EtherAuthority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project: OXYO2 (OX2)
Coin Name: OXYO2
Coin Ticker: KRPZA
Website: https://oxyo2.org
Platform: Binance Smart Chain
Language: Solidity
Date: July 6th, 2023

Table of contents

Introduction 4

Project Background 4

Audit Scope 4

Claimed Smart Contract Features 6

Audit Summary 9

Technical Quick Stats 10

Code Quality 11

Documentation 11

Use of Dependencies 11

AS-IS overview 12

Severity Definitions 16

Audit Findings 17

Conclusion 28

Our Methodology 29

Disclaimers 31

Appendix

- Code Flow Diagram 32
- Slither Results Log 42
- Solidity static analysis 48
- Solhint Linter 59

THIS IS SECURITY AUDIT REPORT DOCUMENT AND WHICH MAY CONTAIN INFORMATION WHICH IS CONFIDENTIAL. WHICH INCLUDES ANY POTENTIAL VULNERABILITIES AND MALICIOUS CODES WHICH CAN BE USED TO EXPLOIT THE SOFTWARE. THIS MUST BE REFERRED INTERNALLY AND ONLY SHOULD BE MADE AVAILABLE TO THE PUBLIC AFTER ISSUES ARE RESOLVED.

Introduction

EtherAuthority was contracted by the OXYO2 team to perform the Security audit of the OX2 Token, Vesting and token sale smart contracts code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on July 6th, 2023.

The purpose of this audit was to address the following:

- Ensure that all claimed functions exist and function correctly.
- Identify any security vulnerabilities that may be present in the smart contract.

Project Background

- The OXYO2 protocol covers multiple contracts, and all contracts have different functions.
  - Defi: It is used for updating dest token addresses.
  - OxyO2airdrop: It is used for airdrops.
  - OxyO2CommunityAndTeam: It is used for updating core member addresses and reward tokens.
  - OxyO2Ecosystem: It is used for ecosystems.
  - OxyO2founders: It is used for updating partner details.
  - OxyO2SaveNature: This contract is used for relief tokens.
- The smart contracts have functions like relief, ecosystem, defi, airdrop, founderReward, partnerReward, publicRelease, etc.

Audit scope

Name	**Code Review and Security Analysis Report for OXYO2 Token Smart Contracts**
Platform	**BSC / Solidity**
File 1	Defi.sol
File 1 Initial code link	0xa984192419f90d26ee7927e4e9e72af7fa48ab2a
File 2	OxyO2publicSale.sol
File 2 Initial code link	0x6714539bca94643d7d3ee2f413101762a0689bca

File 2 Revised code link	0x74300fe761ef64feeddac1a6cdf2d12374b6f462
File 3	OxyO2Ecosystem.sol
File 3 Initial code link	0xc3216b52187adb99017a43496c8780acf134ae7b
File 3 Revised code link	0xcbdc184d785f81eee042d22c9962841288315a81
File 4	OxyO2founders.sol
File 4 Initial code link	0x3c82ca3fda7506109a4a7c7bb38e02f1cc8140df
File 4 Revised code link	0xacff0eaccb5e484df716391f85813cd742a861a4
File 5	OxyO2founders1.sol
File 5 Initial code link	0x1fe33469fdd387467843b7f1d40cd3c874f6dfcc
File 5 Revised code link	0xfc5587551dee7caa0f9c8fd6c165fc3b8dabb256
File 6	OxyO2founders2.sol
File 6 Initial code link	0x6d6dfb2dd13b0603c27f0292d0d4db6fa8883215
File 6 Revised code link	0x8ab53a81f7d7ed95ee331b7a9fb50f47f4f2624b
File 7	OxyO2Strategic.sol
File 7 Initial code link	0x132848ee052e6cbdd8ef421982279e8c38d492e4
File 7 Revised code link	0xa74059d74be1e2d8c722d63f80c5c4ad2e61b619
File 8	OxyO2SaveNature.sol
File 8 Initial code link	0xca0d1822794d2c6e10cade98b143f9d3153a2520
File 8 Revised code link	0x0c0c721dc1dbc8b6a9da8ac5ae93a4b9e437ffa8
File 9	OxyO2CommunityAndTeam.sol
File 9 Initial code link	0x30588adec25edec8b3f56ef42402dd3cad4d41bf
File 9 Revised code link	0x59d125675e65f7cddb8c8c74b6ae28290ce60681
File 10	OxyO2airdrop.sol
File 10 Initial code link	0x26ba6d5de9b353c17c6642d8611a922c84707e53
File 10 Revised code link	0x12f5a4c0e0815b2d7548f65fac3f34a9956c22a3
Audit Date	July 6th, 2023
Revised Audit Date	July 10th, 2023

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
Tokenomics: ● Name : Oxyo2 (Ox2) ● Network : Binance Chain (Bep20) ● Decimals: 18 ● Maximum supply: 1 billion ● Burning: 50%	
File 1 Defi.sol ● Amount: 0.3 million ● Max: 470 million ● Defi: 47% ● Per day 300,000. ● Every 90 days 30000 coins increase. ● Total 1566 days. **The owner has control over the following functions:** ● Current owners can transfer ownership. ● Update the dest token addresses.	YES, This is valid.
File 2 OxyO2publicSale.sol ● Amount: 0.15 million **The owner has control over the following functions:** ● Release public tokens. ● Current owners can transfer ownership.	YES, This is valid.
File 3 OxyO2Ecosystem.sol ● Amount: 1.35 million ● 8% ecosystem ● Every month 13,50,000 ● Total 6 years. **The owner has control over the following functions:** ● Access the ecosystem.	YES, This is valid.

● Current owners can transfer ownership.	
File 4 OxyO2founders.sol ● Amount: 0.270 million ● 18% of founders **<u>The owner has control over the following functions:</u>** ● Added / Removed the partner addresses. ● Receive founder reward tokens. ● Current owners can transfer ownership.	**YES, This is valid.**
File 5 OxyO2founders1.sol ● Amount: 0.270 million ● Founder 10% (60000 per month). **<u>The owner has control over the following functions:</u>** ● Added / Removed the partner addresses. ● Receive founder reward tokens. ● Current owners can transfer ownership.	**YES, This is valid.**
File 6 OxyO2founders2.sol ● Amount: 0.060 million ● Founders 45%-45% (270000 each per month). **<u>The owner has control over the following functions:</u>** ● Added / Removed the partner addresses. ● Receive founder reward tokens. ● Current owners can transfer ownership.	**YES, This is valid.**
File 7 OxyO2Strategic.sol ● Amount: 1.25 million ● 90 days lock. ● After that it was divided into 24 months. ● Private & strategic: 3% **<u>The owner has control over the following functions:</u>** ● Current owners can transfer ownership. ● Private strategic access by the owner.	**YES, This is valid.**

File 8 OxyO2SaveNature.sol ● Amount: 0.5 million ● 1% save nature ● Every year 500000. **<u>The owner has control over the following functions:</u>** ● Use relief tokens. ● Current owners can transfer ownership.	**YES, This is valid.**
File 9 OxyO2CommunityAndTeam.sol ● Amount: 0.080 million ● 1% community, team & partner. ● 80000 per month. **<u>The owner has control over the following functions:</u>** ● Added / removed core member addresses. ● Current owners can transfer ownership. ● Reward tokens.	**YES, This is valid.**
File 10 OxyO2airdrop.sol ● Amount: 0.080 million ● 1% airdrop & bounty. ● 80000 per month. **<u>The owner has control over the following functions:</u>** ● Current owners can transfer ownership. ● Handle the airdrops.	**YES, This is valid.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. Also, these contracts contain owner control, which does not make them fully decentralized.

Insecure
Poor secured
Secure
Well-secured
You are here

We used various tools like Slither, Solhint and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.

All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 1 high, 3 medium and 6 low and 5 very low level issues.

We confirm that these issues have been fixed / acknowledged in the revised code.

Investors Advice: Technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Passed
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	N/A
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Features claimed	Passed
	Other programming issues	Passed
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Unused code	Passed
Gas Optimization	“Out of Gas” Issue	Passed
	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: PASSED

Code Quality

This audit scope has 10 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well-written smart contract.

The libraries in the OXYO2 protocol are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the OXYO2 protocol.

The OXYO2 team has not provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Code parts are not well commented on smart contracts.

Documentation

We were given an OXYO2 protocol smart contract code in the form of a https://bscscan.com web link. The hash of that code is mentioned above in the table.

As mentioned above, the code parts are not well commented. but the logic is straightforward. So it is easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website: https://oxyo2stake.com which provided rich information about the project architecture and tokenomics.

Use of Dependencies

As per our observation, the libraries are used in this smart contracts infrastructure that are based on well known industry standard open source projects.

Apart from libraries, its functions are used in external smart contract calls.

AS-IS overview

Defi.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	Defi	write	Fixed	No Issue
7	remainingdays	read	Passed	No Issue
8	gettime	write	Passed	No Issue
9	updateDestToken	write	Fixed	No Issue

OxyO2airdrop.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	getremainingmonths	read	Passed	No Issue
7	Airdrop	write	Fixed	No Issue

OxyO2CommunityAndTeam.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	addCoreMember	write	access only Owner	No Issue
7	subtractCoreMember	write	access only Owner	No Issue
8	getremainingmonths	read	Passed	No Issue
9	getremainingmonths	read	Fixed	No Issue
10	Reward	write	Fixed	No Issue
11	coreReward	write	Unable to get remaining rewards	Refer Audit Findings

OxyO2Ecosystem.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	transferOwnership	internal	Passed	No Issue
6	getremaining	read	Fixed	No Issue
7	ecosystem	write	access only Owner	No Issue

OxyO2founders.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	addPartner	write	access only Owner	No Issue
7	subtractPartner	write	access only Owner	No Issue
8	getremainingmonths	read	Passed	No Issue
9	getremainingmonths	read	Passed	No Issue
10	founderReward	write	access only Owner	No Issue
11	partnerReward	write	Passed	No Issue

OxyO2founders1.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	addPartner	write	access only Owner	No Issue
7	subtractPartner	write	access only Owner	No Issue
8	getremainingmonths	read	Passed	No Issue
9	getremainingmonths	read	Passed	No Issue
10	founderReward	write	access only Owner	No Issue
11	partnerReward	write	Passed	No Issue

OxyO2founders2.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	addPartner	write	access only Owner	No Issue
7	subtractPartner	write	access only Owner	No Issue
8	getremainingmonths	read	Passed	No Issue
9	getremainingmonths	read	Passed	No Issue
10	founderReward	write	access only Owner	No Issue
11	partnerReward	write	Passed	No Issue

OxyO2Strategic.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	addVoters	write	Removed	No Issue
7	getremainingmonths	read	Fixed	No Issue
8	VoterReward	write	Removed	No Issue
9	PrivateStrategic	write	access only Owner	No Issue

OxyO2publicSale.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	transferOwnership	write	access only Owner	No Issue
5	_transferOwnership	internal	Passed	No Issue
6	publicRelease	write	Fixed	No Issue
7	remainingdays	read	Passed	No Issue
8	gettime	write	Passed	No Issue

OxyO2SaveNature.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Fixed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	_transferOwnership	internal	Passed	No Issue
5	getremainingyears	read	Fixed	No Issue
6	Relief	write	Fixed	No Issue

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to token loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on 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.

Audit Findings

Critical Severity

No critical severity vulnerabilities were found in the contract code.

High Severity

(1) Fund is not released every month: **OxyO2Ecosystem.sol**

In ecosystem function, funds are released every 6 months which is wrong as per the requirement.

Resolution: We suggest correcting the logic to release funds every month.

Status: Fixed

Medium

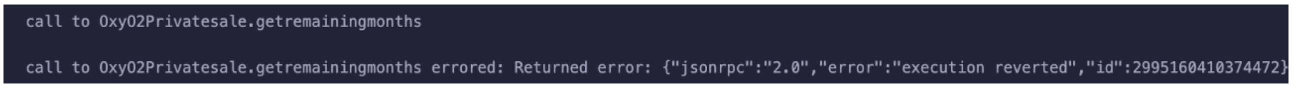
(1) Unable to get remaining rewards: **OxyO2CommunityAndTeam.sol**

In coreReward, if the contract does not have enough tokens, the user will get less tokens than set for his reward amount. And there is no functionality to give remaining tokens afterwards.

Resolution: We suggest adding functionality to give the remaining reward amount. But if this is a part of the plan, ignore this issue.

Status: Acknowledged

(2) The getremainingmonths is reverted: **OxyO2Strategic.sol**



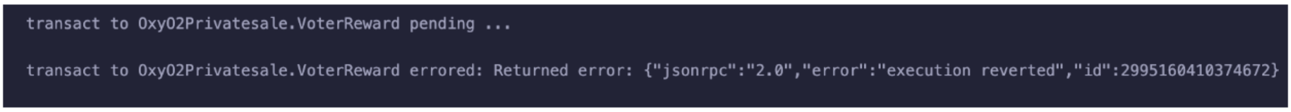
```
call to OxyO2Privatesale.getremainingmonths

call to OxyO2Privatesale.getremainingmonths errored: Returned error: {"jsonrpc":"2.0","error":"execution reverted","id":2995160410374472}
```

If the return value is negative it is reverting

Resolution: We suggest checking the logic.

Status: Function has been removed from the revised code.

(3) VoterReward is reverting: **OxyO2Strategic.sol**



```
transact to Oxy02Privatesale.VoterReward pending ...

transact to Oxy02Privatesale.VoterReward errored: Returned error: {"jsonrpc":"2.0","error":"execution reverted","id":2995160410374672}
```

VoterReward is reverting because it internally calls getremainingmonths which in turn reverts.

Resolution: We suggest checking the logic.

Status: Function has been removed from the revised code.

Low

(1) Centralization risk: **OxyO2CommunityAndTeam.sol, OxyO2founders.sol, OxyO2founders1.sol, OxyO2founders2.sol**

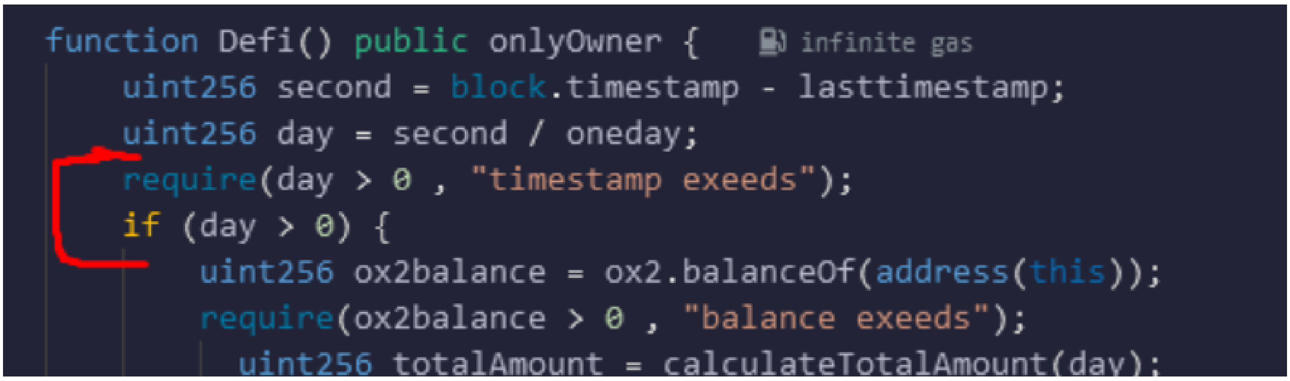
The owner can drain all the tokens before allocating them to any partners or members.

Resolution: We suggest confirming this before deployment.

Status: Acknowledged

(2) Duplicate condition code: **Defi.sol**

Function: Defi()

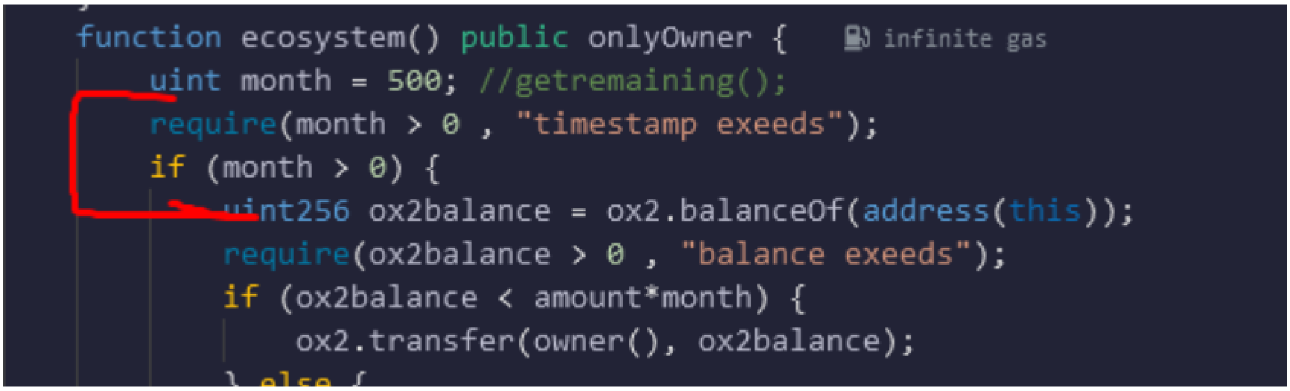


```
function Defi() public onlyOwner {    infinite gas
    uint256 second = block.timestamp - lasttimestamp;
    uint256 day = second / oneday;
    require(day > 0 , "timestamp exeeds");
    if (day > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
         uint256 totalAmount = calculateTotalAmount(day);
```

In the Defi function, day > 0 is checked in both the required and if conditions.

OxyO2Ecosystem.sol

Function: ecosystem()

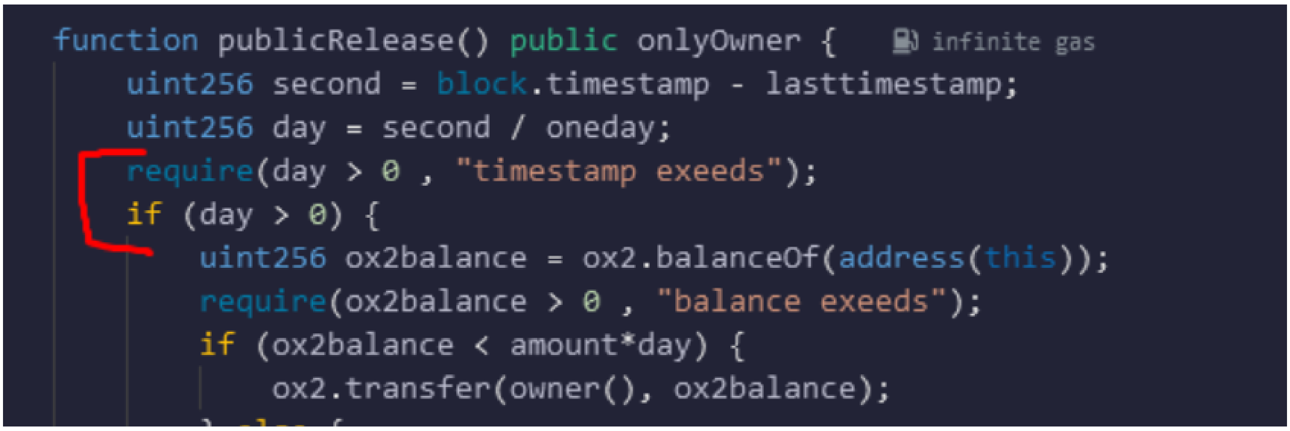


```
function ecosystem() public onlyOwner {    infinite gas
    uint month = 500; //getremaining();
    require(month > 0 , "timestamp exeeds");
    if (month > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
        if (ox2balance < amount*month) {
            ox2.transfer(owner(), ox2balance);
        } else {
```

In ecosystem function, month > 0 is checked in both the require and if condition.

OxyO2publicSale.sol

Function: publicRelease()



```
function publicRelease() public onlyOwner {    infinite gas
    uint256 second = block.timestamp - lasttimestamp;
    uint256 day = second / oneday;
    require(day > 0 , "timestamp exeeds");
    if (day > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
        if (ox2balance < amount*day) {
            ox2.transfer(owner(), ox2balance);
        } else {
```

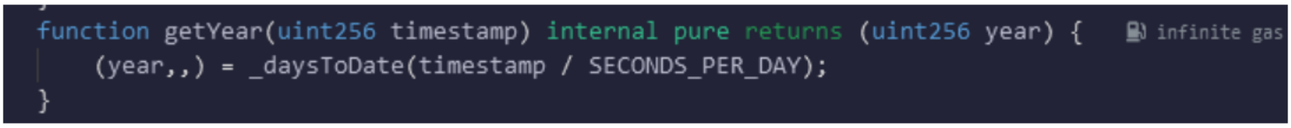
In the publicRelease function, day > 0 is checked in both the require and if condition.

Resolution: We suggest removing the require or if condition.

