Chainport
Smart Contract Audit

Date: 16/05/21

Language: Solidity

This document may contain confidential information about IT systems and the customer's intellectual property and information about potential vulnerabilities and exploitation methods.
The report contains confidential information. This information can be used internally by the customer. The customer can release the information after fixing all vulnerabilities.

Document

Name	Chainport
Link	**https://github.com/chainport/smart-contracts/tree/develop**
Date	16/05/21

Table of contents

Scope 4
Executive Summary 4
Severity Definitions 5
AS-IS overview 5
AS-IS Governance overview 7
Audit Governance overview 10
AS-IS ChainportToken overview 11
Audit ChainportToken overview 13
Conclusion 14
Disclaimers 15
Appendix A. Evidences 16
Appendix B. Automated tools report 17
Appendix C. Gas report 19
Appendix D. Qa Automated report 20

Introduction

This report presents the Customer`s smart contract's security assessment findings and its code review conducted between May 8 – May 16 2021.

Scope

The scope of the project is Chainport smart contract, which can be found by the link below:

https://github.com/chainport/smart-contracts/tree/develop

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the widely known vulnerabilities that are considered (the full list includes them but does not limit by them):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Compiler version not fixed
- Unchecked external call - Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

According to the assessment, Customer' smart contracts are secured.

Our team performed an analysis of code functionality, manual audit, and automated checks with Slither and remix IDE (see Appendix B pic 1-4). All issues found during automated analysis reviewed have been manually, and application vulnerabilities are presented in the Audit overview section. A general overview is presented in the AS-IS section, and all found issues can be found in the Audit overview section.

We found one medium and three low issues.

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also significantly impact smart contract execution, e.g., public access to crucial functions.
Medium	Medium-level vulnerabilities are essential to fix; however, they can't lead to tokens loss.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc., code snippets that can't significantly impact execution.
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations, and info statements can't affect smart contract execution and can be ignored.

AS-IS overview

Chainport protocol contract consists of the next smart contracts:

1. **ChainportCongress.sol,ChainportCongressMembersRegistry.sol, ChainportToken.sol, Context.sol**
2. **SafeMath.sol** contracts – Openzeppelin
3. **IERC20.sol,IERC20Metadata.sol, ICongressMembersRegistry.sol –** Interfaces

Contracts from point 2 were compared to original "Openzeppelin" templates no logic differences were found. They are considered secure.

Contracts from point 3 are interfaces that include header files.

AS–IS Governance overview

ChainportCongress.sol contract does not inherit but implements a safe match library.

ChainportCongress.sol contract **init** functions:

setMembersRegistry function was called with following parameters:

- address(_membersRegistry)

propose function was called with following parameters:

- address[] memory(targets)
- uint[] memory(values)
- string[] memory(signatures)
- bytes[] memory(calldatas)
- string memory(description)

castVote function was called with following parameters:

- uint(proposalId)
- bool(support)

execute function was called with following parameters:

- uint(proposalId)

cancel function was called with following parameters:

- uint(proposalId)

_castVote function was called with following parameters:

- address(voter)
- uint(proposalId)

- bool(support)

getActions function was called with following parameters:

- uint(proposalId)

add256 function was called with following parameters:

- uint256(a)
- uint256(b)

getMembersRegistry function was called without parameters.

receive function was called without parameters.

ChainportCongressMembersRegistry.sol contract **init** functions:

changeMinimumQuorum function was called with following parameters:

- uint(newMinimumQuorum)

addMember function was called with following parameters:

- address(targetMember)
- bytes32(memberName)

addMemberInternal function was called with following parameters:

- address(targetMember)
- bytes32(memberName)

removeMember function was called with following parameters:

- address(targetMember)

isMember function was called with following parameters:

- address(_address)

getMemberInfo function was called with following parameters:

- address(_member)

getMinimalQuorum function was called without parameters.

getNumberOfMembers function was called without parameters.

getAllMemberAddresses function was called without parameters.

Context.sol contract **init** functions:

_msgSender function was called without parameters.

_msgData function was called without parameters.

Governance overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

1. There is a certain possibility of the risk of losing two quorum members which will lead to the impossibility of approving the vote, or set a new minimum number of members by a new vote. In this case, the voting system will become ineffective.
    a. Consider either changing the logic or add the ability to add members.

Low

2. The following syntax is deprecated:
   f.gas(...)(), f.value(...)() and (new C).value(...)().
   You can replace these calls by f{gas: ..., value: ...}()
   and (new C){value: ...}(). (see Appendix A pic. 1 for evidence)
3. Requirement Not informative:
   It is recommended to add a message. (see Appendix A pic. 2 for evidence)
4. Code is not optimized for gas usage:
   There is a lot of logic in the function. It is recommended that the user active flag be set to false. (see Appendix A pic. 3 for evidence)

AS-IS ChainportToken overview

ChainportToken.sol contract inherits the class Context, IERC20 and IERC20Metadata.

