··················|·······················|··············|·············|·············|··············|··············
|  Deployments                            ·                                          ·  % of limit  ·             │
··········································|··············|·············|·············|··············|··············
|  Vesting                                ·           -  ·          -  ·    2262270  ·      33.7 %  ·          -  │
··········································|··············|·············|·············|··············|··············
|  VestingToken                           ·           -  ·          -  ·    1223631  ·      18.2 %  ·          -  │
·-----------------------------------------|--------------|-------------|-------------|--------------|-------------·
```