Status: Fixed

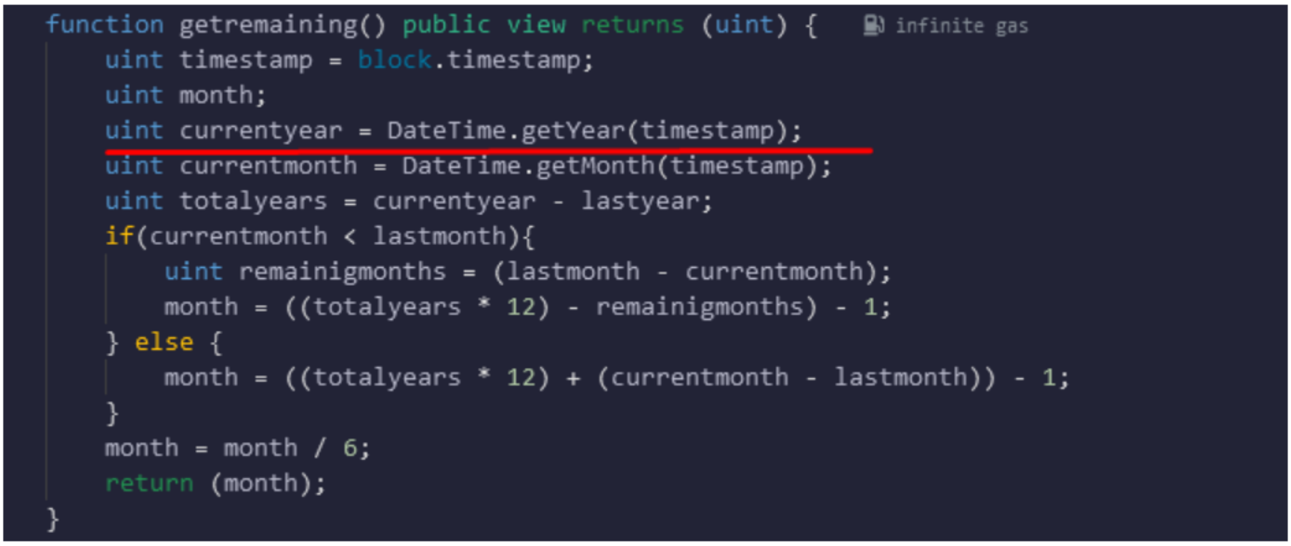
(3) Different variable type: **OxyO2Ecosystem.sol**

Function: getYear()



```
function getYear(uint256 timestamp) internal pure returns (uint256 year) {    infinite gas
    (year,,) = _daysToDate(timestamp / SECONDS_PER_DAY);
}
```

Function: getremaining()



```
function getremaining() public view returns (uint) {
    uint timestamp = block.timestamp;
    uint month;
    uint currentyear = DateTime.getYear(timestamp);
    uint currentmonth = DateTime.getMonth(timestamp);
    uint totalyears = currentyear - lastyear;
    if(currentmonth < lastmonth){
        uint remainigmonths = (lastmonth - currentmonth);
        month = ((totalyears * 12) - remainigmonths) - 1;
    } else {
        month = ((totalyears * 12) + (currentmonth - lastmonth)) - 1;
    }
    month = month / 6;
    return (month);
}
```

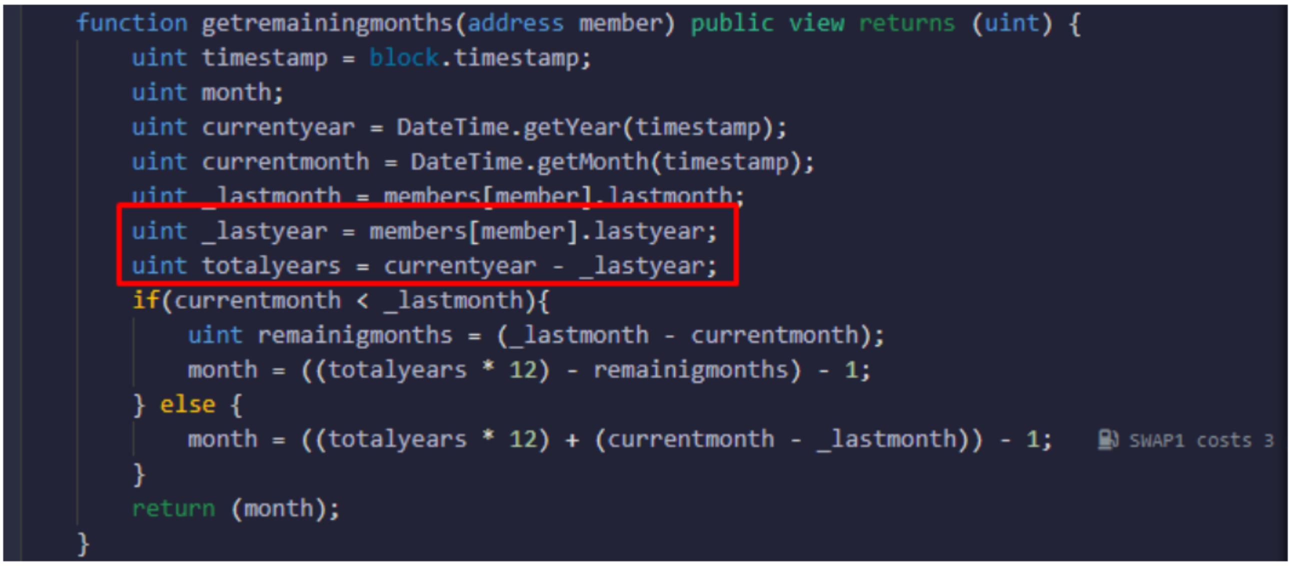
In the getremaining function, the current year is declared as a uint type variable and the getYear function returns uint256 type. This is a mismatched variable type.

Resolution: We suggest keeping the same variable type.

Status: Fixed

(4) Returns the wrong value: **OxyO2CommunityAndTeam.sol**

Function: getremainingmonths(member)



```
function getremainingmonths(address member) public view returns (uint) {
    uint timestamp = block.timestamp;
    uint month;
    uint currentyear = DateTime.getYear(timestamp);
    uint currentmonth = DateTime.getMonth(timestamp);
    uint _lastmonth = members[member].lastmonth;
    uint _lastyear = members[member].lastyear;
    uint totalyears = currentyear - _lastyear;
    if(currentmonth < _lastmonth){
        uint remainigmonths = (_lastmonth - currentmonth);
        month = ((totalyears * 12) - remainigmonths) - 1;
    } else {
        month = ((totalyears * 12) + (currentmonth - _lastmonth)) - 1;
    }
    return (month);
}
```

In the getremainingmonths(member) function, wrong months are returned for the non-existent members.

Resolution: We suggest checking if a member exists or not to avoid this issue.

Status: Fixed

(5) Function input parameters lack a check: **Defi.sol**

Variable validation is not performed in the below functions:

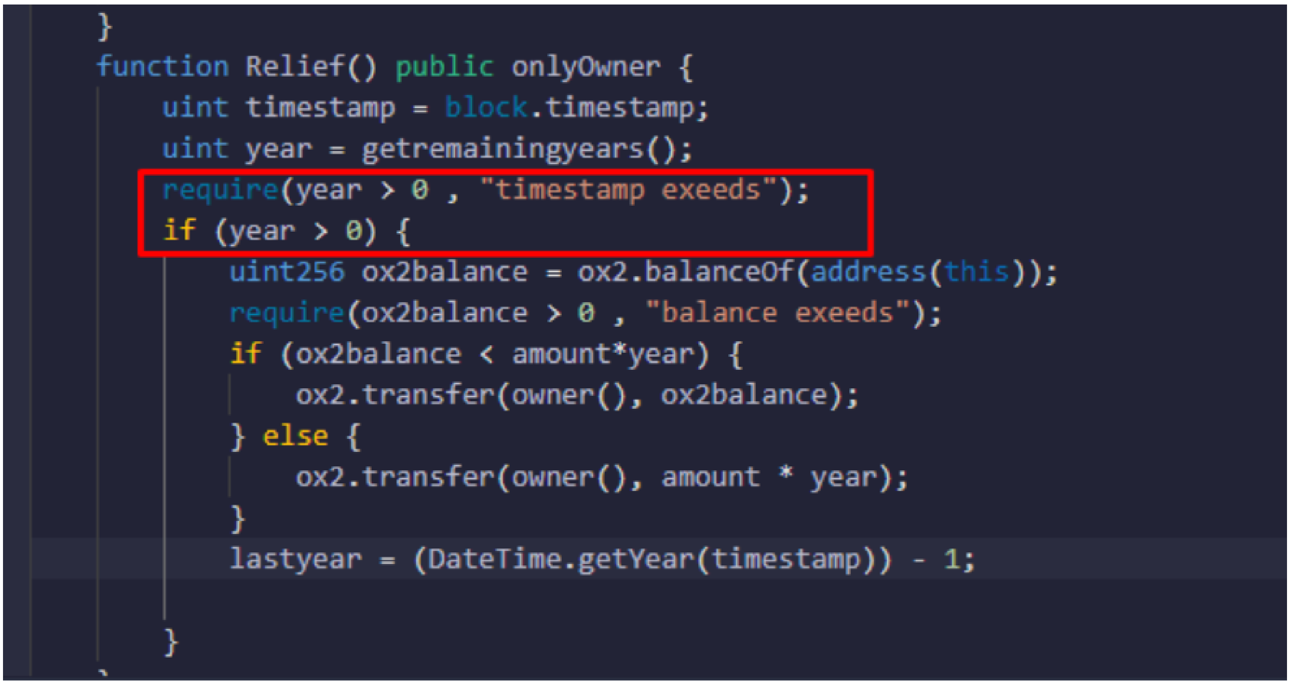
- updateDestToken = _token

Resolution: We advise putting validation: int type variables should not be empty and greater than 0 & address type variables should not be address(0).

Status: Fixed

(6) Perform the same validation:

OxyO2SaveNature.sol : Function: Relief()

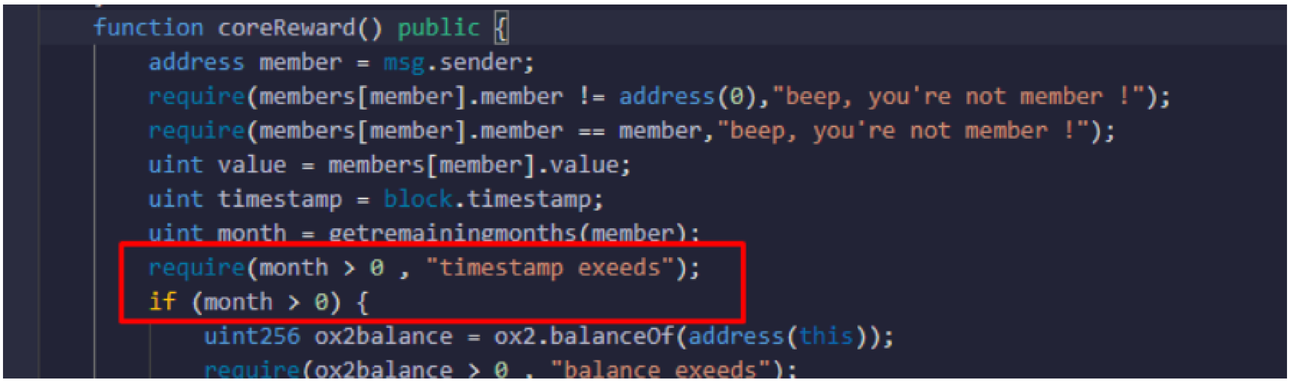


```
}
function Relief() public onlyOwner {
    uint timestamp = block.timestamp;
    uint year = getremainingyears();
    require(year > 0 , "timestamp exeeds");
    if (year > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
        if (ox2balance < amount*year) {
            ox2.transfer(owner(), ox2balance);
        } else {
            ox2.transfer(owner(), amount * year);
        }
        lastyear = (DateTime.getYear(timestamp)) - 1;

    }
```

In the Relief function, if is checking the same condition which is checked by the require just above the if condition. This will increase the gas fee.

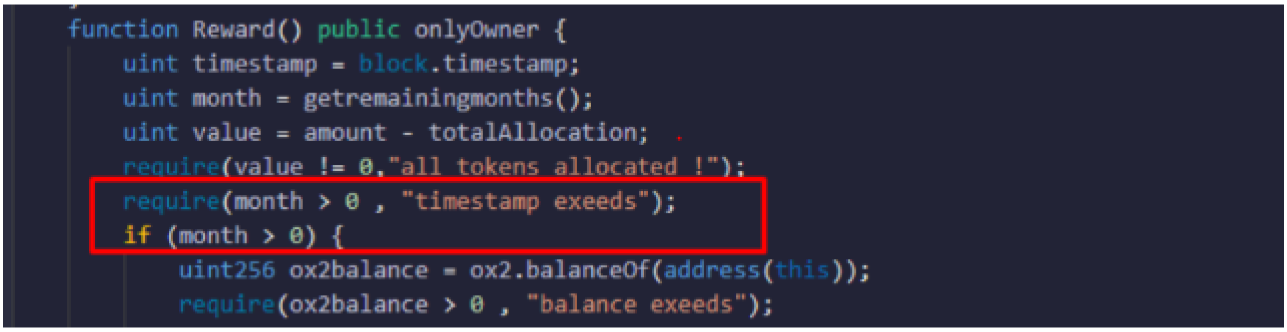
OxyO2CommunityAndTeam.sol

Function: coreReward()



```
function coreReward() public {
    address member = msg.sender;
    require(members[member].member != address(0),"beep, you're not member !");
    require(members[member].member == member,"beep, you're not member !");
    uint value = members[member].value;
    uint timestamp = block.timestamp;
    uint month = getremainingmonths(member);
    require(month > 0 , "timestamp exeeds");
    if (month > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
```

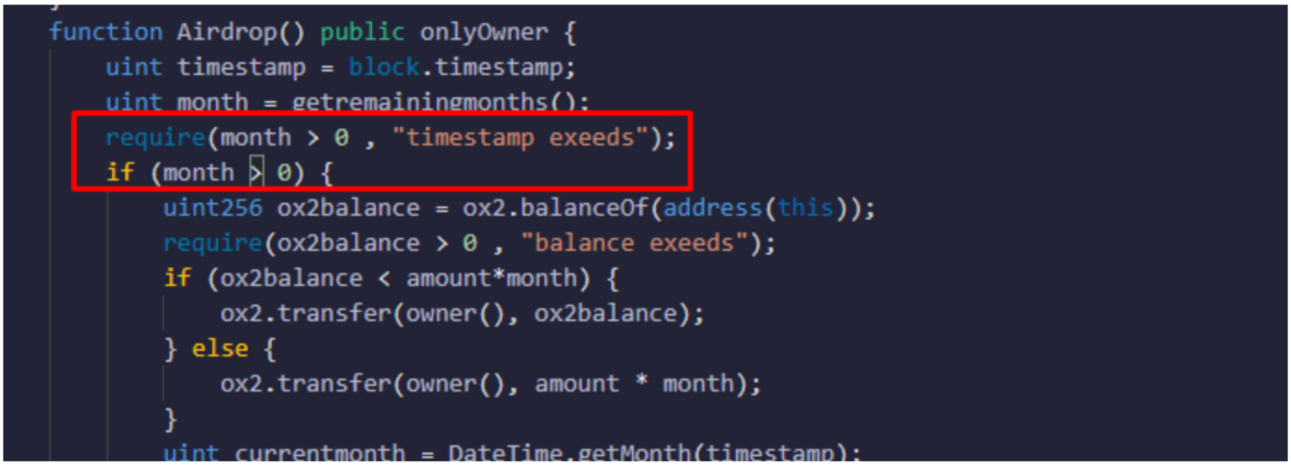
Function: Reward()



```
function Reward() public onlyOwner {
    uint timestamp = block.timestamp;
    uint month = getremainingmonths();
    uint value = amount - totalAllocation;
    require(value != 0,"all tokens allocated !");
    require(month > 0 , "timestamp exeeds");
    if (month > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
```

In the Reward function, if is checking the same condition which is checked by the require just above the if condition. This will increase the gas fee.

OxyO2airdrop.sol: Function: Airdrop()



```
function Airdrop() public onlyOwner {
    uint timestamp = block.timestamp;
    uint month = getremainingmonths();
    require(month > 0 , "timestamp exeeds");
    if (month > 0) {
        uint256 ox2balance = ox2.balanceOf(address(this));
        require(ox2balance > 0 , "balance exeeds");
        if (ox2balance < amount*month) {
            ox2.transfer(owner(), ox2balance);
        } else {
            ox2.transfer(owner(), amount * month);
        }
        uint currentmonth = DateTime.getMonth(timestamp);
```

In the airdrop function, if is checking the same condition which is checked by the require just above the if condition. This will increase the gas fee.

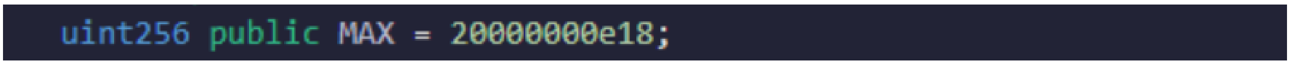
Resolution: We suggest removing the conditions as required. We will handle the same validation.

Status: Fixed

Very Low / Informational / Best practices:

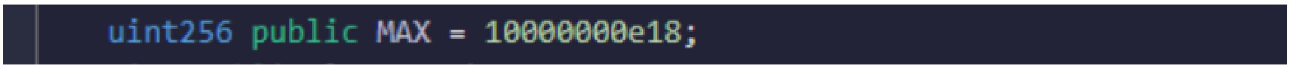
(1) Unused variables, state variables:

OxyO2SaveNature.sol



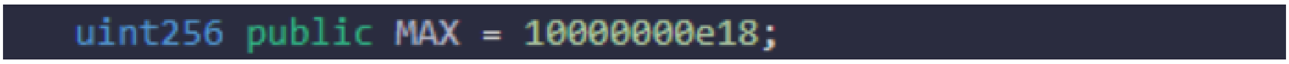
```
uint256 public MAX = 20000000e18;
```

OxyO2CommunityAndTeam.sol



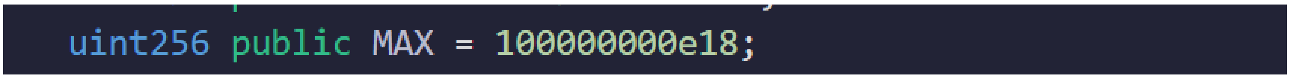
```
uint256 public MAX = 10000000e18;
```

OxyO2airdrop.sol



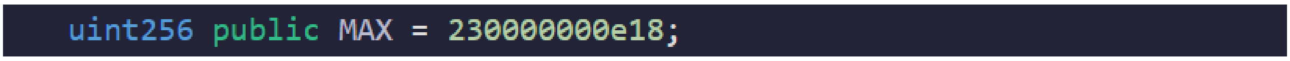
```
uint256 public MAX = 10000000e18;
```

OxyO2Ecosystem.sol



```
uint256 public MAX = 100000000e18;
```

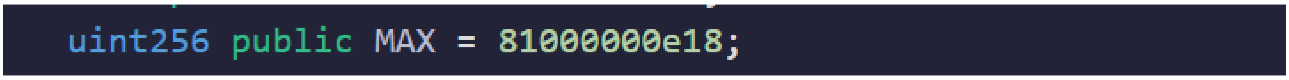
OxyO2publicSale.sol



```
uint256 public MAX = 230000000e18;
```

MAX is defined but not used anywhere.

OxyO2founders.sol, OxyO2founders1.sol, OxyO2founders2.sol, OxyO2Strategic.sol



```
uint256 public MAX = 81000000e18;
```

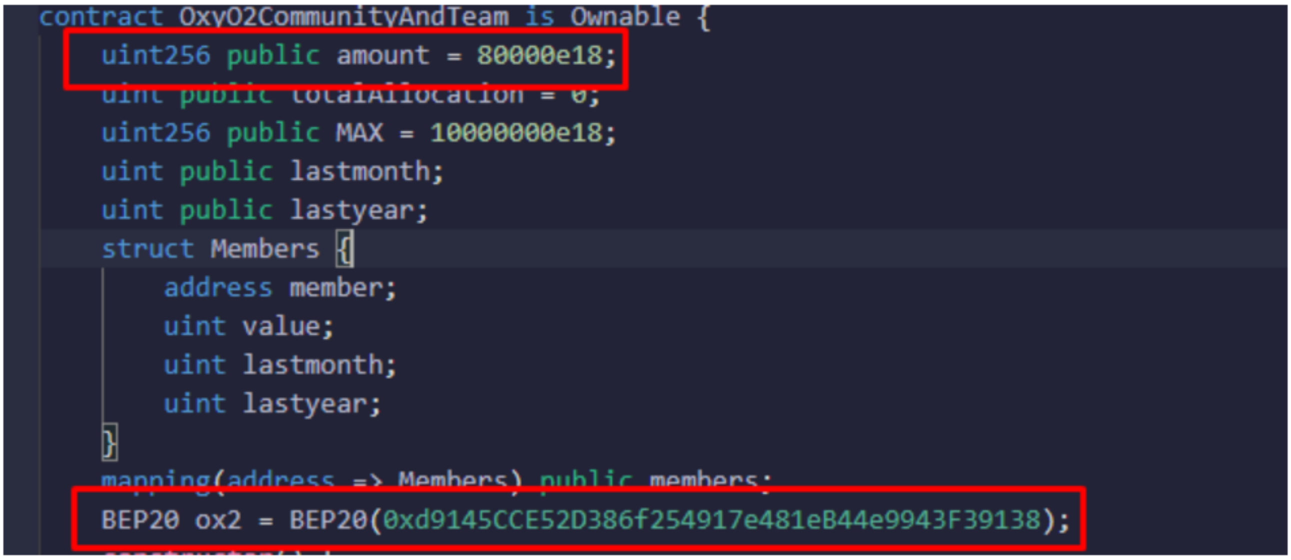
The state variable MAX is declared but never used in the contract.

Resolution: We suggest removing unused variables.

Status: Fixed

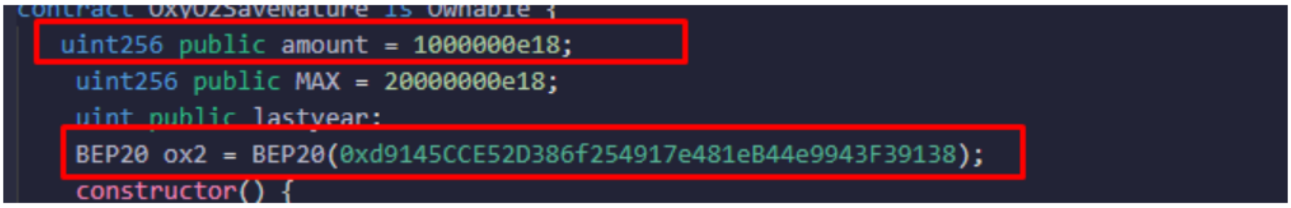
(2) Variables should be constant:

OxyO2CommunityAndTeam.sol



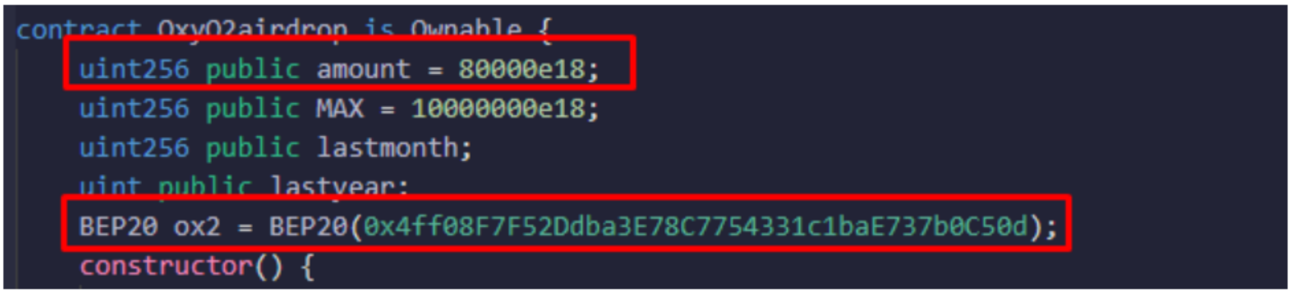
```
contract OxyO2CommunityAndTeam is Ownable {
    uint256 public amount = 80000e18;
    uint public totalAllocation = 0;
    uint256 public MAX = 10000000e18;
    uint public lastmonth;
    uint public lastyear;
    struct Members {
        address member;
        uint value;
        uint lastmonth;
        uint lastyear;
    }
    mapping(address => Members) public members;
    BEP20 ox2 = BEP20(0xd9145CCE52D386f254917e481eB44e9943F39138);
    constructor() {
```

OxyO2SaveNature.sol



```
contract OxyO2SaveNature is Ownable {
    uint256 public amount = 1000000e18;
    uint256 public MAX = 20000000e18;
    uint public lastyear;
    BEP20 ox2 = BEP20(0xd9145CCE52D386f254917e481eB44e9943F39138);
    constructor() {
```

OxyO2airdrop.sol



```
contract OxyO2airdrop is Ownable {
    uint256 public amount = 80000e18;
    uint256 public MAX = 10000000e18;
    uint256 public lastmonth;
    uint public lastyear;
    BEP20 ox2 = BEP20(0x4ff08F7F52Ddba3E78C7754331c1baE737b0C50d);
    constructor() {
```

The ox2, amount variables are not getting changed in the smart contract after deployment.