ChainportToken.sol contract **init** functions:

balanceOf function was called with following parameters:

- address(account)

allowance function was called with following parameters:

- address(owner)
- address(spender)

approve function was called with following parameters:

- address(spender)
- uint256(amount)

transferFrom function was called with following parameters:

- address(sender)
- address(recipient)
- uint256(amount)

burn function was called with following parameters:

- uint(amount)

increaseAllowance function was called with following parameters:

- address(spender)
- uint256(addedValue)

decreaseAllowance function was called with following parameters:

- address(spender)

- uint256(subtractedValue)

_transfer function was called with following parameters:

- address(sender)
- address(recipient)
- uint256(amount)

_burn function was called with following parameters:

- address(account)
- uint256(amount)

_approve function was called with following parameters:

- address(owner)
- address(spender)

name function was called without parameters.

symbol function was called without parameters.

totalSupply function was called without parameters.

decimals function was called without parameters.

ChainportToken overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found

Low

No low severity vulnerabilities were found

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For the contract, a high-level description of functionality was presented in the report's As-is overview section.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

The overall quality of the reviewed contracts is secured. Security engineers found three low and one medium vulnerability, which couldn't have any significant security impact.

Disclaimers

Disclaimer

The smart contracts given for audit had been analyzed following the best industry practices at the date of this report, concerning: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

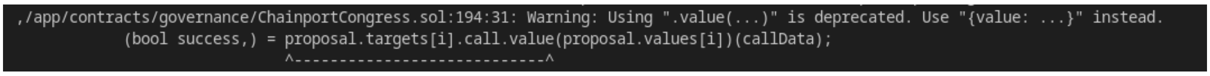
The audit makes no statements or warranties on the security of the code. It can also not be considered a sufficient assessment regarding the code's utility and safety, bug-free status, or any other contract statements. While we have done our best to conduct the analysis and produce this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. The platform, programming language, and other software related to the smart contract can have their vulnerabilities leading to hacks. Thus, the audit can’t guarantee the explicit security of the audited smart contracts.

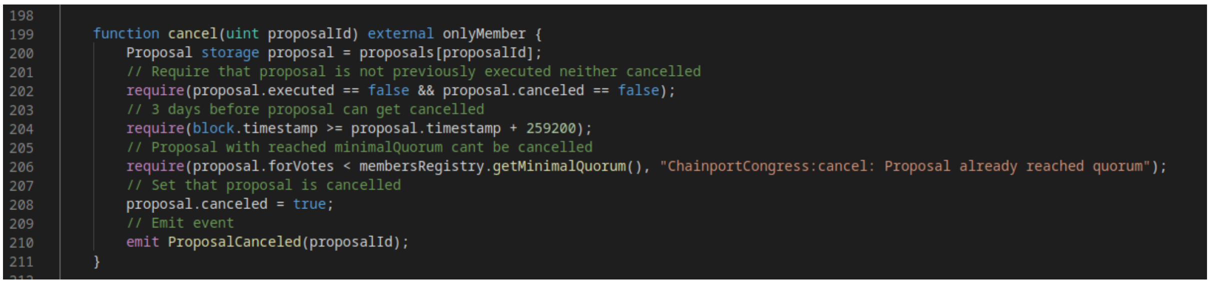
Appendix A. Evidences

Pic 1. Syntax is deprecated



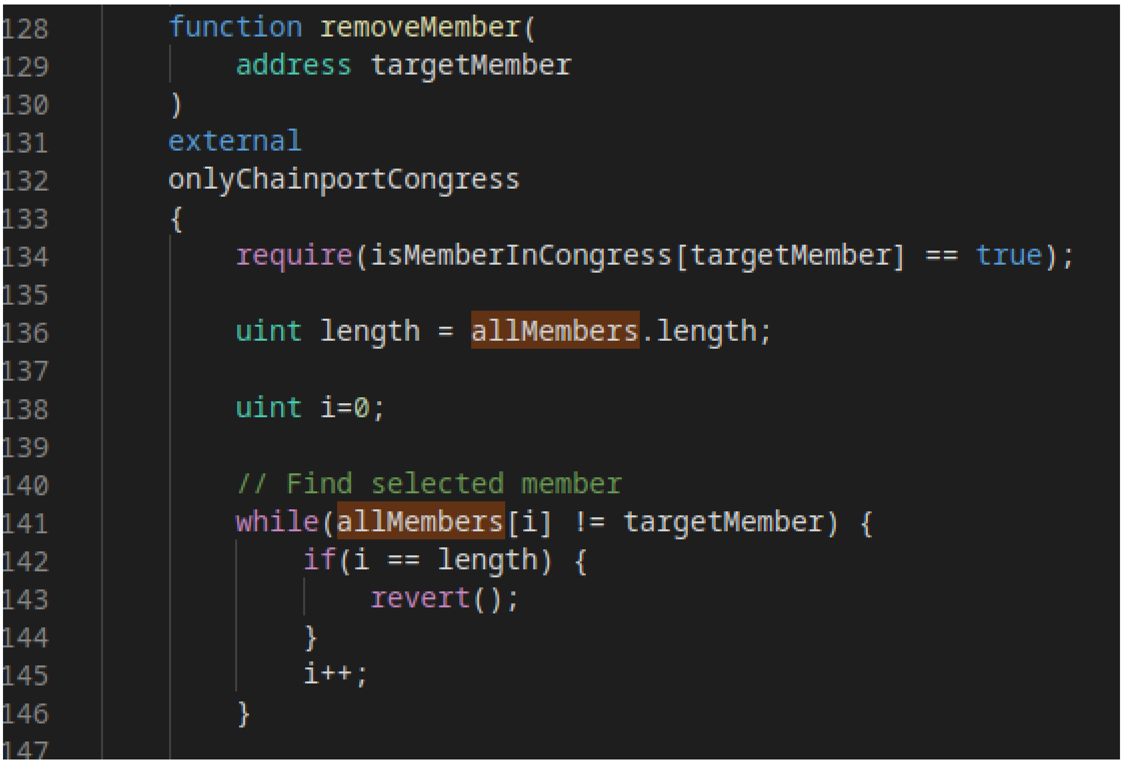
```
,/app/contracts/governance/ChainportCongress.sol:194:31: Warning: Using ".value(...)" is deprecated. Use "{value: ...}" instead.
            (bool success,) = proposal.targets[i].call.value(proposal.values[i])(callData);
                              ^--------------------------^
```