Resolution: We suggest declaring these variables as constants to save gas.

Status: Fixed

(3) Hardcoded addresses used: **OxyO2SaveNature.sol, OxyO2airdrop.sol, OxyO2CommunityAndTeam.sol**

The OX2 is set by the hardcoded address of the OX2 token contract.

Resolution: We suggest confirming this before deployment.

Status: Fixed

(4) Hardcoded addresses:

OxyO2Strategic.sol

BEP20 token address and DestAddress are hardcoded.

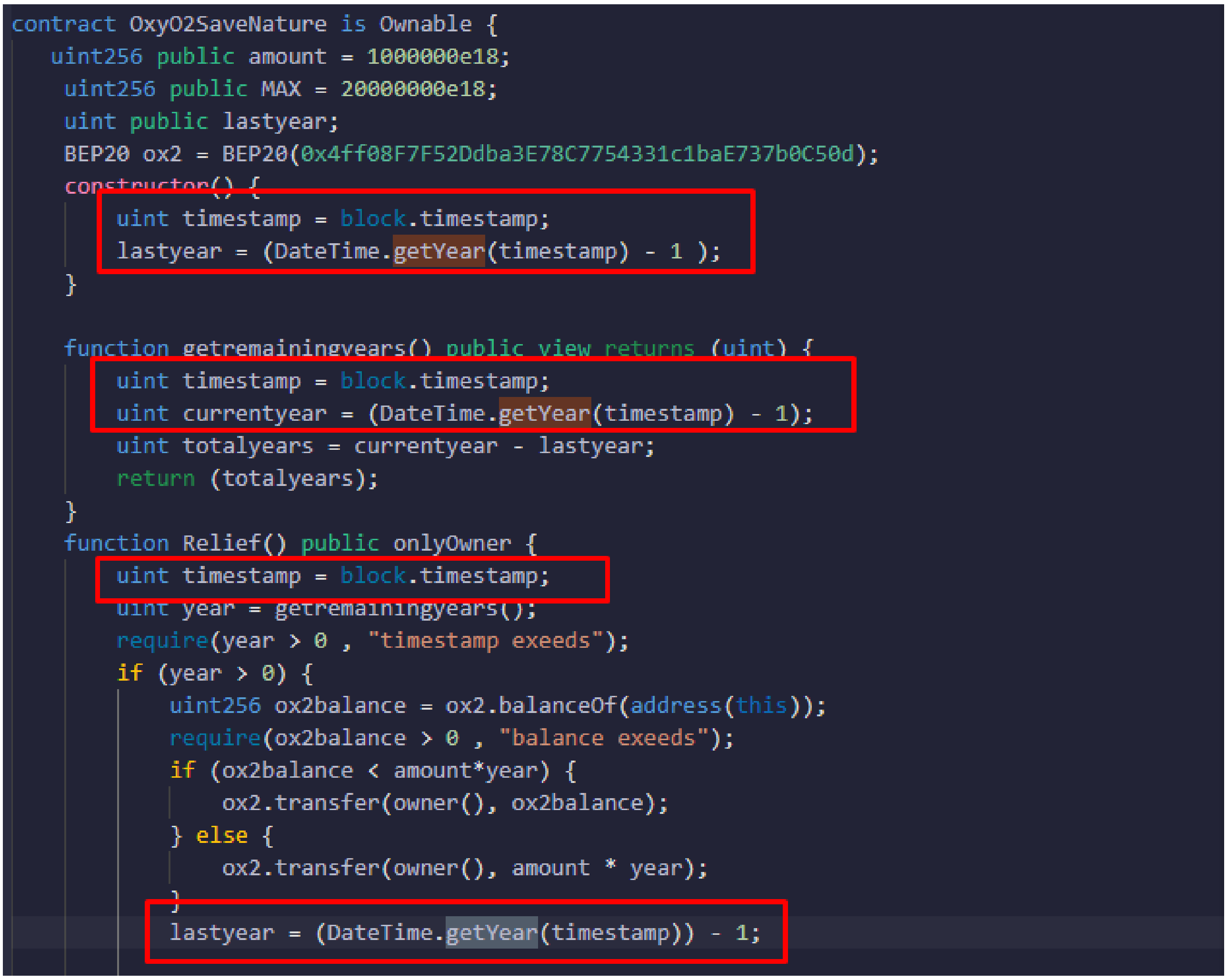
OxyO2Ecosystem.sol, OxyO2publicSale.sol

BEP20 token address is hardcoded.

Resolution: We advise to always ensure the correct address.

Status: Fixed

(5) Different data type: **OxyO2SaveNature.sol**



```
contract OxyO2SaveNature is Ownable {
   uint256 public amount = 1000000e18;
    uint256 public MAX = 20000000e18;
    uint public lastyear;
    BEP20 ox2 = BEP20(0x4ff08F7F52Ddba3E78C7754331c1baE737b0C50d);
    constructor() {
        uint timestamp = block.timestamp;
        lastyear = (DateTime.getYear(timestamp) - 1 );
    }

    function getremainingyears() public view returns (uint) {
        uint timestamp = block.timestamp;
        uint currentyear = (DateTime.getYear(timestamp) - 1);
        uint totalyears = currentyear - lastyear;
        return (totalyears);
    }
    function Relief() public onlyOwner {
        uint timestamp = block.timestamp;
        uint year = getremainingyears();
        require(year > 0 , "timestamp exeeds");
        if (year > 0) {
            uint256 ox2balance = ox2.balanceOf(address(this));
            require(ox2balance > 0 , "balance exeeds");
            if (ox2balance < amount*year) {
                ox2.transfer(owner(), ox2balance);
            } else {
                ox2.transfer(owner(), amount * year);
            }
            lastyear = (DateTime.getYear(timestamp)) - 1;
```

The getYear function has been defined with a parameter of uint256 , but it is used with a different datatype in OxyO2SaveNature.

OxyO2CommunityAndTeam.sol

The getYear function has been defined with parameter of uint256 , but it is used with different datatype in OxyO2CommunityAndTeam.

OxyO2airdrop.sol

The getYear function has been defined with a parameter of uint256, but it is used with a different datatype in OxyO2airdrop.

Resolution: We suggest using the same datatype for the getYear function in OxyO2SaveNature, OxyO2CommunityAndTeam and OxyO2airdrop.

Status: **Fixed**

Centralization

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

Defi.sol

- Defi: Owner can access the defi function.
- updateDestToken: Dest Token address can be set by the owner.

OxyO2airdrop.sol

- Airdrop: Owners can access the airdrop tokens.

OxyO2CommunityAndTeam.sol

- addCoreMember: Core member addresses can be added by the owner.
- subtractCoreMember: Core member addresses can be deleted by the owner.
- Reward: The owner can reward tokens.

OxyO2Ecosystem.sol

- ecosystem: The owner can access the ecosystem.

OxyO2founders.sol

- addPartner: Partner addresses can be added by the owner.
- subtractPartner: Partner addresses can be deleted by the owner.
- founderReward: The owner can receive founder reward tokens.

OxyO2founders1.sol

- addPartner: Partner addresses can be added by the owner.
- subtractPartner: Partner addresses can be deleted by the owner.
- founderReward: The owner can receive founder reward tokens.

OxyO2founders2.sol

- addPartner: Partner addresses can be added by the owner.
- subtractPartner: Partner addresses can be deleted by the owner.
- founderReward: The owner can receive founder reward tokens.

OxyO2Strategic.sol

- PrivateStrategic: Owner can handle private strategy.

OxyO2publicSale.sol

- publicRelease: The owner can release public tokens.

OxyO2SaveNature.sol

- Relief: The owner can use relief tokens.

Ownable.sol

- transferOwnership: Current owner can transfer ownership of the contract to a new account.

To make the smart contract 100% decentralized, we suggest renouncing ownership of the smart contract once its function is completed.

Conclusion

We were given a contract code in the form of a https://bscscan.com web link. And we have used all possible tests based on the given objects as files. We observed 1 high severity issue, 3 medium severity issue, 6 low severity issue and 5 informational severity issues in the smart contracts. We confirm that these issues have been fixed / acknowledged in the revised code. **So, the smart contracts are ready for the mainnet deployment.**

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

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

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

The security state of the reviewed contract, based on the standard audit procedure scope, is **"Secured".**

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review:
In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis:
Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results:

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

Suggested Solutions:

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

Disclaimers

EtherAuthority.io Disclaimer

EtherAuthority team has analyzed this smart contract in accordance with the best industry practices at the date of this report, in relation to: 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).

Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only. We also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

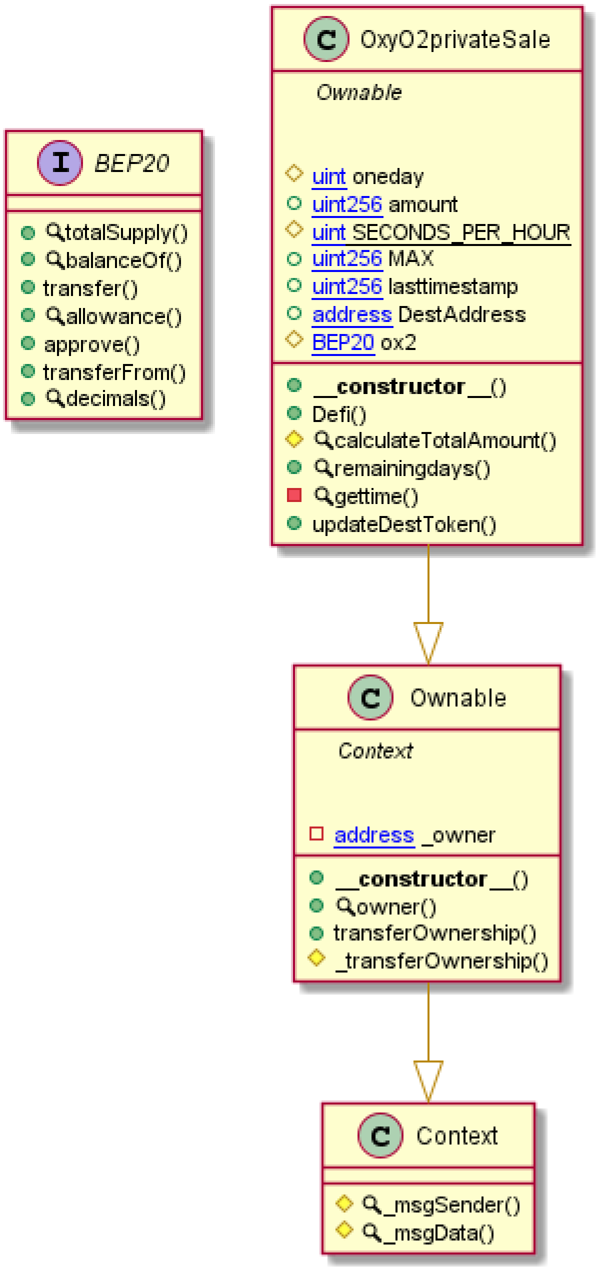
Technical Disclaimer

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

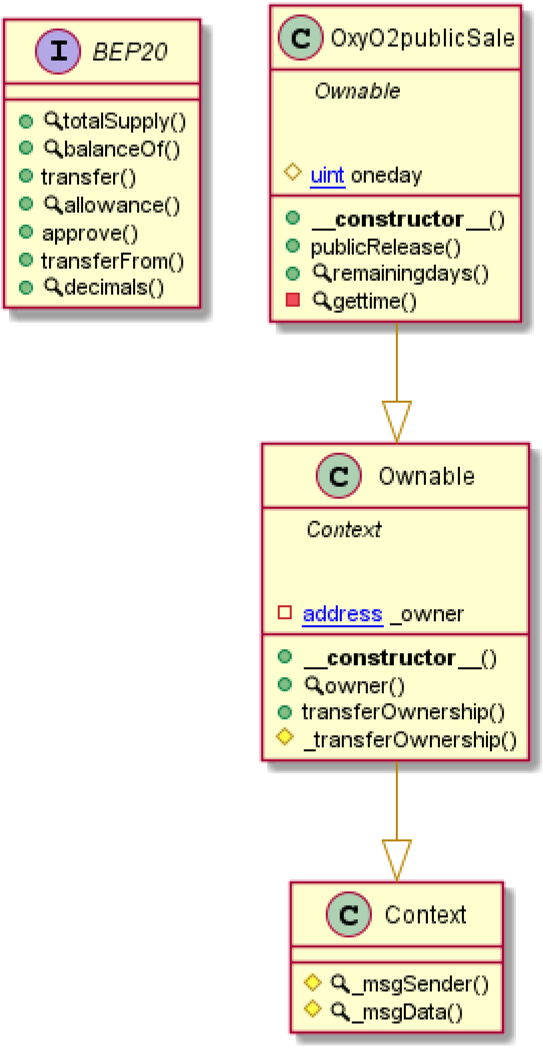
Appendix

Code Flow Diagram - OXYO2 Protocol

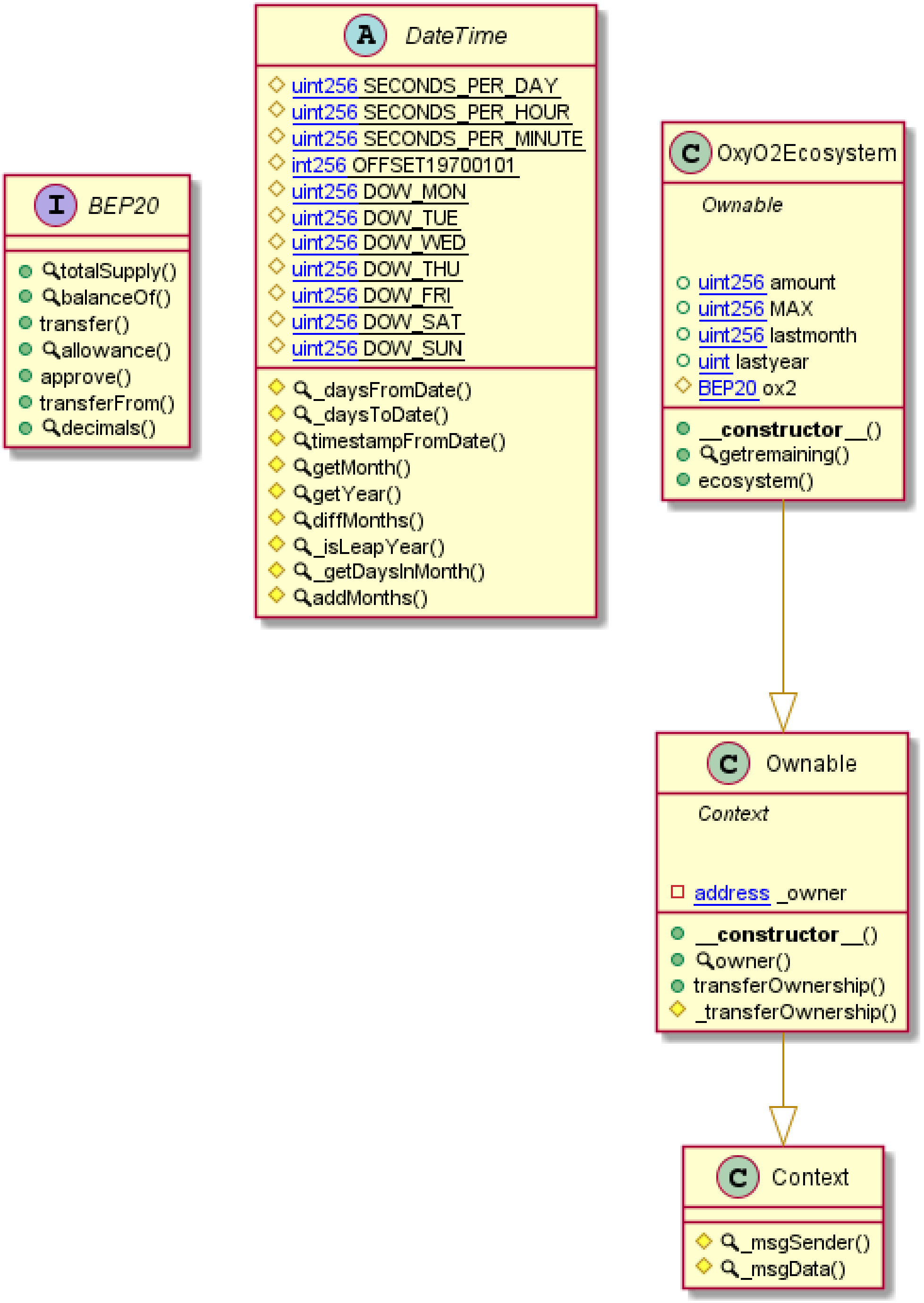
Defi Diagram



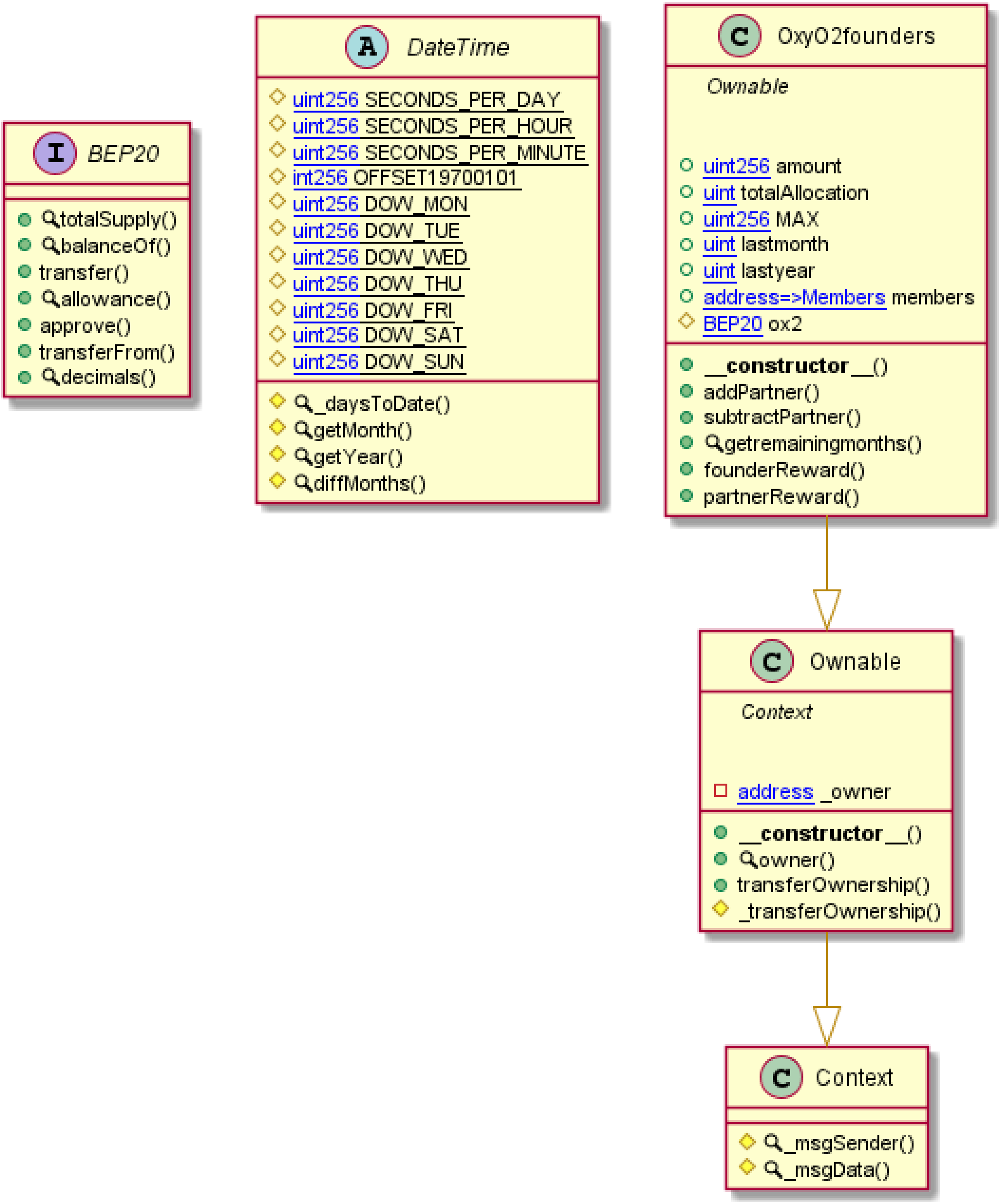
OxyO2publicSale Diagram



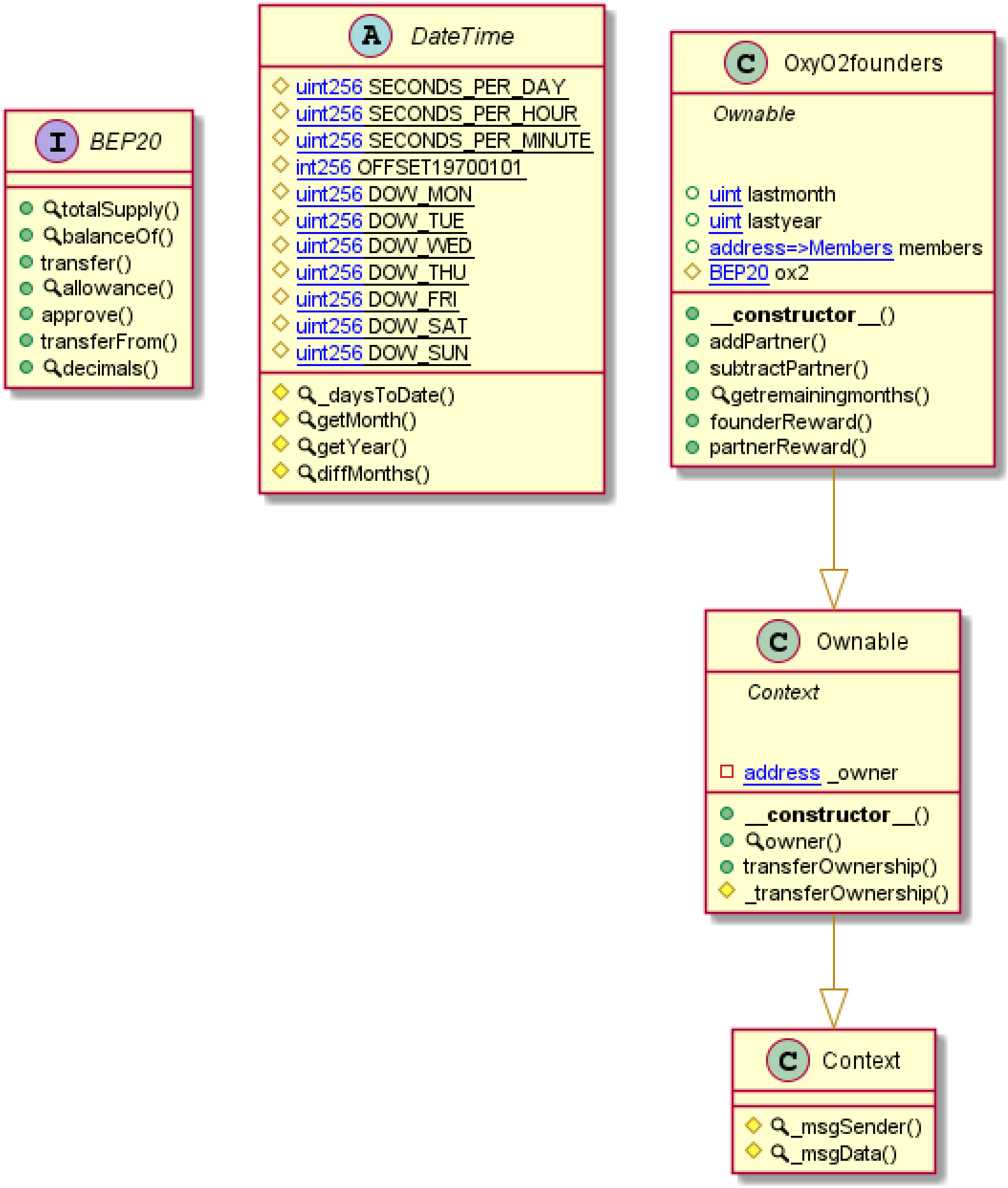
OxyO2Ecosystem Diagram



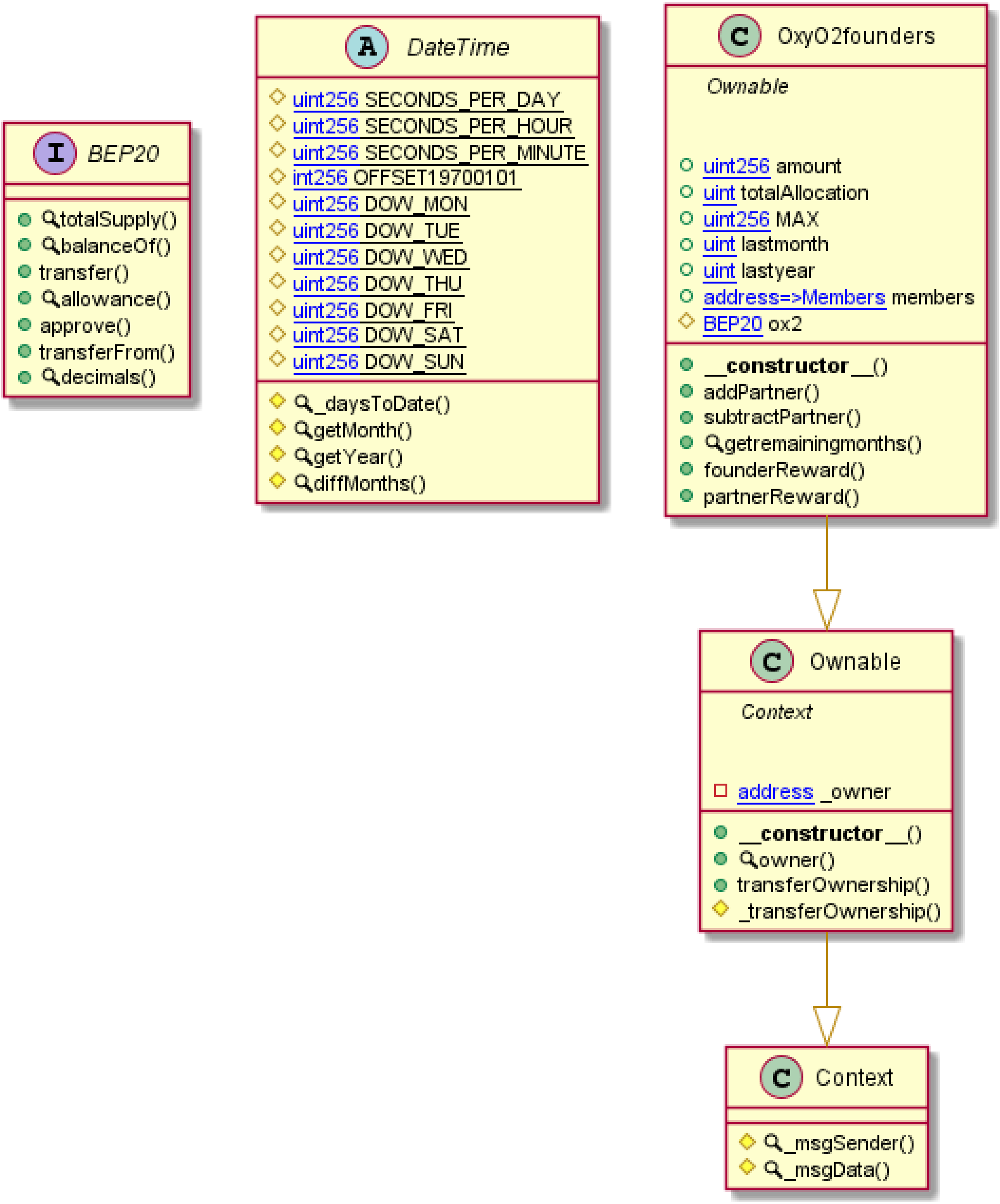
OxyO2founders Diagram



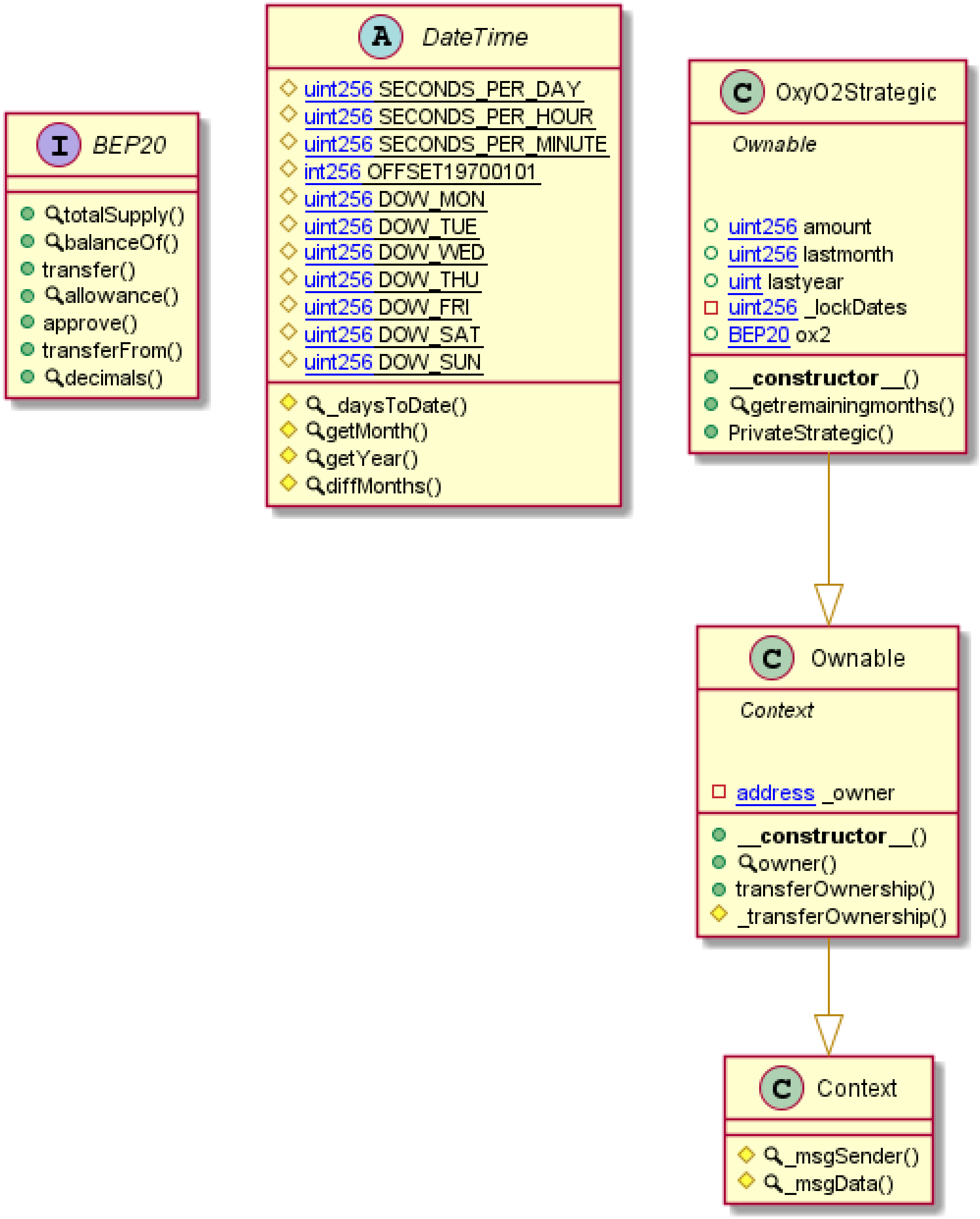
OxyO2founders1 Diagram



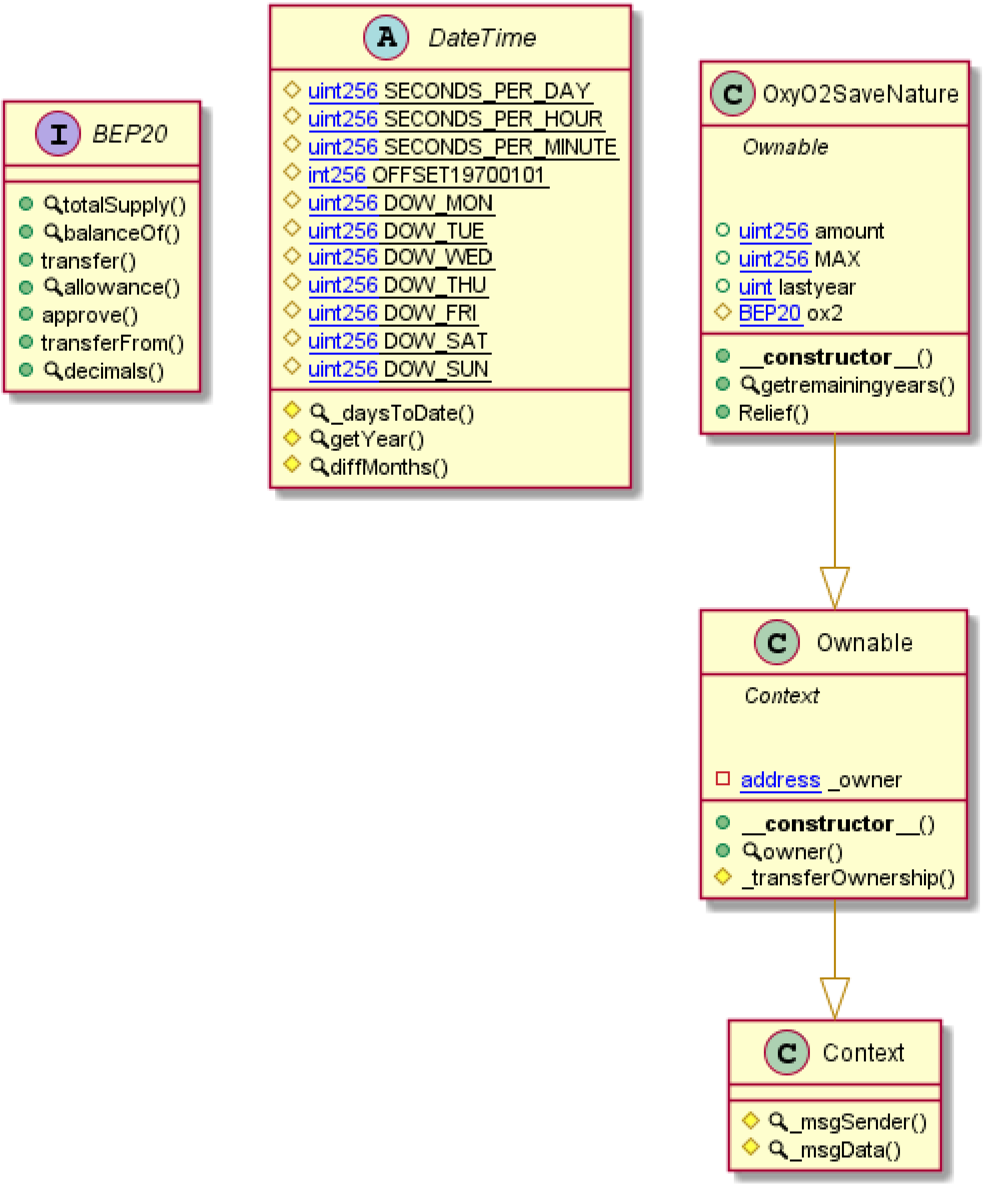
OxyO2founders2 Diagram



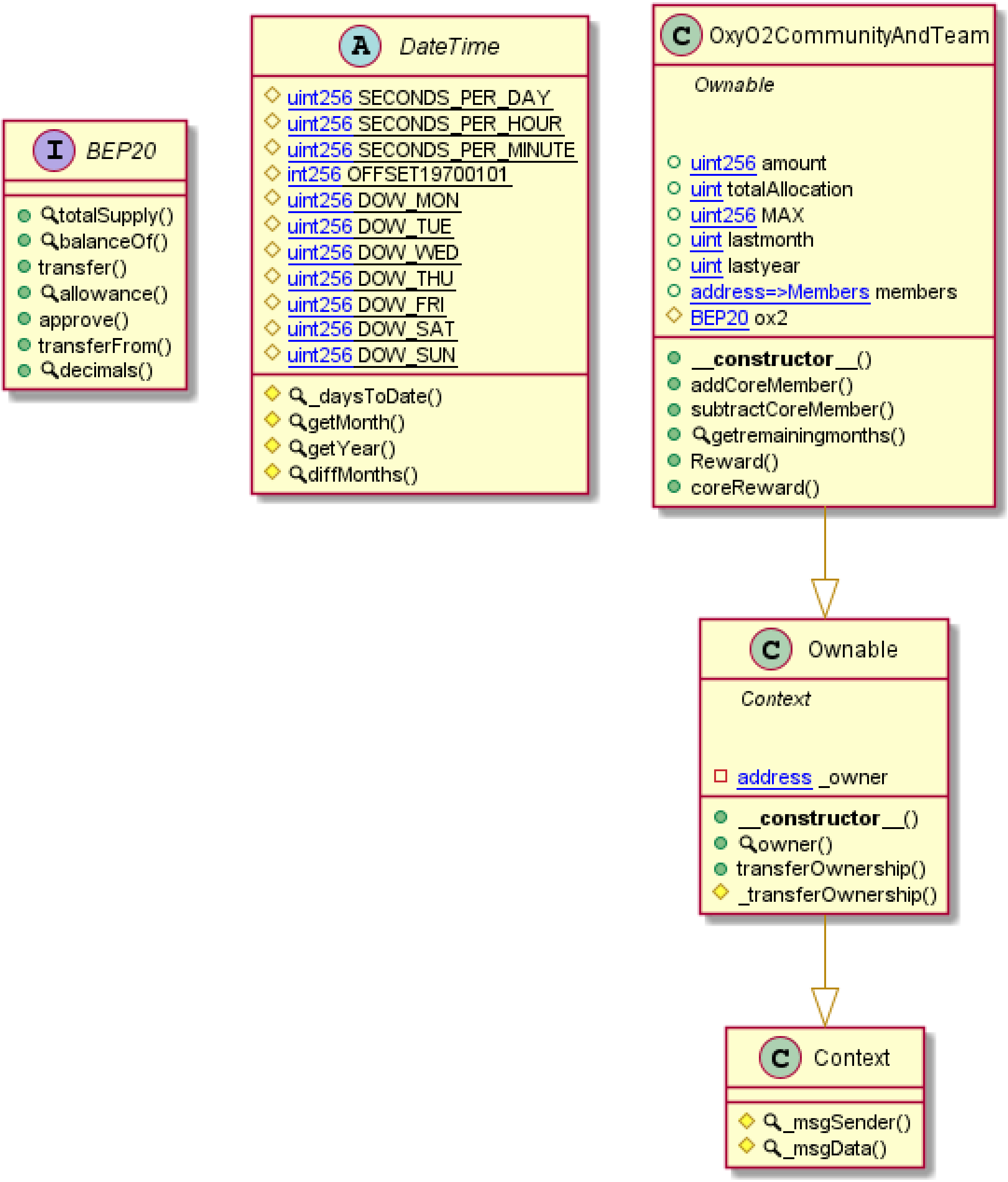
OxyO2Strategic Diagram



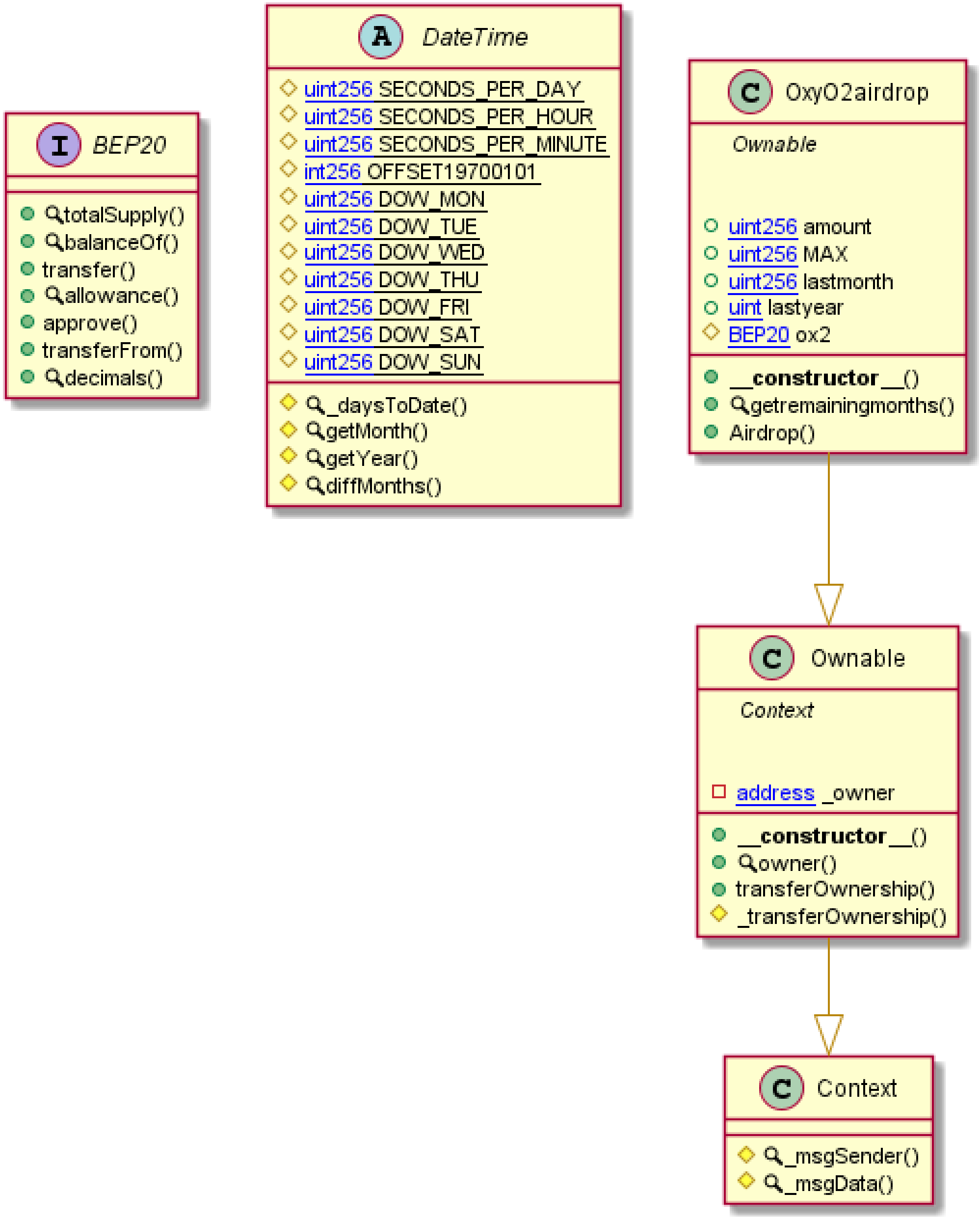
OxyO2SaveNature Diagram



OxyO2CommunityAndTeam Diagram

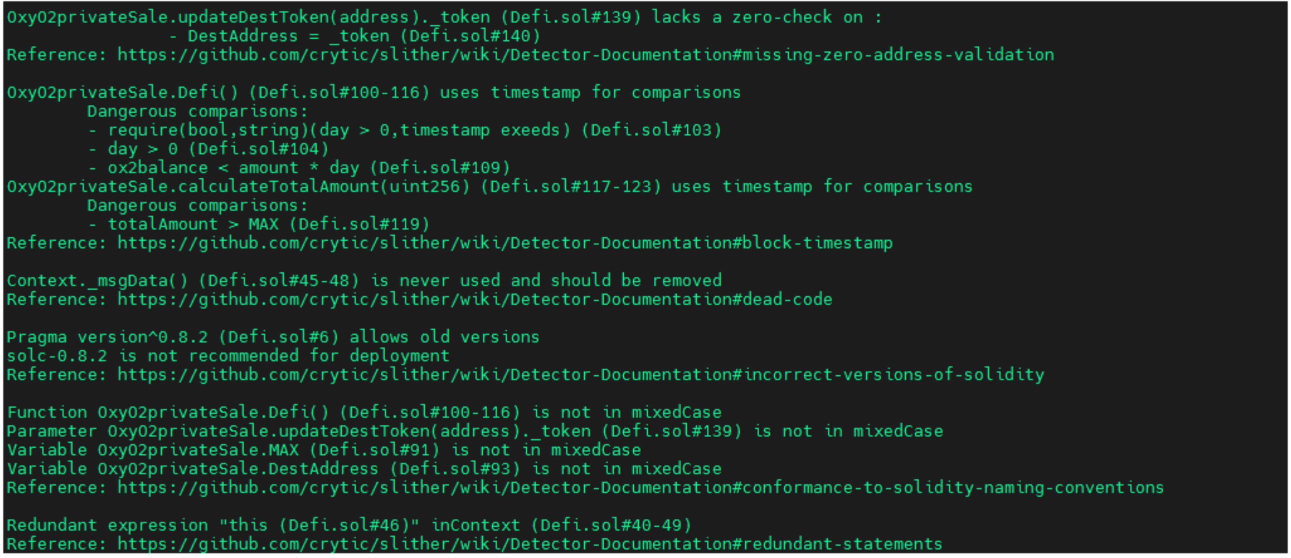


OxyO2airdrop Diagram

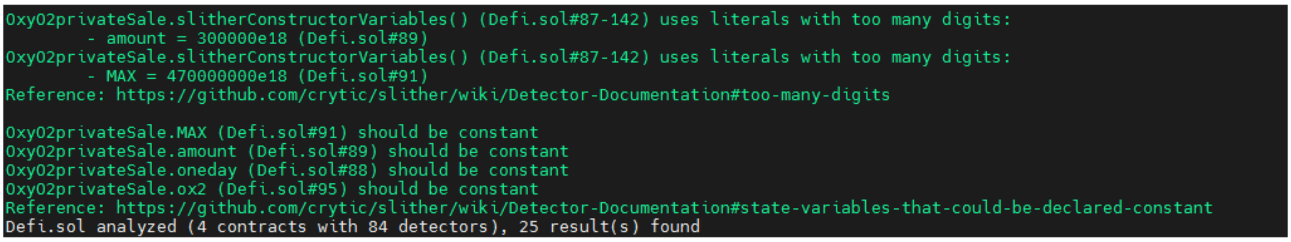


Slither Results Log

Slither log >> Defi.sol

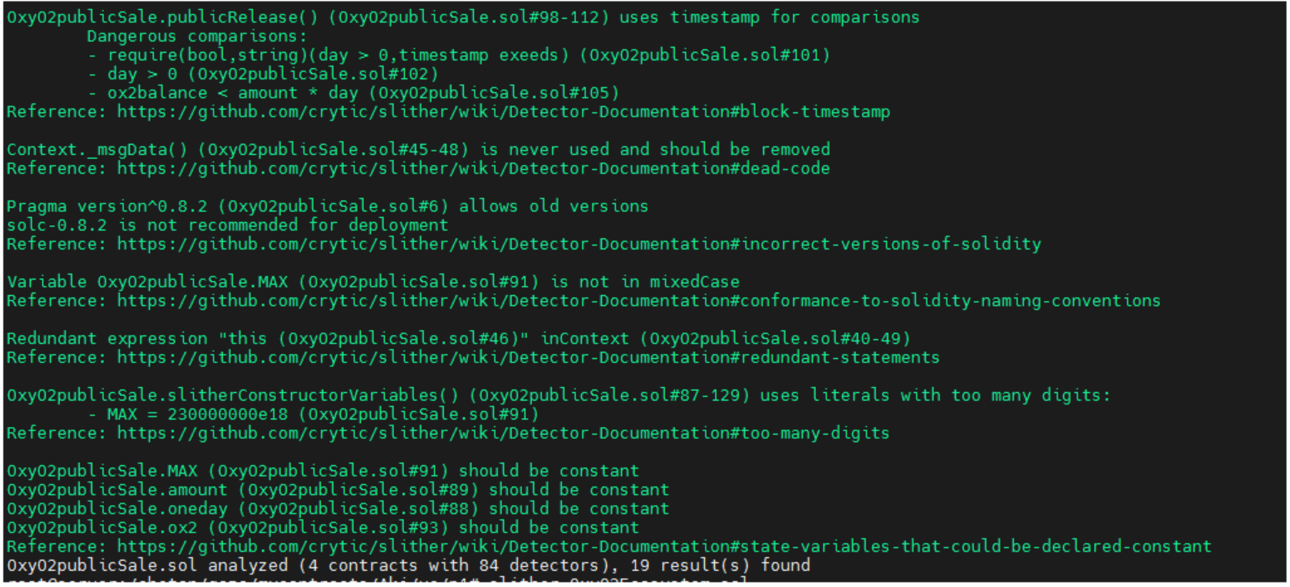


```
OxyO2privateSale.updateDestToken(address)._token (Defi.sol#139) lacks a zero-check on :
                - DestAddress = _token (Defi.sol#140)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

OxyO2privateSale.Defi() (Defi.sol#100-116) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(day > 0,timestamp exeeds) (Defi.sol#103)
        - day > 0 (Defi.sol#104)
        - ox2balance < amount * day (Defi.sol#109)
OxyO2privateSale.calculateTotalAmount(uint256) (Defi.sol#117-123) uses timestamp for comparisons
        Dangerous comparisons:
        - totalAmount > MAX (Defi.sol#119)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (Defi.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (Defi.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function OxyO2privateSale.Defi() (Defi.sol#100-116) is not in mixedCase
Parameter OxyO2privateSale.updateDestToken(address)._token (Defi.sol#139) is not in mixedCase
Variable OxyO2privateSale.MAX (Defi.sol#91) is not in mixedCase
Variable OxyO2privateSale.DestAddress (Defi.sol#93) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Defi.sol#46)" inContext (Defi.sol#40-49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
```



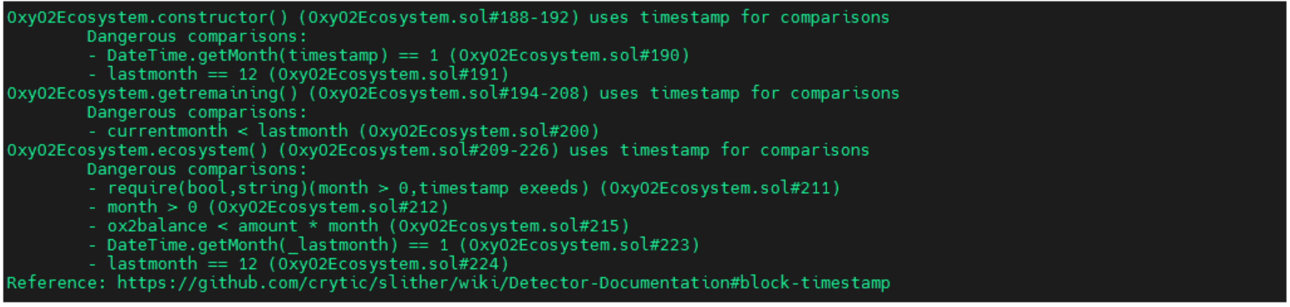
```
OxyO2privateSale.slitherConstructorVariables() (Defi.sol#87-142) uses literals with too many digits:
        - amount = 300000e18 (Defi.sol#89)
OxyO2privateSale.slitherConstructorVariables() (Defi.sol#87-142) uses literals with too many digits:
        - MAX = 470000000e18 (Defi.sol#91)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

OxyO2privateSale.MAX (Defi.sol#91) should be constant
OxyO2privateSale.amount (Defi.sol#89) should be constant
OxyO2privateSale.oneday (Defi.sol#88) should be constant
OxyO2privateSale.ox2 (Defi.sol#95) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
Defi.sol analyzed (4 contracts with 84 detectors), 25 result(s) found
```

Slither log >> OxyO2publicSale.sol

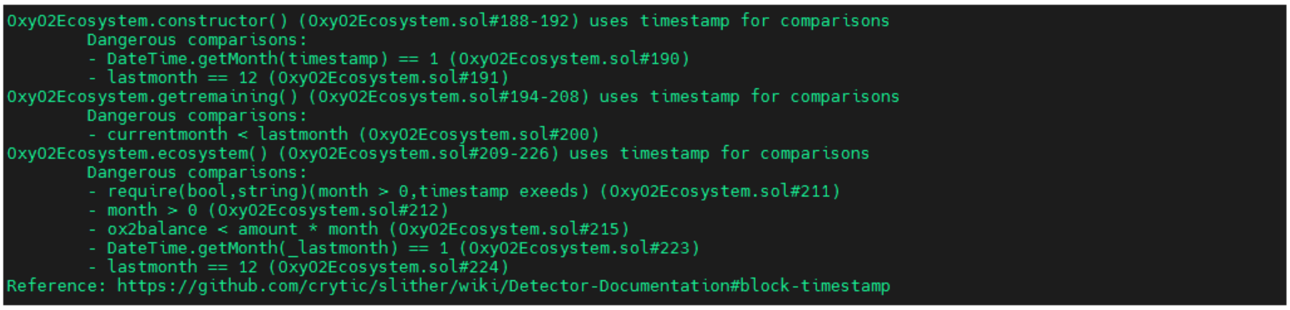


```
OxyO2publicSale.publicRelease() (OxyO2publicSale.sol#98-112) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(day > 0,timestamp exeeds) (OxyO2publicSale.sol#101)
        - day > 0 (OxyO2publicSale.sol#102)
        - ox2balance < amount * day (OxyO2publicSale.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (OxyO2publicSale.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (OxyO2publicSale.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable OxyO2publicSale.MAX (OxyO2publicSale.sol#91) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (OxyO2publicSale.sol#46)" inContext (OxyO2publicSale.sol#40-49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

OxyO2publicSale.slitherConstructorVariables() (OxyO2publicSale.sol#87-129) uses literals with too many digits:
        - MAX = 230000000e18 (OxyO2publicSale.sol#91)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

OxyO2publicSale.MAX (OxyO2publicSale.sol#91) should be constant
OxyO2publicSale.amount (OxyO2publicSale.sol#89) should be constant
OxyO2publicSale.oneday (OxyO2publicSale.sol#88) should be constant
OxyO2publicSale.ox2 (OxyO2publicSale.sol#93) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
OxyO2publicSale.sol analyzed (4 contracts with 84 detectors), 19 result(s) found
```

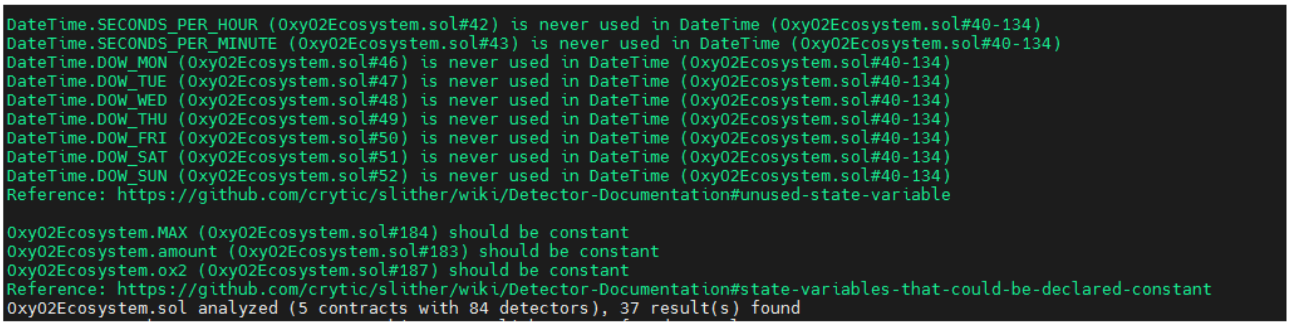