pic 2. Requirement Not informative



```
    function cancel(uint proposalId) external onlyMember {
        Proposal storage proposal = proposals[proposalId];
        // Require that proposal is not previously executed neither cancelled
        require(proposal.executed == false && proposal.canceled == false);
        // 3 days before proposal can get cancelled
        require(block.timestamp >= proposal.timestamp + 259200);
        // Proposal with reached minimalQuorum cant be cancelled
        require(proposal.forVotes < membersRegistry.getMinimalQuorum(), "ChainportCongress:cancel: Proposal already reached quorum");
        // Set that proposal is cancelled
        proposal.canceled = true;
        // Emit event
        emit ProposalCanceled(proposalId);
    }
```

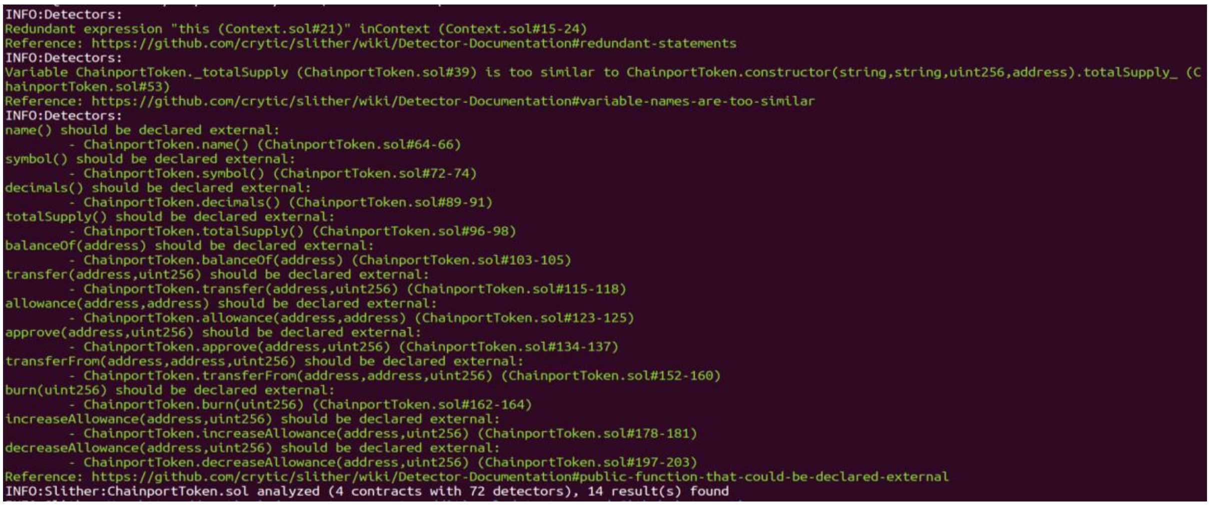
pic 3. Gas optimization.



```
    function removeMember(
        address targetMember
    )
    external
    onlyChainportCongress
    {
        require(isMemberInCongress[targetMember] == true);

        uint length = allMembers.length;

        uint i=0;

        // Find selected member
        while(allMembers[i] != targetMember) {
            if(i == length) {
                revert();
            }
            i++;
        }
```

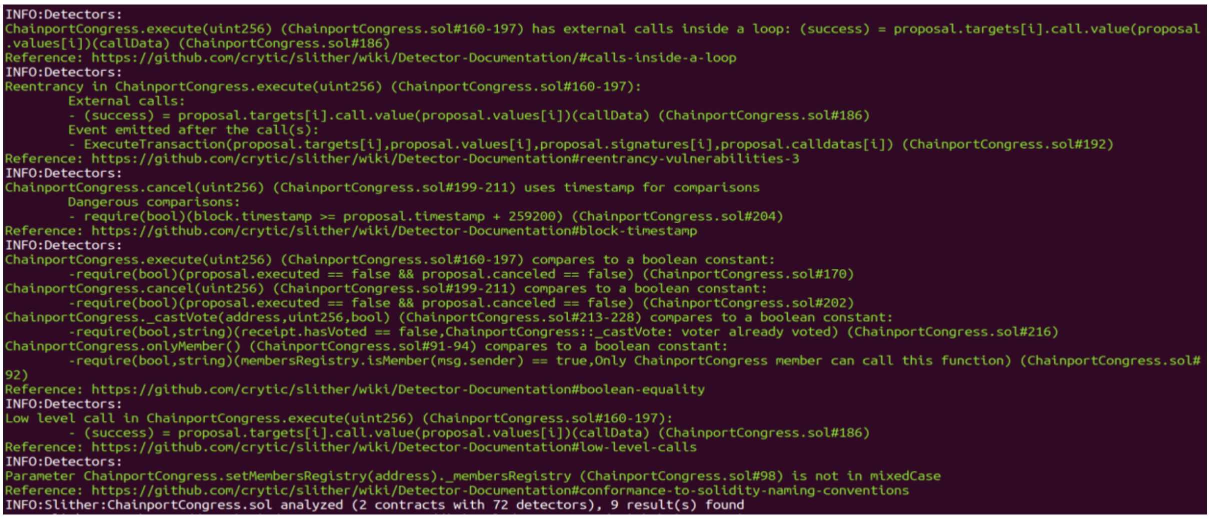
Appendix B. Automated tools reports

Pic 1. **ChainportToken** Slither automated report:



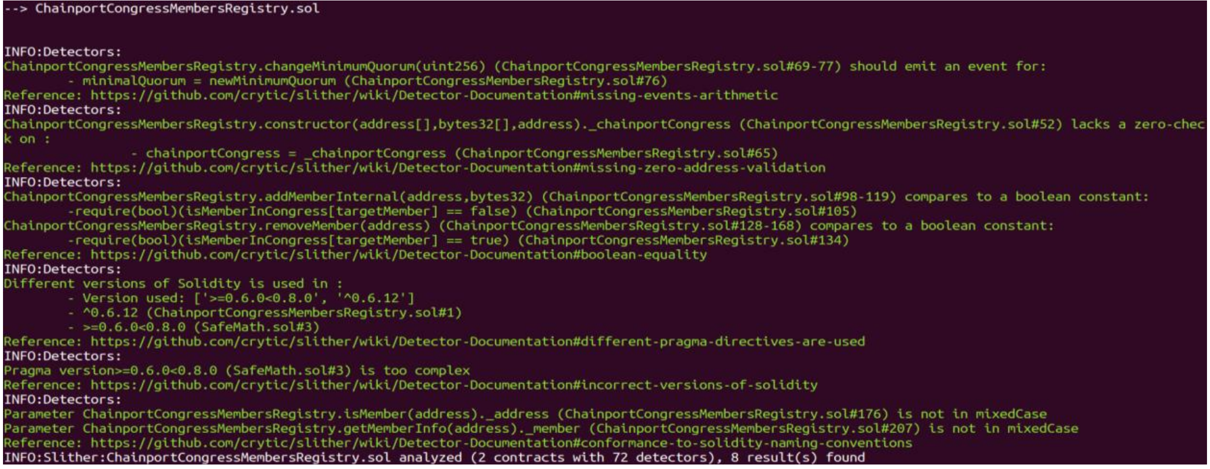
```
INFO:Detectors:
Redundant expression "this (Context.sol#21)" inContext (Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Variable ChainportToken._totalSupply (ChainportToken.sol#39) is too similar to ChainportToken.constructor(string,string,uint256,address).totalSupply_ (C
hainportToken.sol#53)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
name() should be declared external:
        - ChainportToken.name() (ChainportToken.sol#64-66)
symbol() should be declared external:
        - ChainportToken.symbol() (ChainportToken.sol#72-74)
decimals() should be declared external:
        - ChainportToken.decimals() (ChainportToken.sol#89-91)
totalSupply() should be declared external:
        - ChainportToken.totalSupply() (ChainportToken.sol#96-98)
balanceOf(address) should be declared external:
        - ChainportToken.balanceOf(address) (ChainportToken.sol#103-105)
transfer(address,uint256) should be declared external:
        - ChainportToken.transfer(address,uint256) (ChainportToken.sol#115-118)
allowance(address,address) should be declared external:
        - ChainportToken.allowance(address,address) (ChainportToken.sol#123-125)
approve(address,uint256) should be declared external:
        - ChainportToken.approve(address,uint256) (ChainportToken.sol#134-137)
transferFrom(address,address,uint256) should be declared external:
        - ChainportToken.transferFrom(address,address,uint256) (ChainportToken.sol#152-160)
burn(uint256) should be declared external:
        - ChainportToken.burn(uint256) (ChainportToken.sol#162-164)
increaseAllowance(address,uint256) should be declared external:
        - ChainportToken.increaseAllowance(address,uint256) (ChainportToken.sol#178-181)
decreaseAllowance(address,uint256) should be declared external:
        - ChainportToken.decreaseAllowance(address,uint256) (ChainportToken.sol#197-203)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:ChainportToken.sol analyzed (4 contracts with 72 detectors), 14 result(s) found
```

Pic 2. **ChainportCongress** Slither automated report:



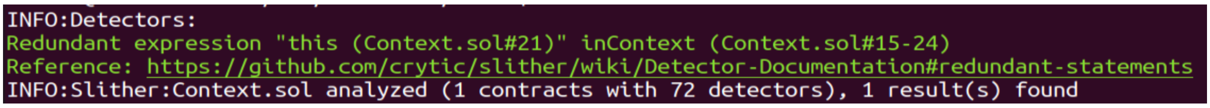
```
INFO:Detectors:
ChainportCongress.execute(uint256) (ChainportCongress.sol#160-197) has external calls inside a loop: (success) = proposal.targets[i].call.value(proposal
.values[i])(callData) (ChainportCongress.sol#186)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop
INFO:Detectors:
Reentrancy in ChainportCongress.execute(uint256) (ChainportCongress.sol#160-197):
        External calls:
        - (success) = proposal.targets[i].call.value(proposal.values[i])(callData) (ChainportCongress.sol#186)
        Event emitted after the call(s):
        - ExecuteTransaction(proposal.targets[i],proposal.values[i],proposal.signatures[i],proposal.calldatas[i]) (ChainportCongress.sol#192)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
ChainportCongress.cancel(uint256) (ChainportCongress.sol#199-211) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool)(block.timestamp >= proposal.timestamp + 259200) (ChainportCongress.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
ChainportCongress.execute(uint256) (ChainportCongress.sol#160-197) compares to a boolean constant:
        -require(bool)(proposal.executed == false && proposal.canceled == false) (ChainportCongress.sol#170)
ChainportCongress.cancel(uint256) (ChainportCongress.sol#199-211) compares to a boolean constant:
        -require(bool)(proposal.executed == false && proposal.canceled == false) (ChainportCongress.sol#202)
ChainportCongress._castVote(address,uint256,bool) (ChainportCongress.sol#213-228) compares to a boolean constant:
        -require(bool,string)(receipt.hasVoted == false,ChainportCongress::_castVote: voter already voted) (ChainportCongress.sol#216)
ChainportCongress.onlyMember() (ChainportCongress.sol#91-94) compares to a boolean constant:
        -require(bool,string)(membersRegistry.isMember(msg.sender) == true,Only ChainportCongress member can call this function) (ChainportCongress.sol#
92)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Low level call in ChainportCongress.execute(uint256) (ChainportCongress.sol#160-197):
        - (success) = proposal.targets[i].call.value(proposal.values[i])(callData) (ChainportCongress.sol#186)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter ChainportCongress.setMembersRegistry(address)._membersRegistry (ChainportCongress.sol#98) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:ChainportCongress.sol analyzed (2 contracts with 72 detectors), 9 result(s) found
```

Pic 3. **ChainportCongressMembersRegistry** Slither automated report:



```
--> ChainportCongressMembersRegistry.sol


INFO:Detectors:
ChainportCongressMembersRegistry.changeMinimumQuorum(uint256) (ChainportCongressMembersRegistry.sol#69-77) should emit an event for:
        - minimalQuorum = newMinimumQuorum (ChainportCongressMembersRegistry.sol#76)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
ChainportCongressMembersRegistry.constructor(address[],bytes32[],address)._chainportCongress (ChainportCongressMembersRegistry.sol#52) lacks a zero-chec
k on :
                - chainportCongress = _chainportCongress (ChainportCongressMembersRegistry.sol#65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
ChainportCongressMembersRegistry.addMemberInternal(address,bytes32) (ChainportCongressMembersRegistry.sol#98-119) compares to a boolean constant:
        -require(bool)(isMemberInCongress[targetMember] == false) (ChainportCongressMembersRegistry.sol#105)
ChainportCongressMembersRegistry.removeMember(address) (ChainportCongressMembersRegistry.sol#128-168) compares to a boolean constant:
        -require(bool)(isMemberInCongress[targetMember] == true) (ChainportCongressMembersRegistry.sol#134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.6.0<0.8.0', '^0.6.12']
        - ^0.6.12 (ChainportCongressMembersRegistry.sol#1)
        - >=0.6.0<0.8.0 (SafeMath.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.6.0<0.8.0 (SafeMath.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter ChainportCongressMembersRegistry.isMember(address)._address (ChainportCongressMembersRegistry.sol#176) is not in mixedCase
Parameter ChainportCongressMembersRegistry.getMemberInfo(address)._member (ChainportCongressMembersRegistry.sol#207) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:ChainportCongressMembersRegistry.sol analyzed (2 contracts with 72 detectors), 8 result(s) found
```