Slither log >> OxyO2Ecosystem.sol



```
OxyO2Ecosystem.constructor() (OxyO2Ecosystem.sol#188-192) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (OxyO2Ecosystem.sol#190)
        - lastmonth == 12 (OxyO2Ecosystem.sol#191)
OxyO2Ecosystem.getremaining() (OxyO2Ecosystem.sol#194-208) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (OxyO2Ecosystem.sol#200)
OxyO2Ecosystem.ecosystem() (OxyO2Ecosystem.sol#209-226) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (OxyO2Ecosystem.sol#211)
        - month > 0 (OxyO2Ecosystem.sol#212)
        - ox2balance < amount * month (OxyO2Ecosystem.sol#215)
        - DateTime.getMonth(_lastmonth) == 1 (OxyO2Ecosystem.sol#223)
        - lastmonth == 12 (OxyO2Ecosystem.sol#224)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

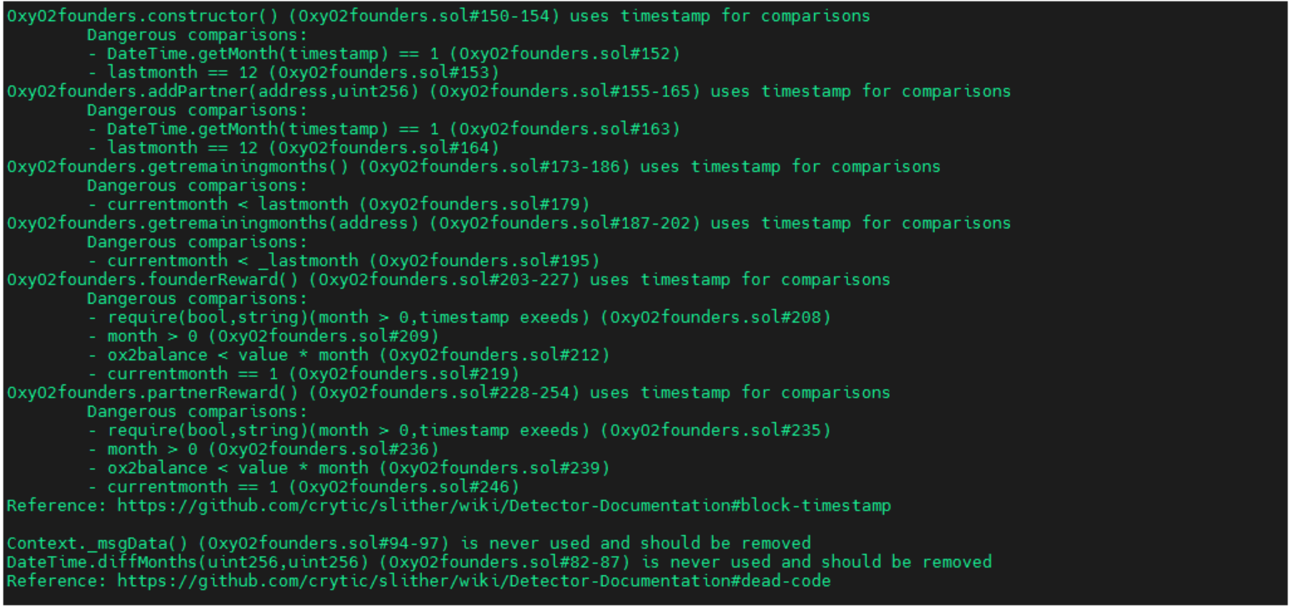


```
Oxy02Ecosystem.constructor() (Oxy02Ecosystem.sol#188-192) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02Ecosystem.sol#190)
        - lastmonth == 12 (Oxy02Ecosystem.sol#191)
Oxy02Ecosystem.getremaining() (Oxy02Ecosystem.sol#194-208) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (Oxy02Ecosystem.sol#200)
Oxy02Ecosystem.ecosystem() (Oxy02Ecosystem.sol#209-226) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02Ecosystem.sol#211)
        - month > 0 (Oxy02Ecosystem.sol#212)
        - ox2balance < amount * month (Oxy02Ecosystem.sol#215)
        - DateTime.getMonth(_lastmonth) == 1 (Oxy02Ecosystem.sol#223)
        - lastmonth == 12 (Oxy02Ecosystem.sol#224)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

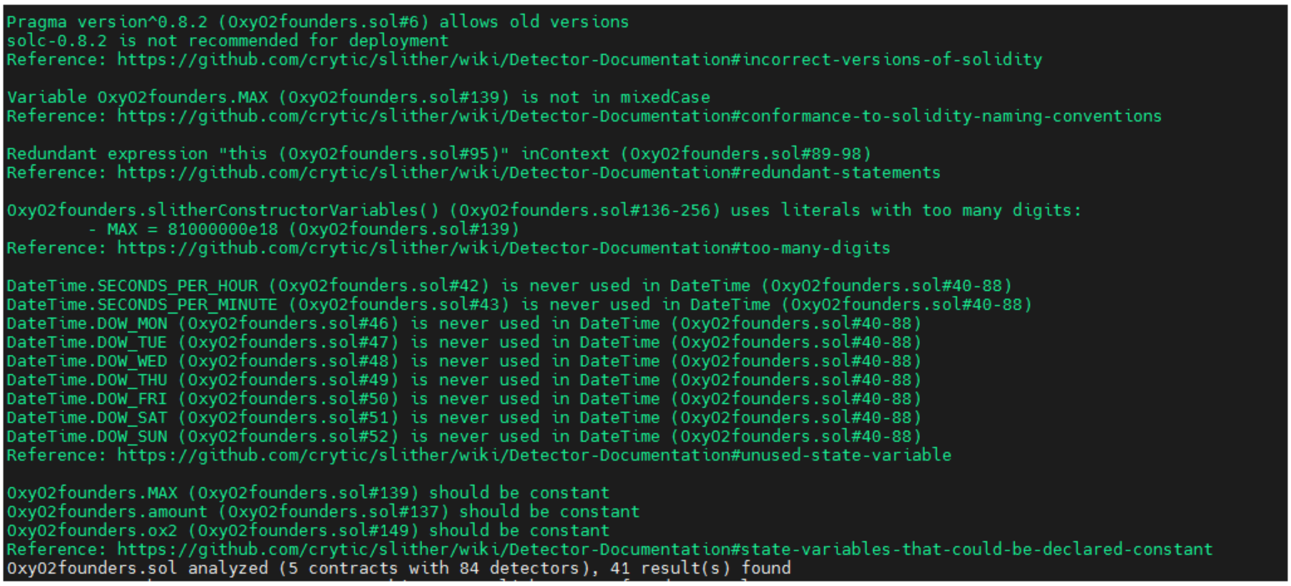


```
DateTime.SECONDS_PER_HOUR (Oxy02Ecosystem.sol#42) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.SECONDS_PER_MINUTE (Oxy02Ecosystem.sol#43) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_MON (Oxy02Ecosystem.sol#46) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_TUE (Oxy02Ecosystem.sol#47) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_WED (Oxy02Ecosystem.sol#48) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_THU (Oxy02Ecosystem.sol#49) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_FRI (Oxy02Ecosystem.sol#50) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_SAT (Oxy02Ecosystem.sol#51) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
DateTime.DOW_SUN (Oxy02Ecosystem.sol#52) is never used in DateTime (Oxy02Ecosystem.sol#40-134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Oxy02Ecosystem.MAX (Oxy02Ecosystem.sol#184) should be constant
Oxy02Ecosystem.amount (Oxy02Ecosystem.sol#183) should be constant
Oxy02Ecosystem.ox2 (Oxy02Ecosystem.sol#187) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
Oxy02Ecosystem.sol analyzed (5 contracts with 84 detectors), 37 result(s) found
```

Slither log >> OxyO2founders.sol

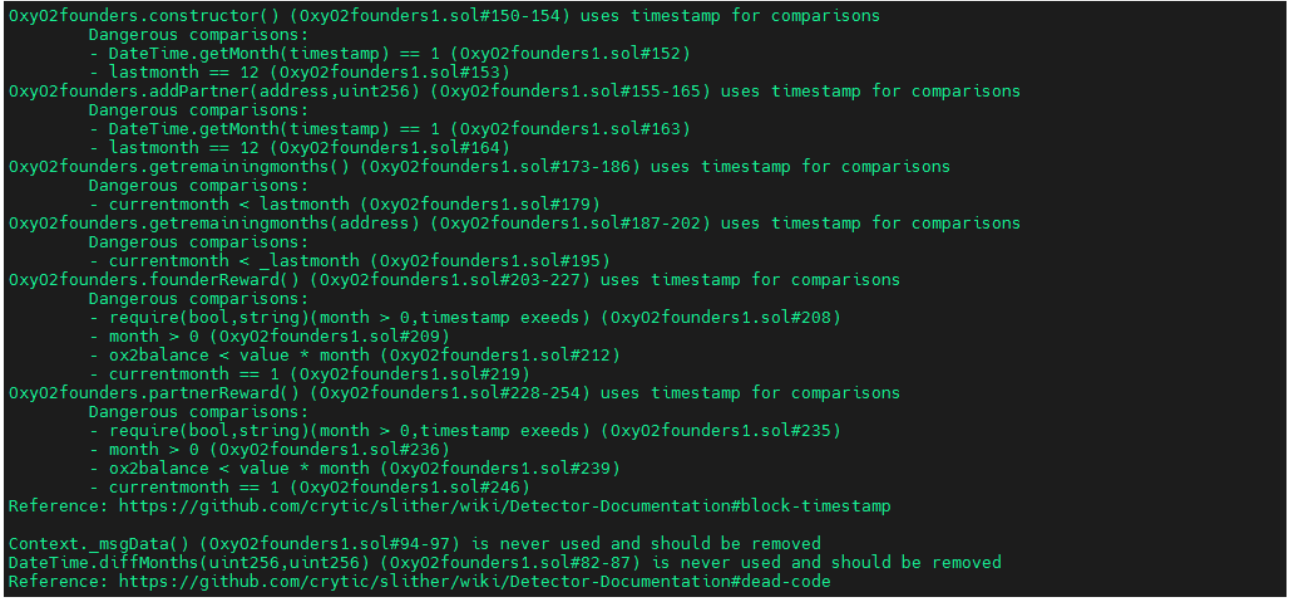


```
Oxy02founders.constructor() (Oxy02founders.sol#150-154) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02founders.sol#152)
        - lastmonth == 12 (Oxy02founders.sol#153)
Oxy02founders.addPartner(address,uint256) (Oxy02founders.sol#155-165) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02founders.sol#163)
        - lastmonth == 12 (Oxy02founders.sol#164)
Oxy02founders.getremainingmonths() (Oxy02founders.sol#173-186) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (Oxy02founders.sol#179)
Oxy02founders.getremainingmonths(address) (Oxy02founders.sol#187-202) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < _lastmonth (Oxy02founders.sol#195)
Oxy02founders.founderReward() (Oxy02founders.sol#203-227) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02founders.sol#208)
        - month > 0 (Oxy02founders.sol#209)
        - ox2balance < value * month (Oxy02founders.sol#212)
        - currentmonth == 1 (Oxy02founders.sol#219)
Oxy02founders.partnerReward() (Oxy02founders.sol#228-254) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02founders.sol#235)
        - month > 0 (Oxy02founders.sol#236)
        - ox2balance < value * month (Oxy02founders.sol#239)
        - currentmonth == 1 (Oxy02founders.sol#246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (Oxy02founders.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (Oxy02founders.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

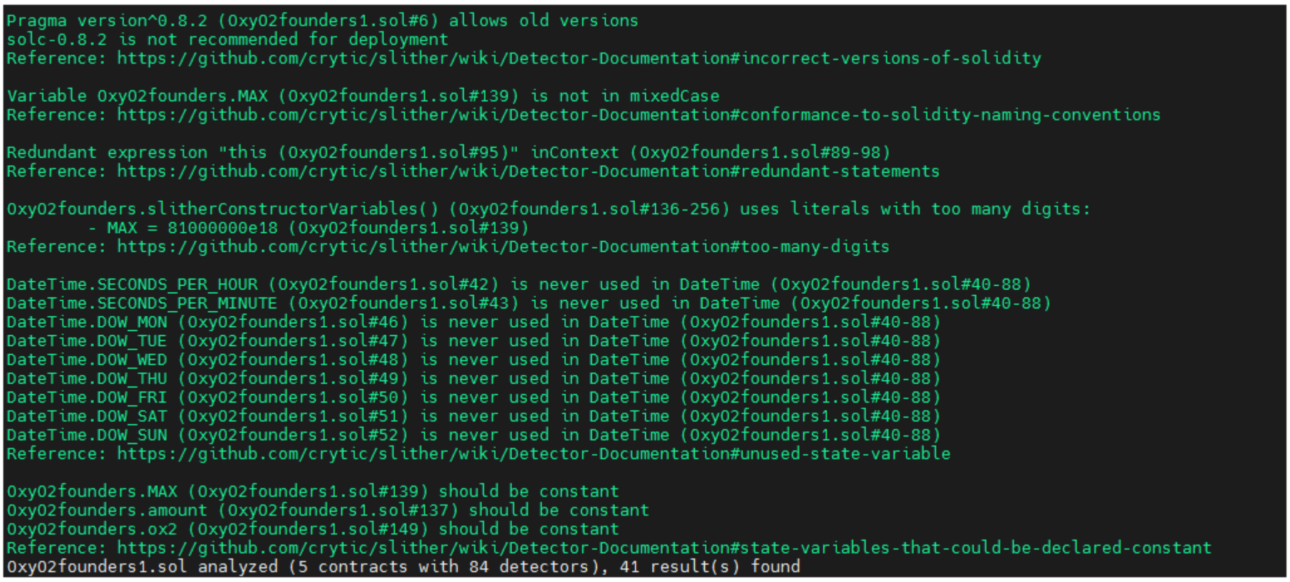


```
Pragma version^0.8.2 (Oxy02founders.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable Oxy02founders.MAX (Oxy02founders.sol#139) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Oxy02founders.sol#95)" inContext (Oxy02founders.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Oxy02founders.slitherConstructorVariables() (Oxy02founders.sol#136-256) uses literals with too many digits:
        - MAX = 81000000e18 (Oxy02founders.sol#139)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DateTime.SECONDS_PER_HOUR (Oxy02founders.sol#42) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.SECONDS_PER_MINUTE (Oxy02founders.sol#43) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_MON (Oxy02founders.sol#46) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_TUE (Oxy02founders.sol#47) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_WED (Oxy02founders.sol#48) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_THU (Oxy02founders.sol#49) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_FRI (Oxy02founders.sol#50) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_SAT (Oxy02founders.sol#51) is never used in DateTime (Oxy02founders.sol#40-88)
DateTime.DOW_SUN (Oxy02founders.sol#52) is never used in DateTime (Oxy02founders.sol#40-88)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Oxy02founders.MAX (Oxy02founders.sol#139) should be constant
Oxy02founders.amount (Oxy02founders.sol#137) should be constant
Oxy02founders.ox2 (Oxy02founders.sol#149) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
Oxy02founders.sol analyzed (5 contracts with 84 detectors), 41 result(s) found
```

Slither log >> OxyO2founders1.sol

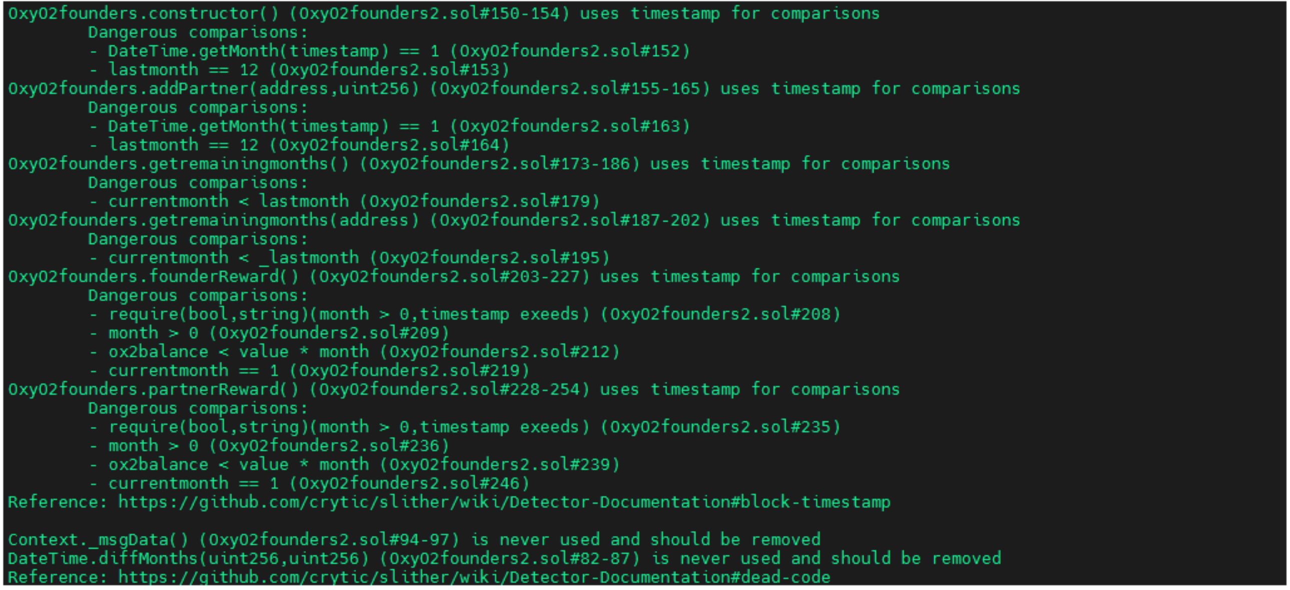


```
Oxy02founders.constructor() (Oxy02founders1.sol#150-154) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02founders1.sol#152)
        - lastmonth == 12 (Oxy02founders1.sol#153)
Oxy02founders.addPartner(address,uint256) (Oxy02founders1.sol#155-165) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02founders1.sol#163)
        - lastmonth == 12 (Oxy02founders1.sol#164)
Oxy02founders.getremainingmonths() (Oxy02founders1.sol#173-186) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (Oxy02founders1.sol#179)
Oxy02founders.getremainingmonths(address) (Oxy02founders1.sol#187-202) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < _lastmonth (Oxy02founders1.sol#195)
Oxy02founders.founderReward() (Oxy02founders1.sol#203-227) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02founders1.sol#208)
        - month > 0 (Oxy02founders1.sol#209)
        - ox2balance < value * month (Oxy02founders1.sol#212)
        - currentmonth == 1 (Oxy02founders1.sol#219)
Oxy02founders.partnerReward() (Oxy02founders1.sol#228-254) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02founders1.sol#235)
        - month > 0 (Oxy02founders1.sol#236)
        - ox2balance < value * month (Oxy02founders1.sol#239)
        - currentmonth == 1 (Oxy02founders1.sol#246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (Oxy02founders1.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (Oxy02founders1.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

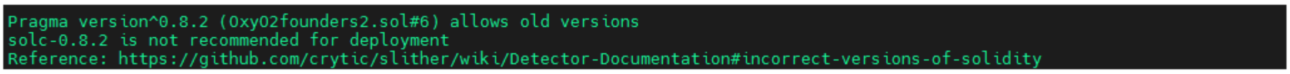


```
Pragma version^0.8.2 (Oxy02founders1.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable Oxy02founders.MAX (Oxy02founders1.sol#139) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Oxy02founders1.sol#95)" inContext (Oxy02founders1.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Oxy02founders.slitherConstructorVariables() (Oxy02founders1.sol#136-256) uses literals with too many digits:
        - MAX = 81000000e18 (Oxy02founders1.sol#139)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DateTime.SECONDS_PER_HOUR (Oxy02founders1.sol#42) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.SECONDS_PER_MINUTE (Oxy02founders1.sol#43) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_MON (Oxy02founders1.sol#46) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_TUE (Oxy02founders1.sol#47) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_WED (Oxy02founders1.sol#48) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_THU (Oxy02founders1.sol#49) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_FRI (Oxy02founders1.sol#50) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_SAT (Oxy02founders1.sol#51) is never used in DateTime (Oxy02founders1.sol#40-88)
DateTime.DOW_SUN (Oxy02founders1.sol#52) is never used in DateTime (Oxy02founders1.sol#40-88)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Oxy02founders.MAX (Oxy02founders1.sol#139) should be constant
Oxy02founders.amount (Oxy02founders1.sol#137) should be constant
Oxy02founders.ox2 (Oxy02founders1.sol#149) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
Oxy02founders1.sol analyzed (5 contracts with 84 detectors), 41 result(s) found
```

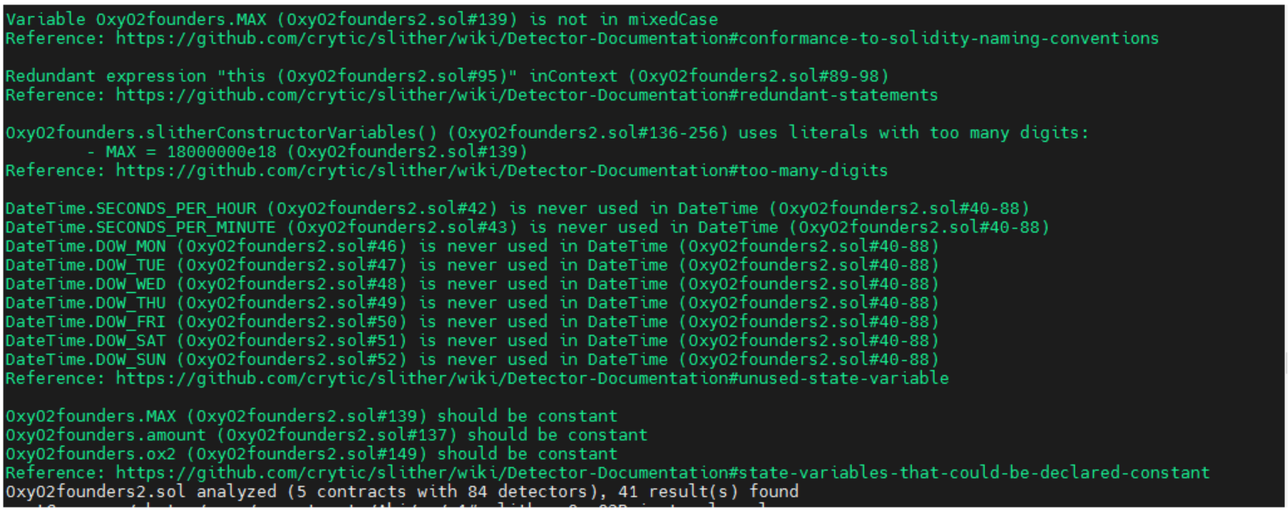
Slither log >> OxyO2founders2.sol



```
Oxy02founders.constructor() (Oxy02founders2.sol#150-154) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02founders2.sol#152)
        - lastmonth == 12 (Oxy02founders2.sol#153)
Oxy02founders.addPartner(address,uint256) (Oxy02founders2.sol#155-165) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (Oxy02founders2.sol#163)
        - lastmonth == 12 (Oxy02founders2.sol#164)
Oxy02founders.getremainingmonths() (Oxy02founders2.sol#173-186) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (Oxy02founders2.sol#179)
Oxy02founders.getremainingmonths(address) (Oxy02founders2.sol#187-202) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < _lastmonth (Oxy02founders2.sol#195)
Oxy02founders.founderReward() (Oxy02founders2.sol#203-227) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02founders2.sol#208)
        - month > 0 (Oxy02founders2.sol#209)
        - ox2balance < value * month (Oxy02founders2.sol#212)
        - currentmonth == 1 (Oxy02founders2.sol#219)
Oxy02founders.partnerReward() (Oxy02founders2.sol#228-254) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02founders2.sol#235)
        - month > 0 (Oxy02founders2.sol#236)
        - ox2balance < value * month (Oxy02founders2.sol#239)
        - currentmonth == 1 (Oxy02founders2.sol#246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (Oxy02founders2.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (Oxy02founders2.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

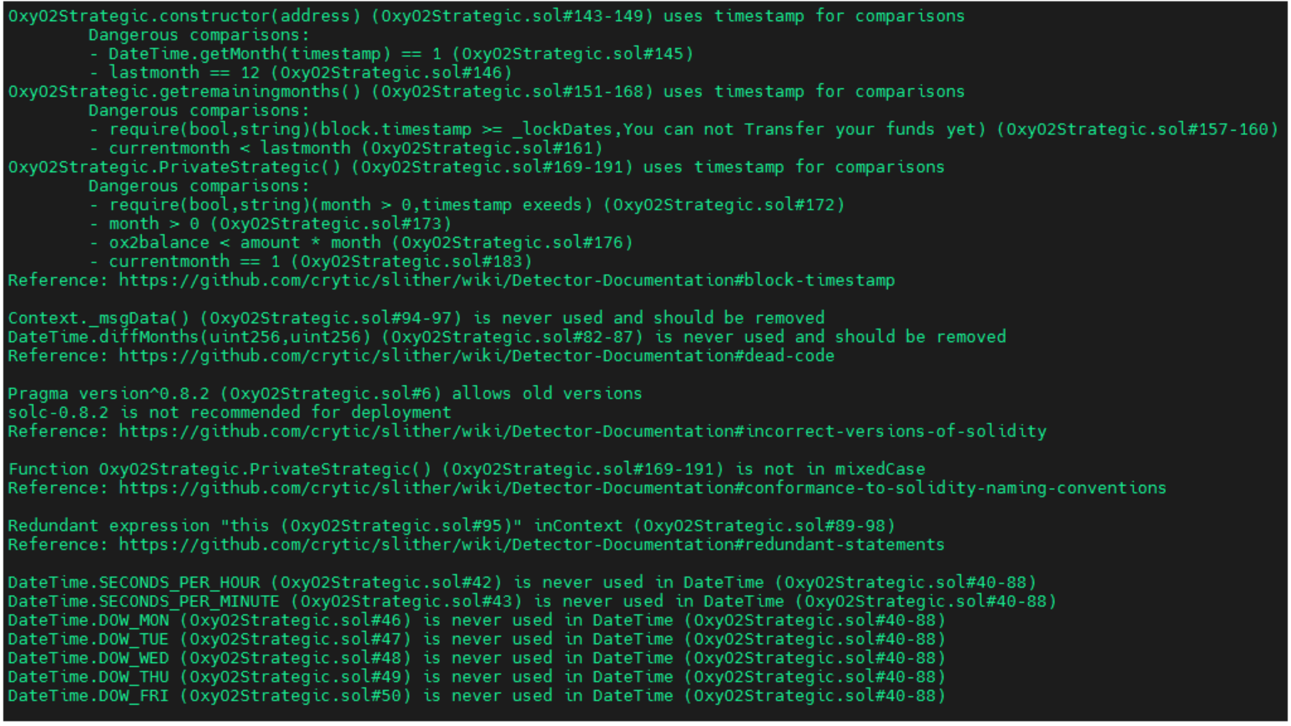


```
Pragma version^0.8.2 (Oxy02founders2.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

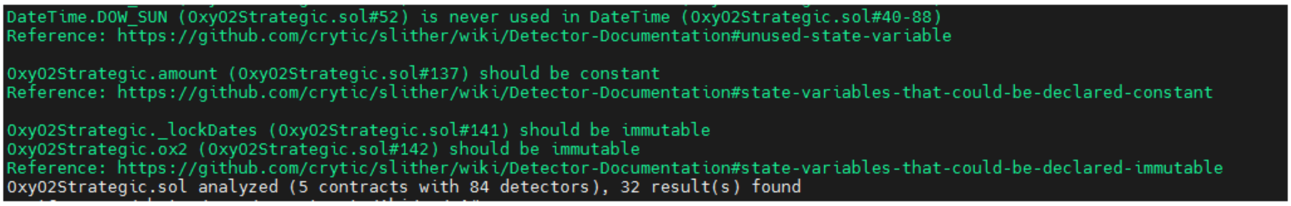


```
Variable OxyO2founders.MAX (OxyO2founders2.sol#139) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (OxyO2founders2.sol#95)" inContext (OxyO2founders2.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

OxyO2founders.slitherConstructorVariables() (OxyO2founders2.sol#136-256) uses literals with too many digits:
        - MAX = 18000000e18 (OxyO2founders2.sol#139)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DateTime.SECONDS_PER_HOUR (OxyO2founders2.sol#42) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.SECONDS_PER_MINUTE (OxyO2founders2.sol#43) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_MON (OxyO2founders2.sol#46) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_TUE (OxyO2founders2.sol#47) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_WED (OxyO2founders2.sol#48) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_THU (OxyO2founders2.sol#49) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_FRI (OxyO2founders2.sol#50) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_SAT (OxyO2founders2.sol#51) is never used in DateTime (OxyO2founders2.sol#40-88)
DateTime.DOW_SUN (OxyO2founders2.sol#52) is never used in DateTime (OxyO2founders2.sol#40-88)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

OxyO2founders.MAX (OxyO2founders2.sol#139) should be constant
OxyO2founders.amount (OxyO2founders2.sol#137) should be constant
OxyO2founders.ox2 (OxyO2founders2.sol#149) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
OxyO2founders2.sol analyzed (5 contracts with 84 detectors), 41 result(s) found
```

Slither log >> OxyO2Strategic.sol

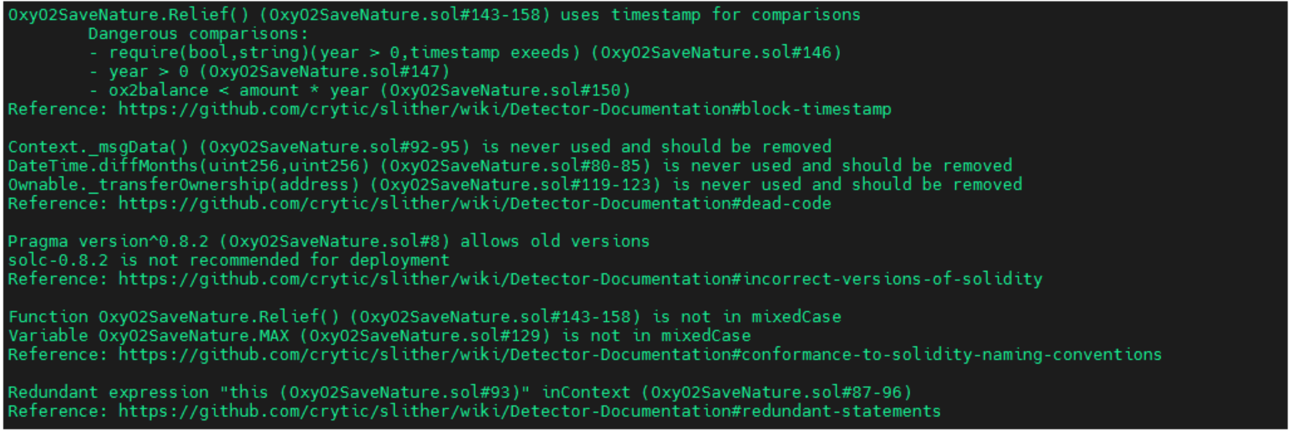


```
OxyO2Strategic.constructor(address) (OxyO2Strategic.sol#143-149) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (OxyO2Strategic.sol#145)
        - lastmonth == 12 (OxyO2Strategic.sol#146)
OxyO2Strategic.getremainingmonths() (OxyO2Strategic.sol#151-168) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= _lockDates,You can not Transfer your funds yet) (OxyO2Strategic.sol#157-160)
        - currentmonth < lastmonth (OxyO2Strategic.sol#161)
OxyO2Strategic.PrivateStrategic() (OxyO2Strategic.sol#169-191) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (OxyO2Strategic.sol#172)
        - month > 0 (OxyO2Strategic.sol#173)
        - ox2balance < amount * month (OxyO2Strategic.sol#176)
        - currentmonth == 1 (OxyO2Strategic.sol#183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (OxyO2Strategic.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (OxyO2Strategic.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (OxyO2Strategic.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function OxyO2Strategic.PrivateStrategic() (OxyO2Strategic.sol#169-191) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (OxyO2Strategic.sol#95)" inContext (OxyO2Strategic.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

DateTime.SECONDS_PER_HOUR (OxyO2Strategic.sol#42) is never used in DateTime (OxyO2Strategic.sol#40-88)
DateTime.SECONDS_PER_MINUTE (OxyO2Strategic.sol#43) is never used in DateTime (OxyO2Strategic.sol#40-88)
DateTime.DOW_MON (OxyO2Strategic.sol#46) is never used in DateTime (OxyO2Strategic.sol#40-88)
DateTime.DOW_TUE (OxyO2Strategic.sol#47) is never used in DateTime (OxyO2Strategic.sol#40-88)
DateTime.DOW_WED (OxyO2Strategic.sol#48) is never used in DateTime (OxyO2Strategic.sol#40-88)
DateTime.DOW_THU (OxyO2Strategic.sol#49) is never used in DateTime (OxyO2Strategic.sol#40-88)
DateTime.DOW_FRI (OxyO2Strategic.sol#50) is never used in DateTime (OxyO2Strategic.sol#40-88)
```

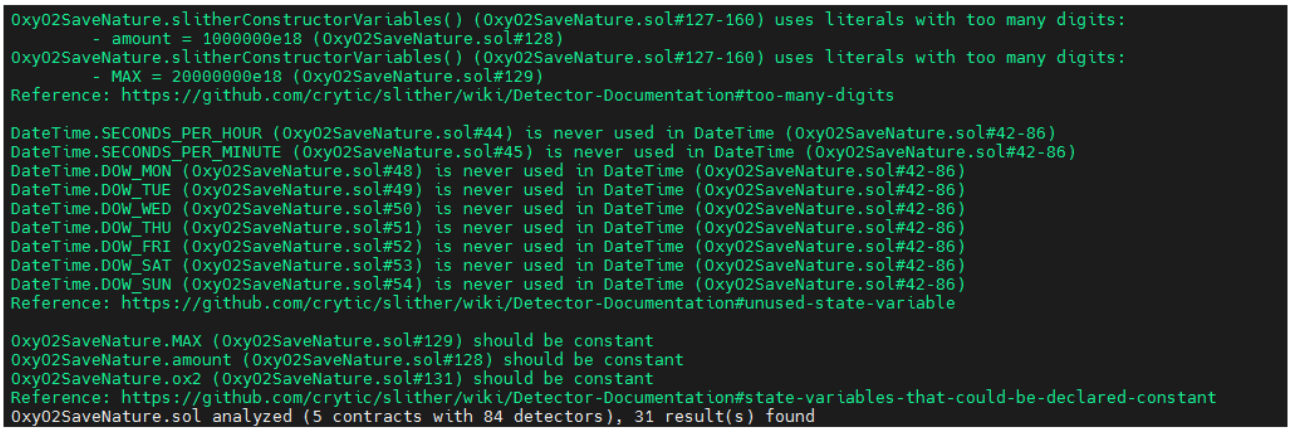


```
DateTime.DOW_SUN (OxyO2Strategic.sol#52) is never used in DateTime (OxyO2Strategic.sol#40-88)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

OxyO2Strategic.amount (OxyO2Strategic.sol#137) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

OxyO2Strategic._lockDates (OxyO2Strategic.sol#141) should be immutable
OxyO2Strategic.ox2 (OxyO2Strategic.sol#142) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
OxyO2Strategic.sol analyzed (5 contracts with 84 detectors), 32 result(s) found
```