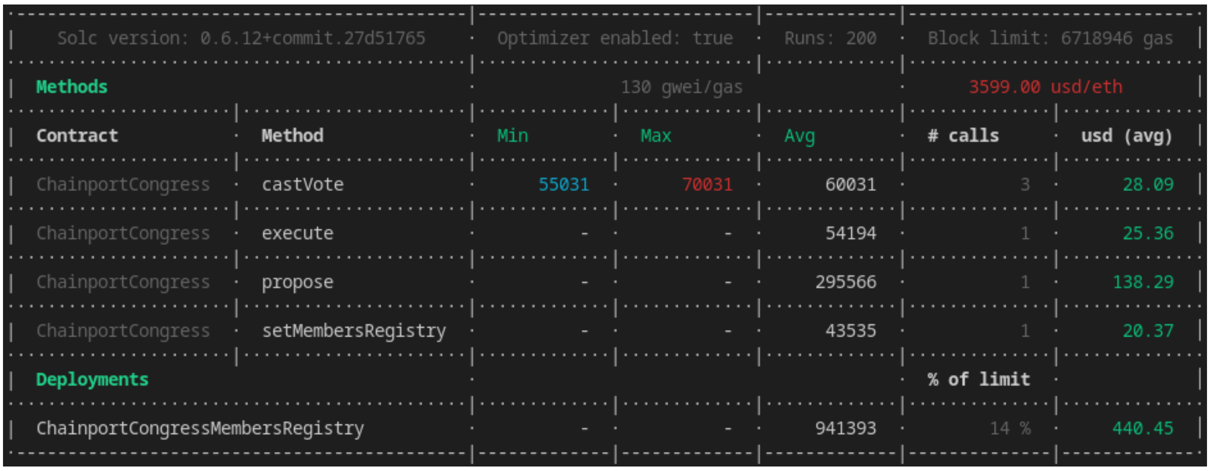
Pic 4. **Context** Slither automated report:



```
INFO:Detectors:
Redundant expression "this (Context.sol#21)" inContext (Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Slither:Context.sol analyzed (1 contracts with 72 detectors), 1 result(s) found
```

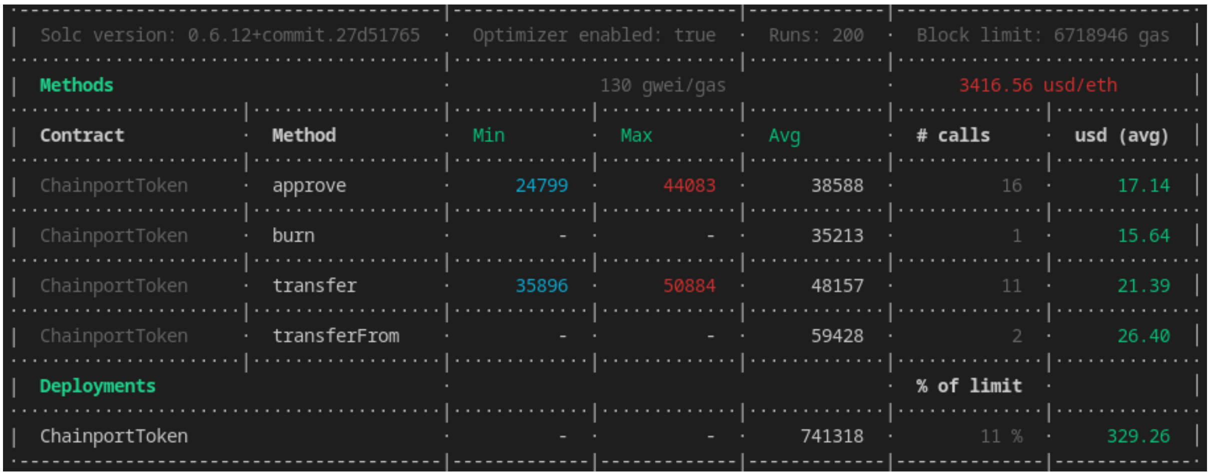
Appendix C. Gas report

Pic 1. **Governance** gas report:



```
·-------------------------------------------|--------------------------|-------------|----------------------------·
|  Solc version: 0.6.12+commit.27d51765     ·  Optimizer enabled: true ·  Runs: 200  ·  Block limit: 6718946 gas  |
············································|··························|·············|·····························
|  Methods                                  ·               130 gwei/gas             ·      3599.00 usd/eth       |
·····················|······················|·············|············|·············|·············|···············
|  Contract          ·  Method              ·  Min        ·  Max       ·  Avg        ·  # calls    ·  usd (avg)    |
·····················|······················|·············|············|·············|·············|···············
|  ChainportCongress ·  castVote            ·      55031  ·     70031  ·      60031  ·          3  ·      28.09    |
·····················|······················|·············|············|·············|·············|···············
|  ChainportCongress ·  execute             ·          -  ·         -  ·      54194  ·          1  ·      25.36    |
·····················|······················|·············|············|·············|·············|···············
|  ChainportCongress ·  propose             ·          -  ·         -  ·     295566  ·          1  ·     138.29    |
·····················|······················|·············|············|·············|·············|···············
|  ChainportCongress ·  setMembersRegistry  ·          -  ·         -  ·      43535  ·          1  ·      20.37    |
·····················|······················|·············|············|·············|·············|···············
|  Deployments                              ·                                        ·  % of limit ·               |
············································|·············|············|·············|·············|···············
|  ChainportCongressMembersRegistry         ·          -  ·         -  ·     941393  ·       14 %  ·     440.45    |
·-------------------------------------------|-------------|------------|-------------|-------------|--------------·
```