Slither log >> OxyO2SaveNature.sol

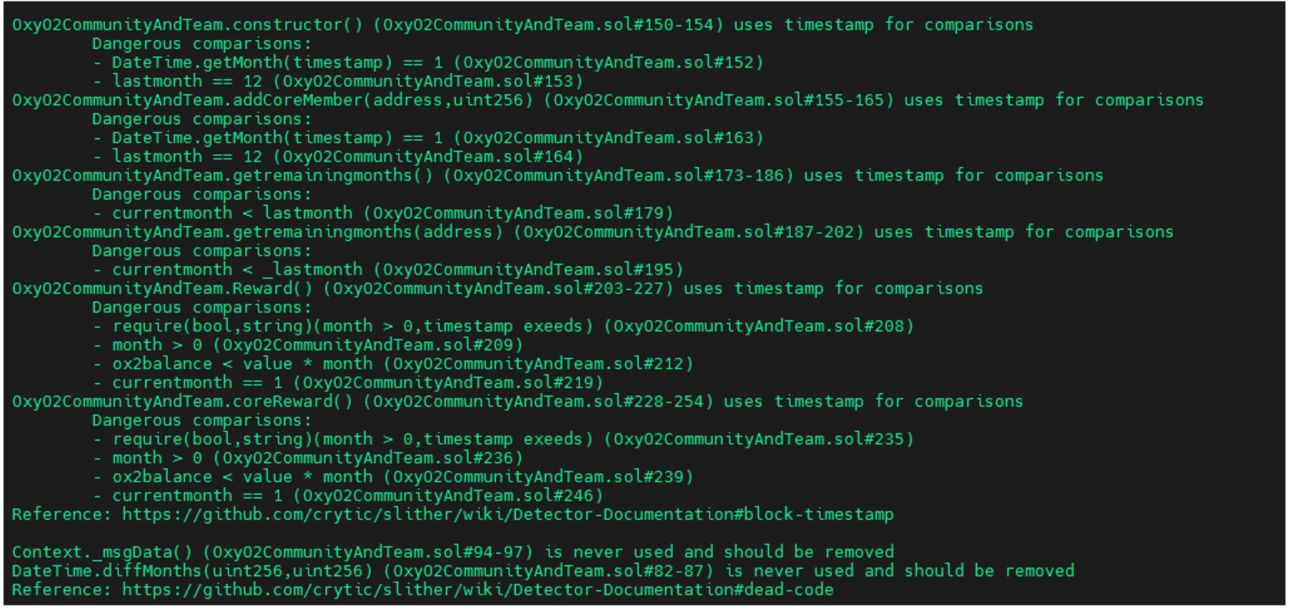


```
OxyO2SaveNature.Relief() (OxyO2SaveNature.sol#143-158) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(year > 0,timestamp exeeds) (OxyO2SaveNature.sol#146)
        - year > 0 (OxyO2SaveNature.sol#147)
        - ox2balance < amount * year (OxyO2SaveNature.sol#150)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (OxyO2SaveNature.sol#92-95) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (OxyO2SaveNature.sol#80-85) is never used and should be removed
Ownable._transferOwnership(address) (OxyO2SaveNature.sol#119-123) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (OxyO2SaveNature.sol#8) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function OxyO2SaveNature.Relief() (OxyO2SaveNature.sol#143-158) is not in mixedCase
Variable OxyO2SaveNature.MAX (OxyO2SaveNature.sol#129) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (OxyO2SaveNature.sol#93)" inContext (OxyO2SaveNature.sol#87-96)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
```

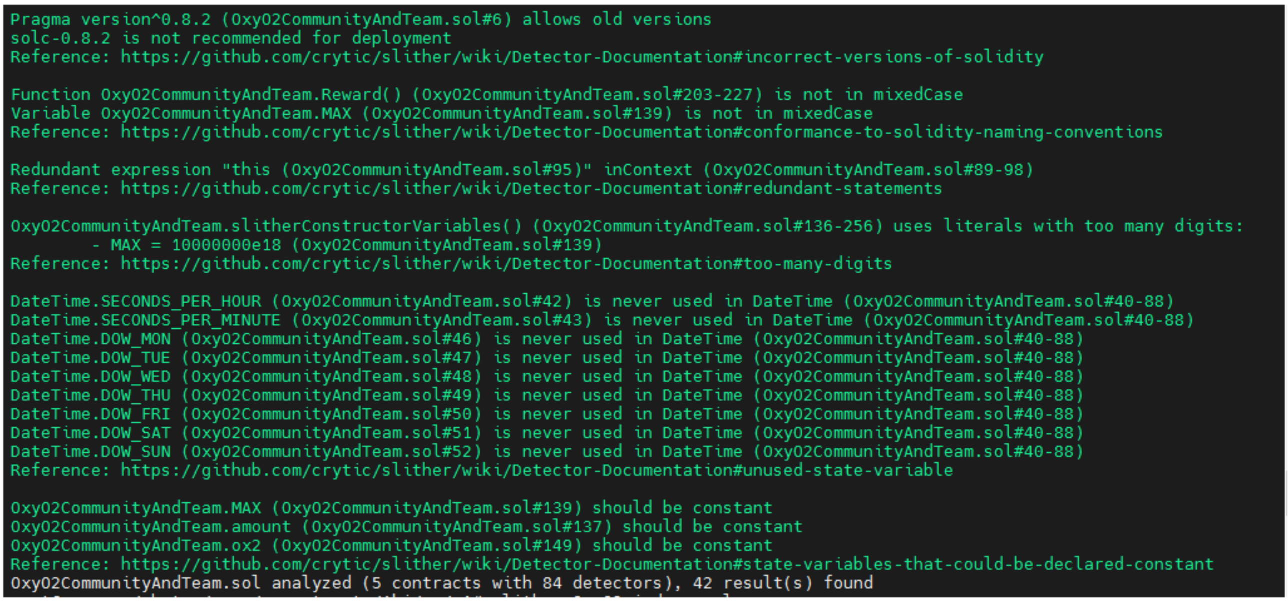


```
OxyO2SaveNature.slitherConstructorVariables() (OxyO2SaveNature.sol#127-160) uses literals with too many digits:
        - amount = 1000000e18 (OxyO2SaveNature.sol#128)
OxyO2SaveNature.slitherConstructorVariables() (OxyO2SaveNature.sol#127-160) uses literals with too many digits:
        - MAX = 20000000e18 (OxyO2SaveNature.sol#129)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DateTime.SECONDS_PER_HOUR (OxyO2SaveNature.sol#44) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.SECONDS_PER_MINUTE (OxyO2SaveNature.sol#45) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_MON (OxyO2SaveNature.sol#48) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_TUE (OxyO2SaveNature.sol#49) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_WED (OxyO2SaveNature.sol#50) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_THU (OxyO2SaveNature.sol#51) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_FRI (OxyO2SaveNature.sol#52) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_SAT (OxyO2SaveNature.sol#53) is never used in DateTime (OxyO2SaveNature.sol#42-86)
DateTime.DOW_SUN (OxyO2SaveNature.sol#54) is never used in DateTime (OxyO2SaveNature.sol#42-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

OxyO2SaveNature.MAX (OxyO2SaveNature.sol#129) should be constant
OxyO2SaveNature.amount (OxyO2SaveNature.sol#128) should be constant
OxyO2SaveNature.ox2 (OxyO2SaveNature.sol#131) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
OxyO2SaveNature.sol analyzed (5 contracts with 84 detectors), 31 result(s) found
```

Slither log >> OxyO2CommunityAndTeam.sol

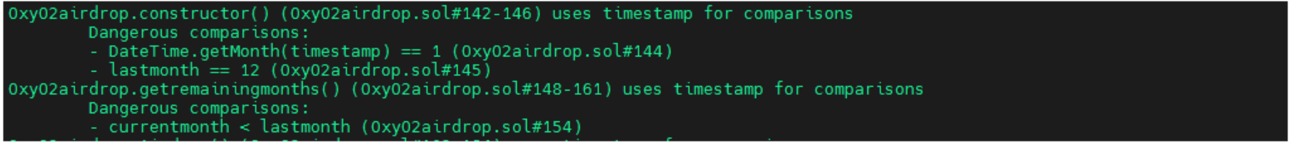


```
OxyO2CommunityAndTeam.constructor() (OxyO2CommunityAndTeam.sol#150-154) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (OxyO2CommunityAndTeam.sol#152)
        - lastmonth == 12 (OxyO2CommunityAndTeam.sol#153)
OxyO2CommunityAndTeam.addCoreMember(address,uint256) (OxyO2CommunityAndTeam.sol#155-165) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (OxyO2CommunityAndTeam.sol#163)
        - lastmonth == 12 (OxyO2CommunityAndTeam.sol#164)
OxyO2CommunityAndTeam.getremainingmonths() (OxyO2CommunityAndTeam.sol#173-186) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (OxyO2CommunityAndTeam.sol#179)
OxyO2CommunityAndTeam.getremainingmonths(address) (OxyO2CommunityAndTeam.sol#187-202) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < _lastmonth (OxyO2CommunityAndTeam.sol#195)
OxyO2CommunityAndTeam.Reward() (OxyO2CommunityAndTeam.sol#203-227) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (OxyO2CommunityAndTeam.sol#208)
        - month > 0 (OxyO2CommunityAndTeam.sol#209)
        - ox2balance < value * month (OxyO2CommunityAndTeam.sol#212)
        - currentmonth == 1 (OxyO2CommunityAndTeam.sol#219)
OxyO2CommunityAndTeam.coreReward() (OxyO2CommunityAndTeam.sol#228-254) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (OxyO2CommunityAndTeam.sol#235)
        - month > 0 (OxyO2CommunityAndTeam.sol#236)
        - ox2balance < value * month (OxyO2CommunityAndTeam.sol#239)
        - currentmonth == 1 (OxyO2CommunityAndTeam.sol#246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (OxyO2CommunityAndTeam.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (OxyO2CommunityAndTeam.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

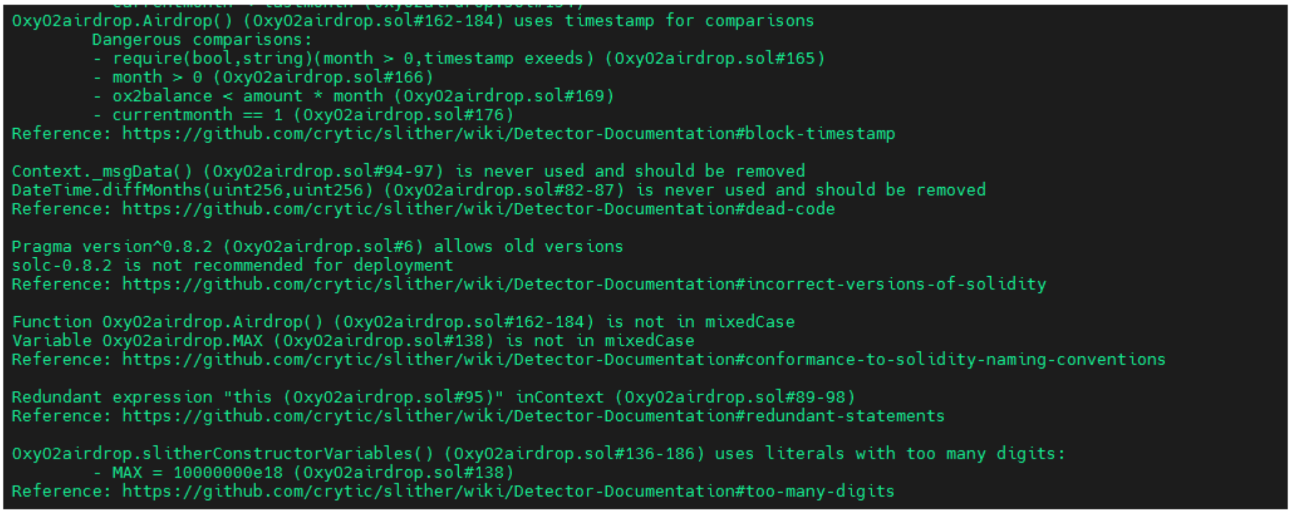


```
Pragma version^0.8.2 (OxyO2CommunityAndTeam.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function OxyO2CommunityAndTeam.Reward() (OxyO2CommunityAndTeam.sol#203-227) is not in mixedCase
Variable OxyO2CommunityAndTeam.MAX (OxyO2CommunityAndTeam.sol#139) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (OxyO2CommunityAndTeam.sol#95)" inContext (OxyO2CommunityAndTeam.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

OxyO2CommunityAndTeam.slitherConstructorVariables() (OxyO2CommunityAndTeam.sol#136-256) uses literals with too many digits:
        - MAX = 10000000e18 (OxyO2CommunityAndTeam.sol#139)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DateTime.SECONDS_PER_HOUR (OxyO2CommunityAndTeam.sol#42) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.SECONDS_PER_MINUTE (OxyO2CommunityAndTeam.sol#43) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_MON (OxyO2CommunityAndTeam.sol#46) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_TUE (OxyO2CommunityAndTeam.sol#47) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_WED (OxyO2CommunityAndTeam.sol#48) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_THU (OxyO2CommunityAndTeam.sol#49) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_FRI (OxyO2CommunityAndTeam.sol#50) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_SAT (OxyO2CommunityAndTeam.sol#51) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
DateTime.DOW_SUN (OxyO2CommunityAndTeam.sol#52) is never used in DateTime (OxyO2CommunityAndTeam.sol#40-88)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

OxyO2CommunityAndTeam.MAX (OxyO2CommunityAndTeam.sol#139) should be constant
OxyO2CommunityAndTeam.amount (OxyO2CommunityAndTeam.sol#137) should be constant
OxyO2CommunityAndTeam.ox2 (OxyO2CommunityAndTeam.sol#149) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
OxyO2CommunityAndTeam.sol analyzed (5 contracts with 84 detectors), 42 result(s) found
```

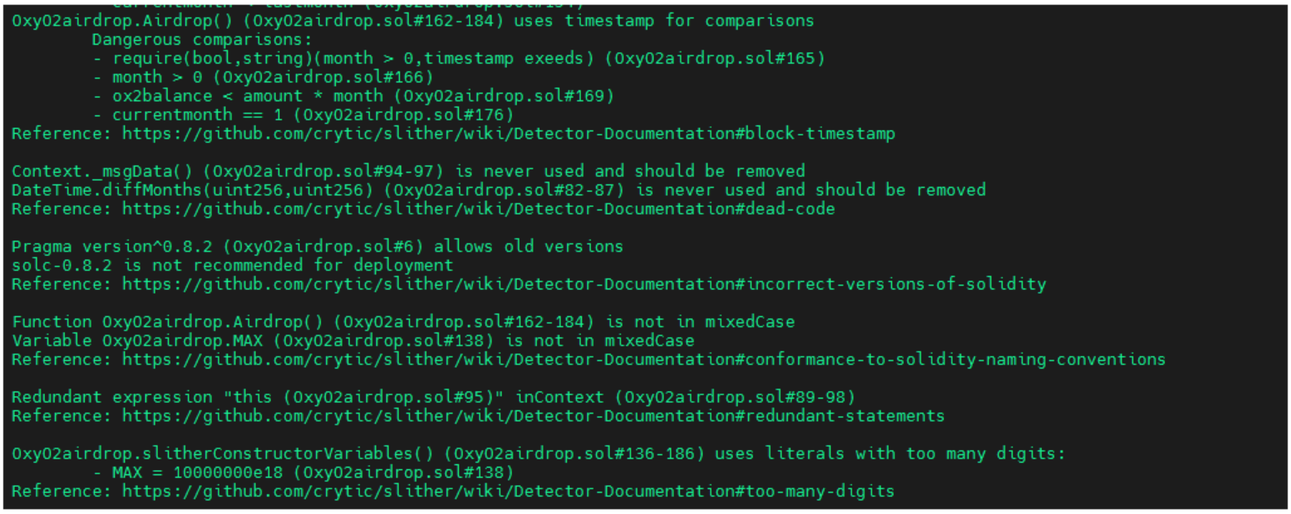
Slither log >> OxyO2airdrop.sol