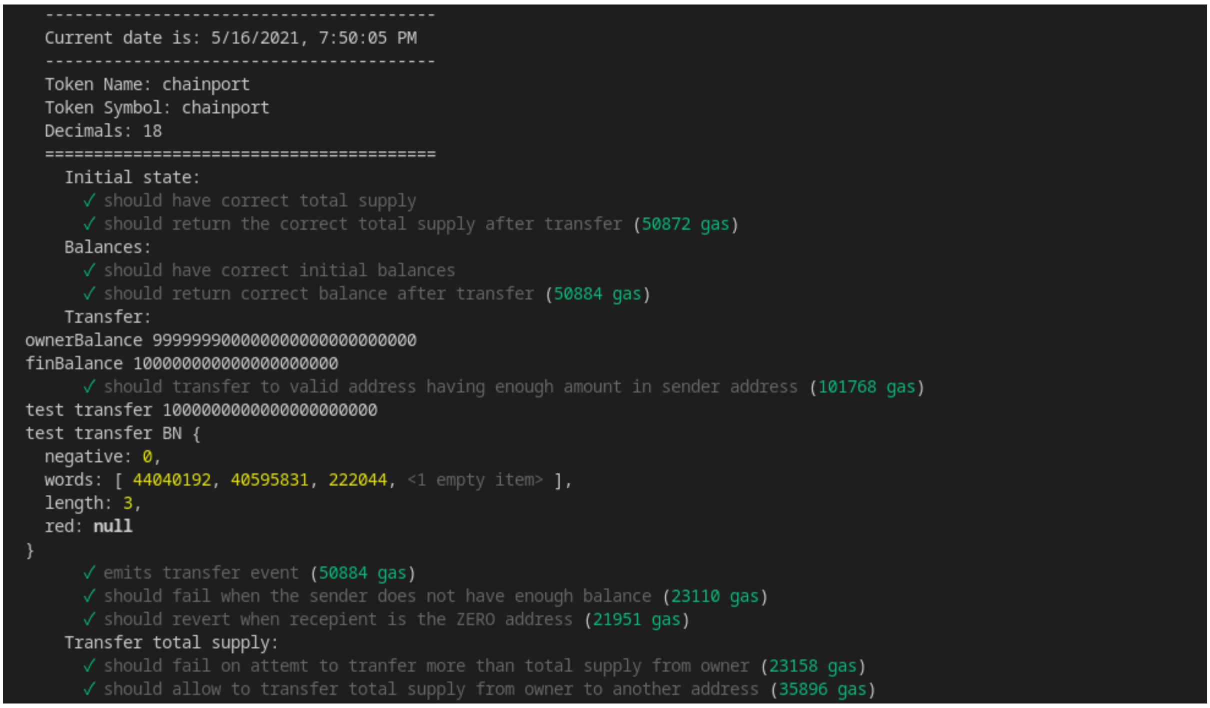
Pic 2. **ChainportToken** gas report:



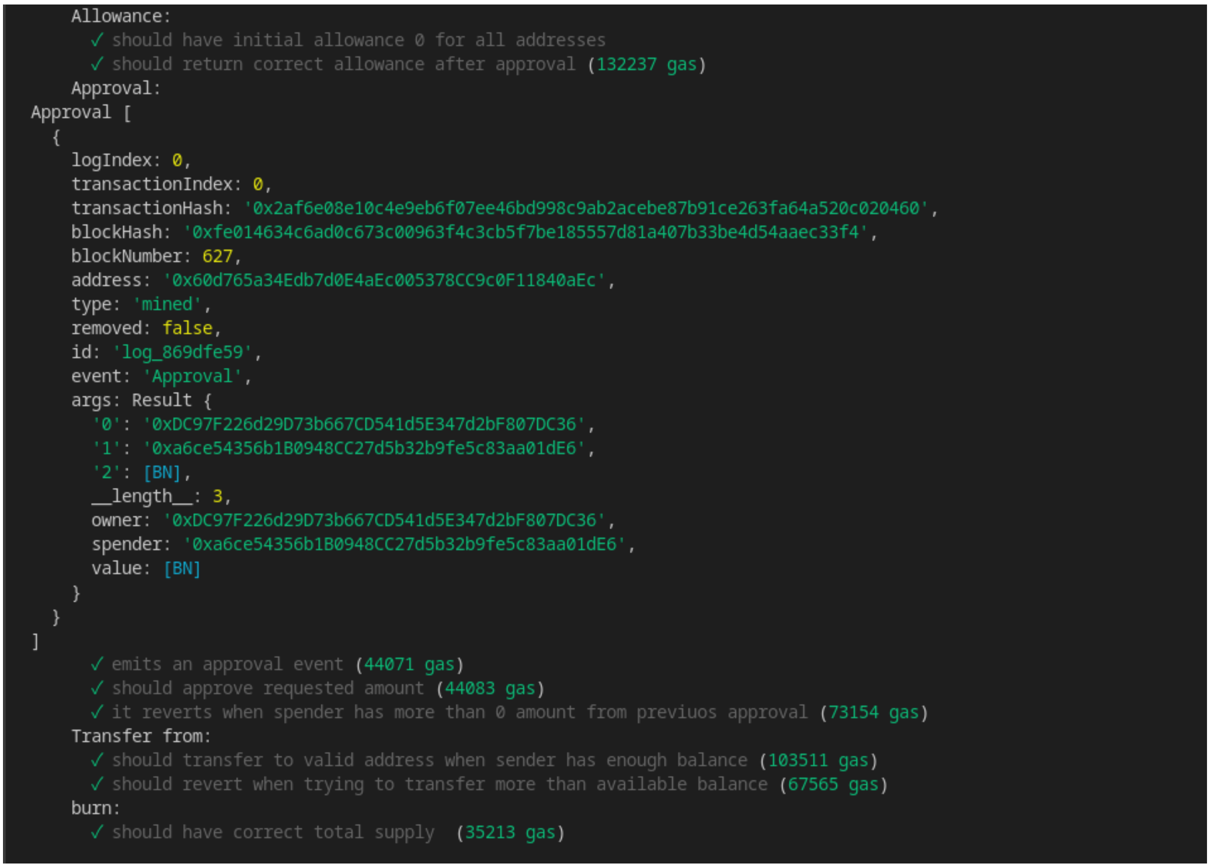
```
·----------------------------------------|---------------------------|-------------|----------------------------·
|  Solc version: 0.6.12+commit.27d51765  ·  Optimizer enabled: true  ·  Runs: 200  ·  Block limit: 6718946 gas  |
·········································|···························|·············|·····························
|  Methods                               ·               130 gwei/gas              ·      3416.56 usd/eth       |
··················|······················|·············|·············|·············|··············|··············
|  Contract       ·  Method              ·  Min        ·  Max        ·  Avg        ·  # calls     ·  usd (avg)   |
··················|······················|·············|·············|·············|··············|··············
|  ChainportToken ·  approve             ·      24799  ·      44083  ·      38588  ·          16  ·      17.14   |
··················|······················|·············|·············|·············|··············|··············
|  ChainportToken ·  burn                ·          -  ·          -  ·      35213  ·           1  ·      15.64   |
··················|······················|·············|·············|·············|··············|··············
|  ChainportToken ·  transfer            ·      35896  ·      50884  ·      48157  ·          11  ·      21.39   |
··················|······················|·············|·············|·············|··············|··············
|  ChainportToken ·  transferFrom        ·          -  ·          -  ·      59428  ·           2  ·      26.40   |
··················|······················|·············|·············|·············|··············|··············
|  Deployments                           ·                                         ·  % of limit  ·             |
·········································|·············|·············|·············|··············|··············
|  ChainportToken                        ·          -  ·          -  ·     741318  ·        11 %  ·     329.26   |
·----------------------------------------|-------------|-------------|-------------|--------------|-------------·
```

Appendix D. Qa Automated report

Pic 1. **ChainportToken** test report:



```
  ----------------------------------------
  Current date is: 5/16/2021, 7:50:05 PM
  ----------------------------------------
  Token Name: chainport
  Token Symbol: chainport
  Decimals: 18
  ========================================
    Initial state:
      ✓ should have correct total supply
      ✓ should return the correct total supply after transfer (50872 gas)
    Balances:
      ✓ should have correct initial balances
      ✓ should return correct balance after transfer (50884 gas)
    Transfer:
ownerBalance 999999900000000000000000000
finBalance 100000000000000000000
      ✓ should transfer to valid address having enough amount in sender address (101768 gas)
test transfer 1000000000000000000000
test transfer BN {
  negative: 0,
  words: [ 44040192, 40595831, 222044, <1 empty item> ],
  length: 3,
  red: null
}
      ✓ emits transfer event (50884 gas)
      ✓ should fail when the sender does not have enough balance (23110 gas)
      ✓ should revert when recepient is the ZERO address (21951 gas)
    Transfer total supply:
      ✓ should fail on attemt to tranfer more than total supply from owner (23158 gas)
      ✓ should allow to transfer total supply from owner to another address (35896 gas)
```



```
    Allowance:
      ✓ should have initial allowance 0 for all addresses
      ✓ should return correct allowance after approval (132237 gas)
    Approval:
Approval [
  {
    logIndex: 0,
    transactionIndex: 0,
    transactionHash: '0x2af6e08e10c4e9eb6f07ee46bd998c9ab2acebe87b91ce263fa64a520c020460',
    blockHash: '0xfe014634c6ad0c673c00963f4c3cb5f7be185557d81a407b33be4d54aaec33f4',
    blockNumber: 627,
    address: '0x60d765a34Edb7d0E4aEc005378CC9c0F11840aEc',
    type: 'mined',
    removed: false,
    id: 'log_869dfe59',
    event: 'Approval',
    args: Result {
      '0': '0xDC97F226d29D73b667CD541d5E347d2bF807DC36',
      '1': '0xa6ce54356b1B0948CC27d5b32b9fe5c83aa01dE6',
      '2': [BN],
      __length__: 3,
      owner: '0xDC97F226d29D73b667CD541d5E347d2bF807DC36',
      spender: '0xa6ce54356b1B0948CC27d5b32b9fe5c83aa01dE6',
      value: [BN]
    }
  }
]
      ✓ emits an approval event (44071 gas)
      ✓ should approve requested amount (44083 gas)
      ✓ it reverts when spender has more than 0 amount from previuos approval (73154 gas)
    Transfer from:
      ✓ should transfer to valid address when sender has enough balance (103511 gas)
      ✓ should revert when trying to transfer more than available balance (67565 gas)
    burn:
      ✓ should have correct total supply  (35213 gas)
```