```
OxyO2airdrop.constructor() (OxyO2airdrop.sol#142-146) uses timestamp for comparisons
        Dangerous comparisons:
        - DateTime.getMonth(timestamp) == 1 (OxyO2airdrop.sol#144)
        - lastmonth == 12 (OxyO2airdrop.sol#145)
OxyO2airdrop.getremainingmonths() (OxyO2airdrop.sol#148-161) uses timestamp for comparisons
        Dangerous comparisons:
        - currentmonth < lastmonth (OxyO2airdrop.sol#154)
```



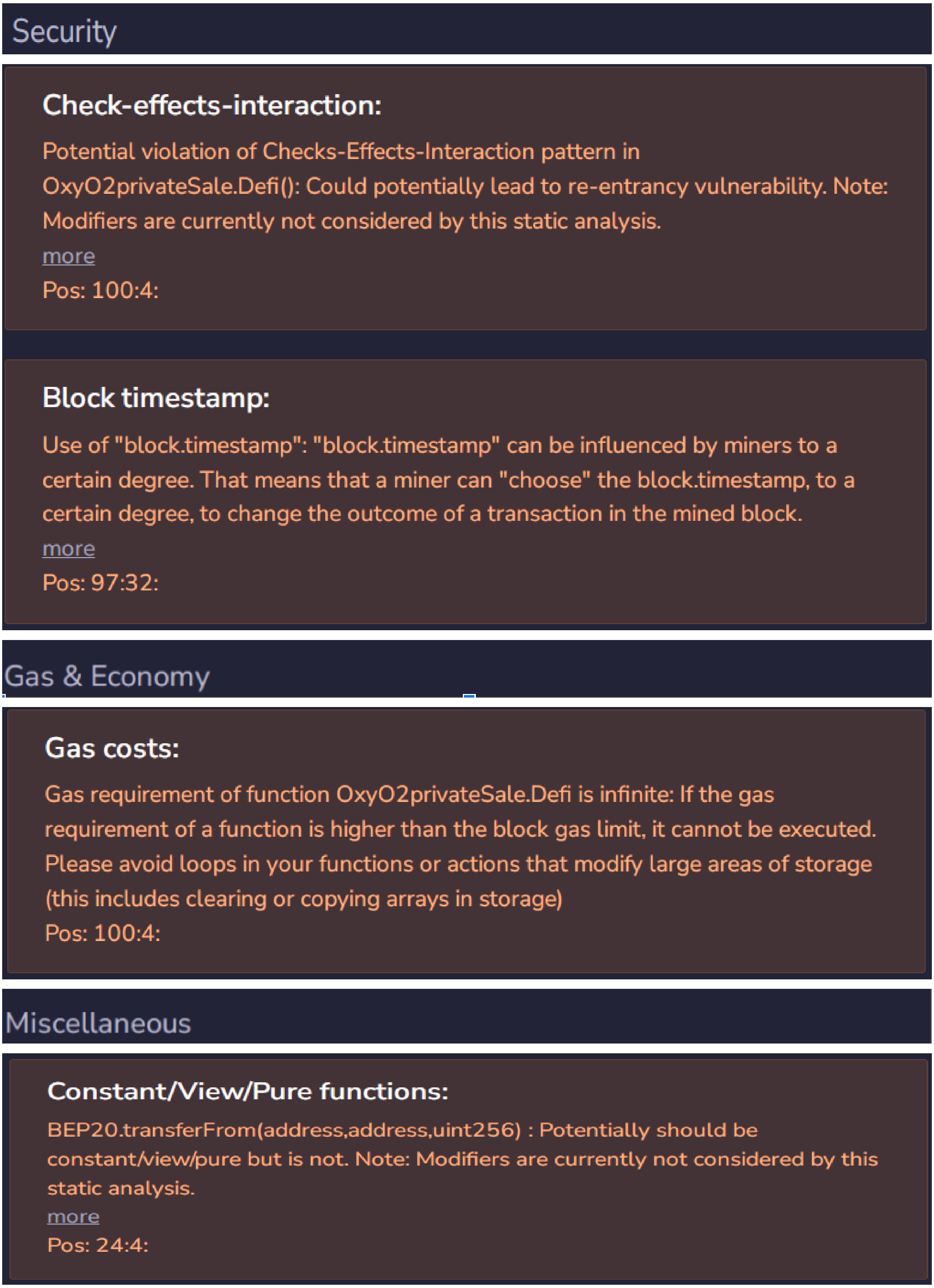
```
Oxy02airdrop.Airdrop() (Oxy02airdrop.sol#162-184) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02airdrop.sol#165)
        - month > 0 (Oxy02airdrop.sol#166)
        - ox2balance < amount * month (Oxy02airdrop.sol#169)
        - currentmonth == 1 (Oxy02airdrop.sol#176)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (Oxy02airdrop.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (Oxy02airdrop.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (Oxy02airdrop.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Oxy02airdrop.Airdrop() (Oxy02airdrop.sol#162-184) is not in mixedCase
Variable Oxy02airdrop.MAX (Oxy02airdrop.sol#138) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Oxy02airdrop.sol#95)" inContext (Oxy02airdrop.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Oxy02airdrop.slitherConstructorVariables() (Oxy02airdrop.sol#136-186) uses literals with too many digits:
        - MAX = 10000000e18 (Oxy02airdrop.sol#138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```



```
Oxy02airdrop.Airdrop() (Oxy02airdrop.sol#162-184) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(month > 0,timestamp exeeds) (Oxy02airdrop.sol#165)
        - month > 0 (Oxy02airdrop.sol#166)
        - ox2balance < amount * month (Oxy02airdrop.sol#169)
        - currentmonth == 1 (Oxy02airdrop.sol#176)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Context._msgData() (Oxy02airdrop.sol#94-97) is never used and should be removed
DateTime.diffMonths(uint256,uint256) (Oxy02airdrop.sol#82-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (Oxy02airdrop.sol#6) allows old versions
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Oxy02airdrop.Airdrop() (Oxy02airdrop.sol#162-184) is not in mixedCase
Variable Oxy02airdrop.MAX (Oxy02airdrop.sol#138) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Oxy02airdrop.sol#95)" inContext (Oxy02airdrop.sol#89-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Oxy02airdrop.slitherConstructorVariables() (Oxy02airdrop.sol#136-186) uses literals with too many digits:
        - MAX = 10000000e18 (Oxy02airdrop.sol#138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Solidity Static Analysis

Defi.sol



Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in OxyO2privateSale.Defi(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 100:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 97:32:

Gas & Economy

Gas costs:

Gas requirement of function OxyO2privateSale.Defi is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 100:4:

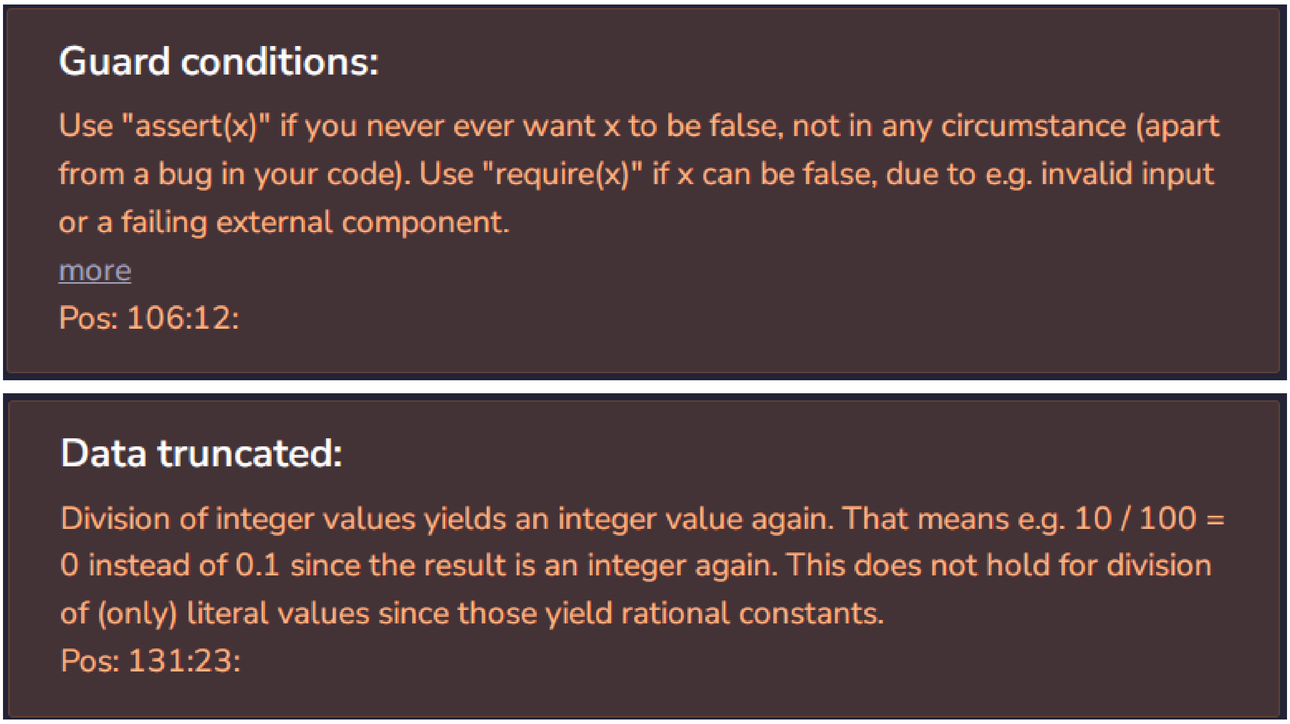
Miscellaneous

Constant/View/Pure functions:

BEP20.transferFrom(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 24:4:



Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

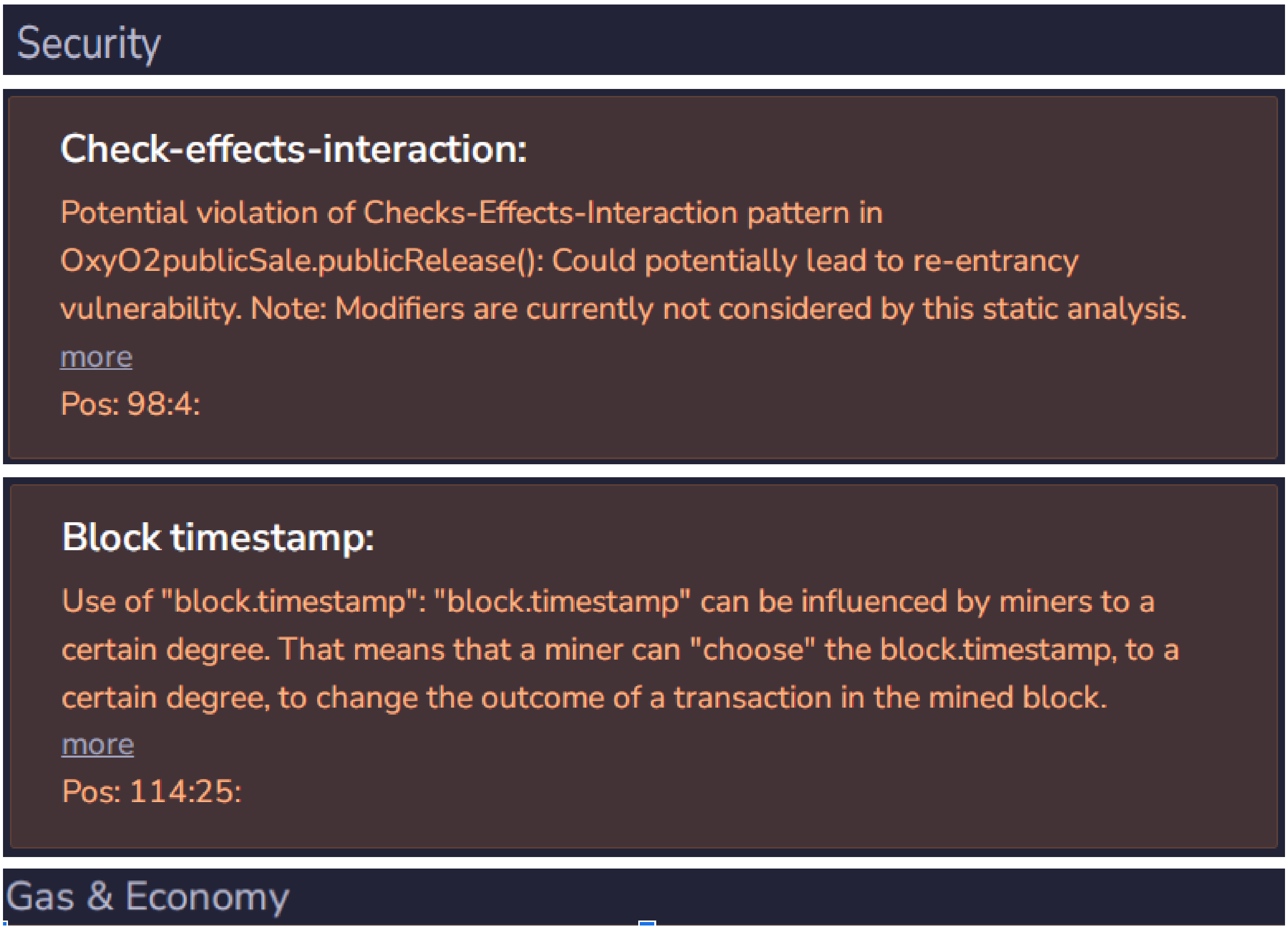
Pos: 106:12:

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 131:23:

OxyO2publicSale.sol



Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in OxyO2publicSale.publicRelease(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 98:4:

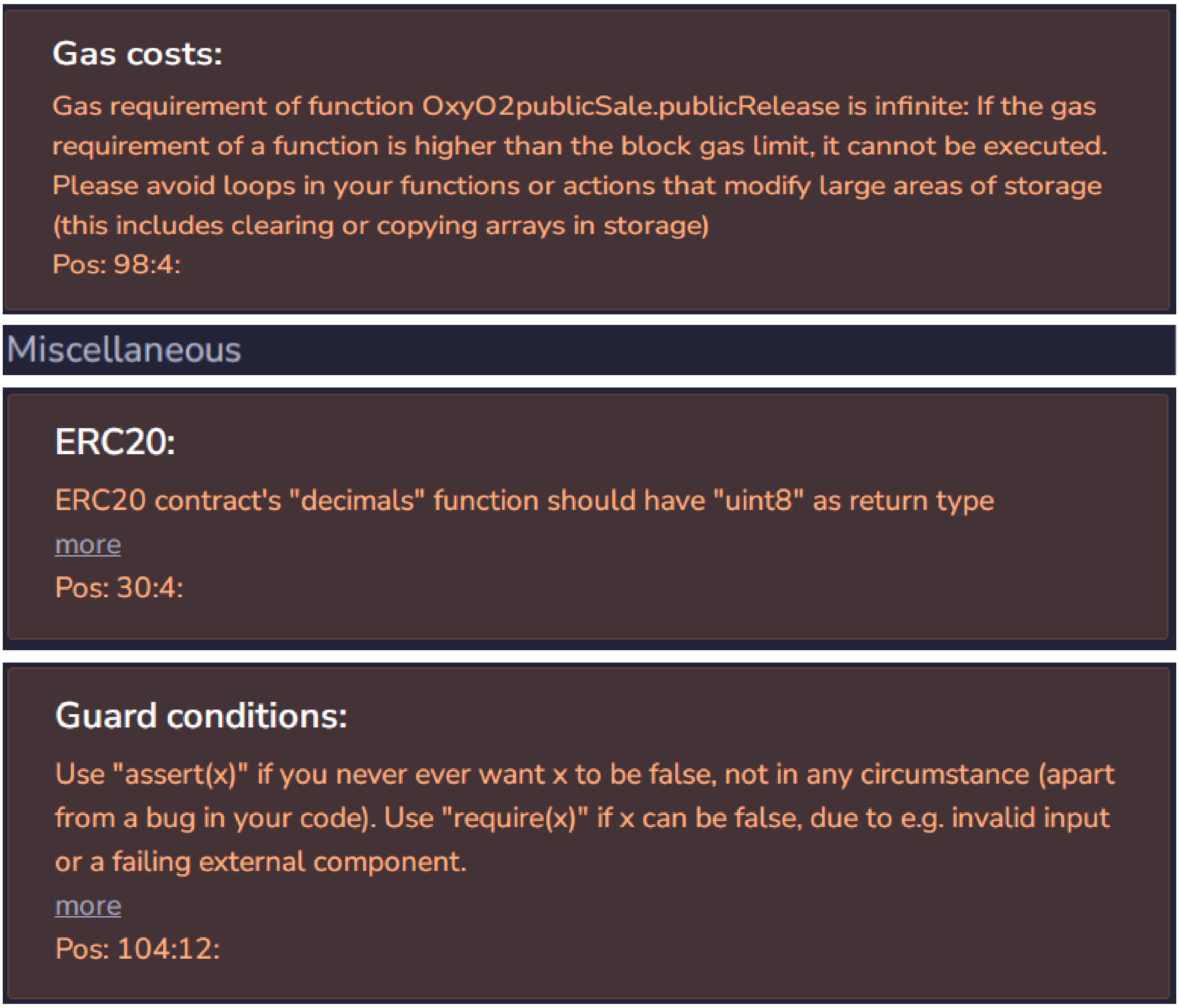
Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 114:25:

Gas & Economy



Gas costs:

Gas requirement of function OxyO2publicSale.publicRelease is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 98:4:

Miscellaneous

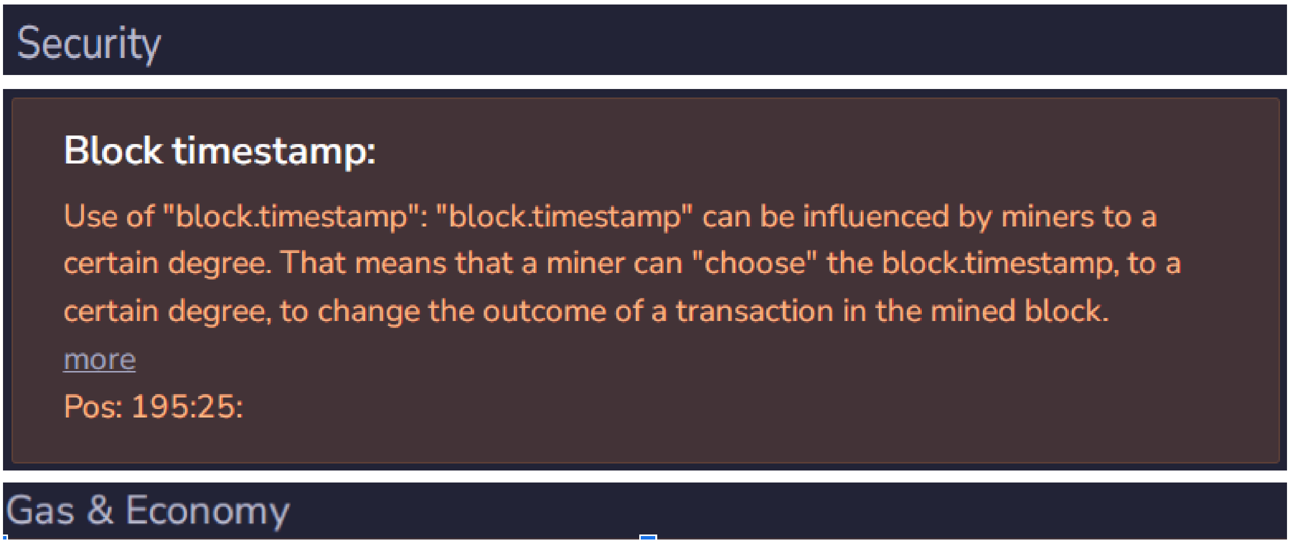
ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type
more
Pos: 30:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 104:12:

OxyO2Ecosystem.sol

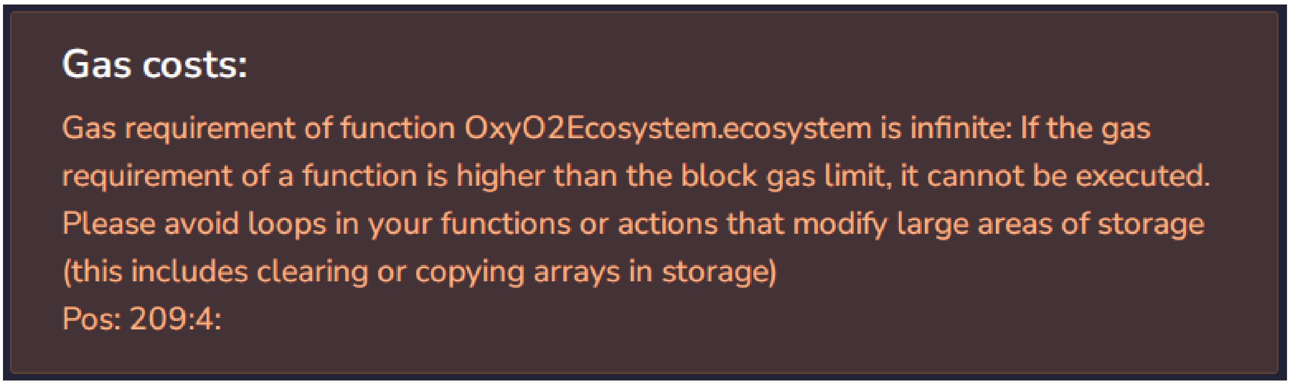


Security

Block timestamp:

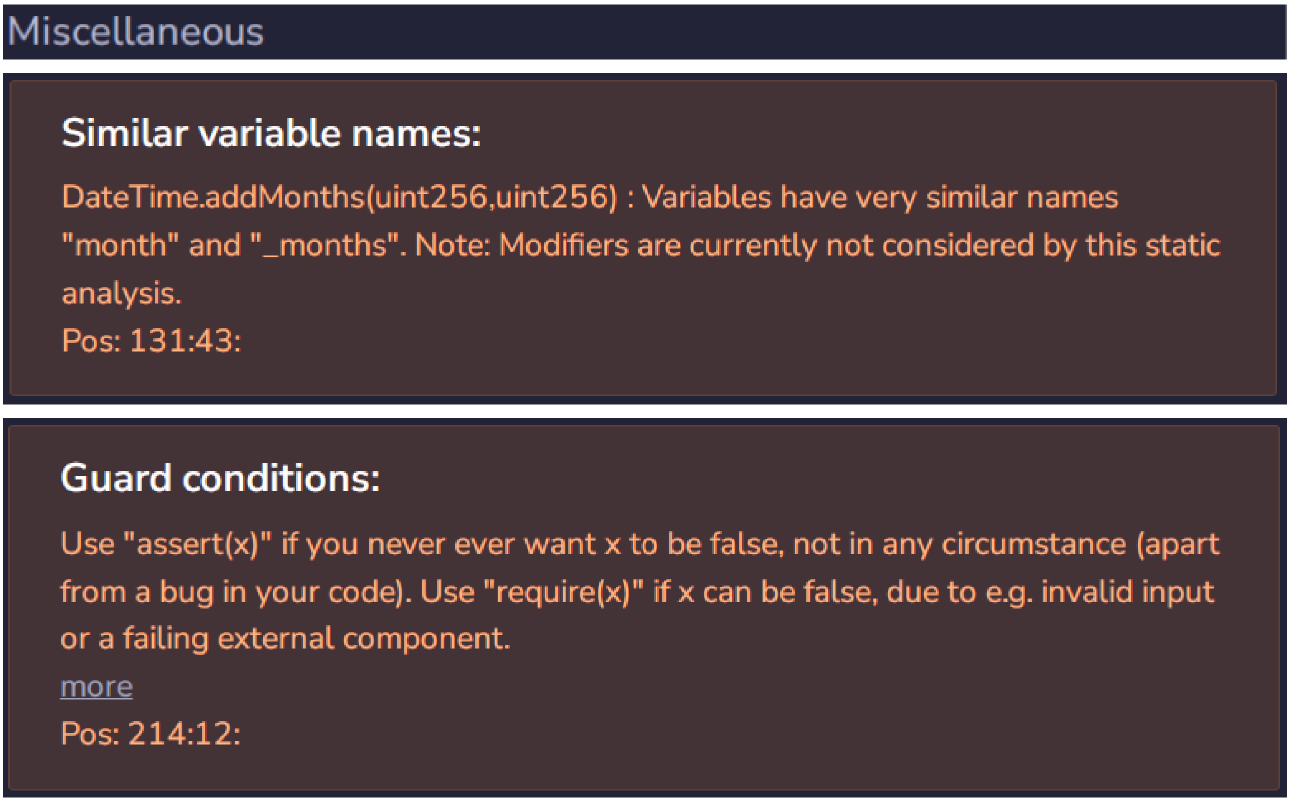
Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 195:25:

Gas & Economy



Gas costs:

Gas requirement of function OxyO2Ecosystem.ecosystem is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 209:4:



Miscellaneous

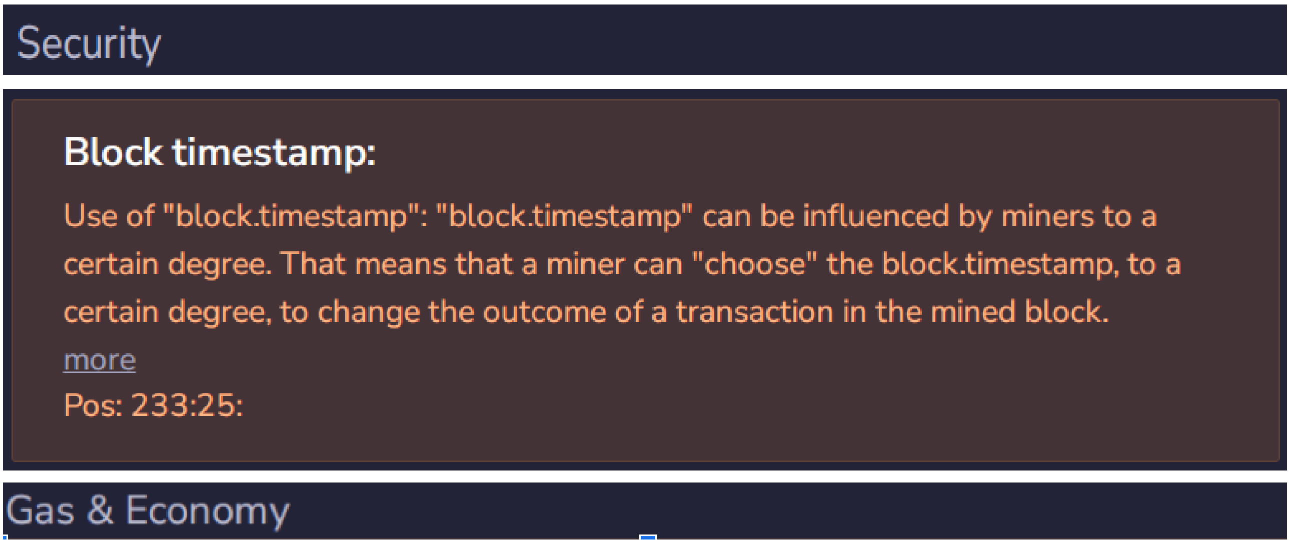
Similar variable names:

DateTime.addMonths(uint256,uint256) : Variables have very similar names "month" and "_months". Note: Modifiers are currently not considered by this static analysis.
Pos: 131:43:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 214:12:

OxyO2founders.sol

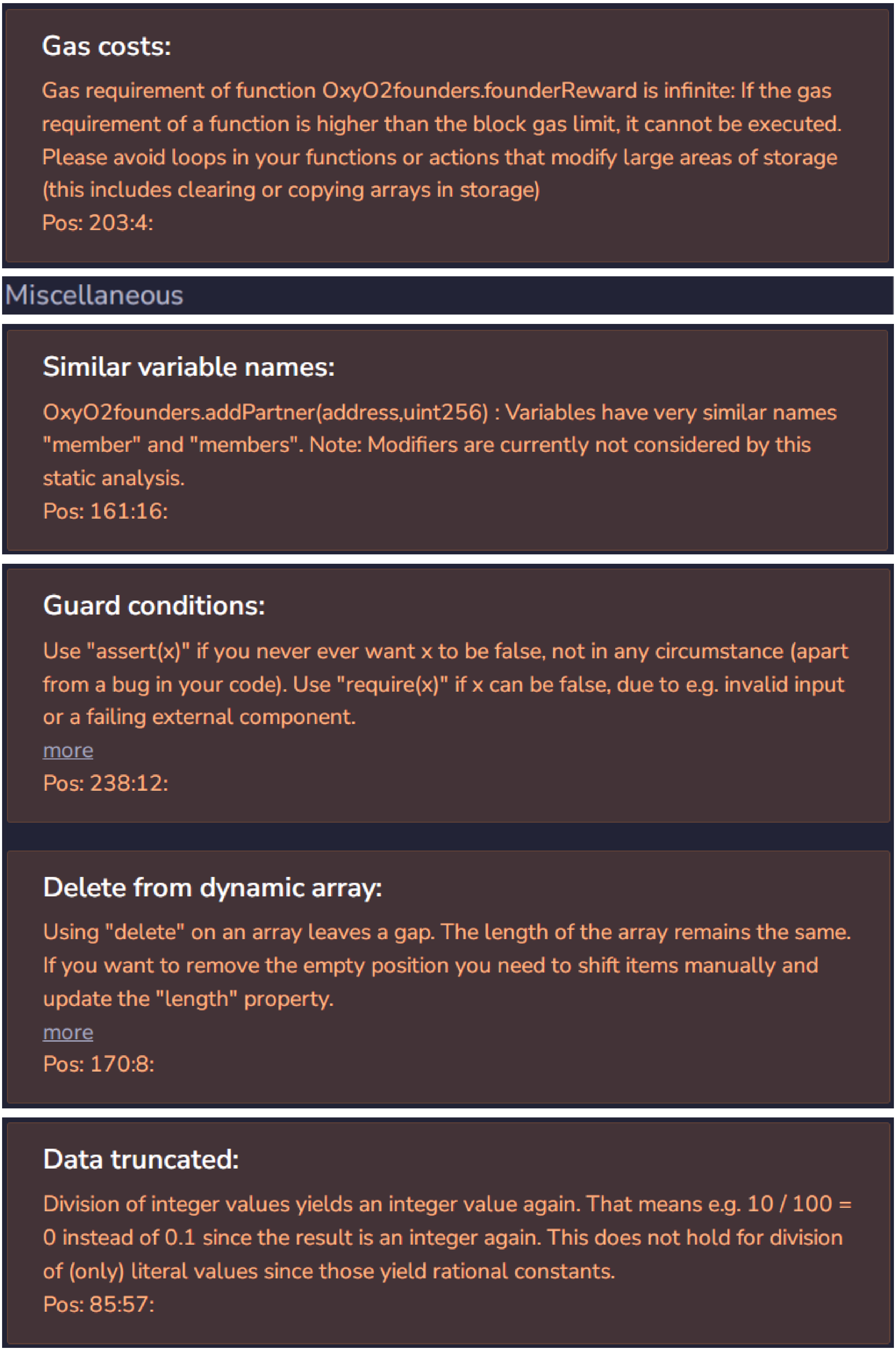


Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 233:25:

Gas & Economy



Gas costs:

Gas requirement of function OxyO2founders.founderReward is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 203:4:

Miscellaneous

Similar variable names:

OxyO2founders.addPartner(address,uint256) : Variables have very similar names "member" and "members". Note: Modifiers are currently not considered by this static analysis.

Pos: 161:16:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 238:12:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

more

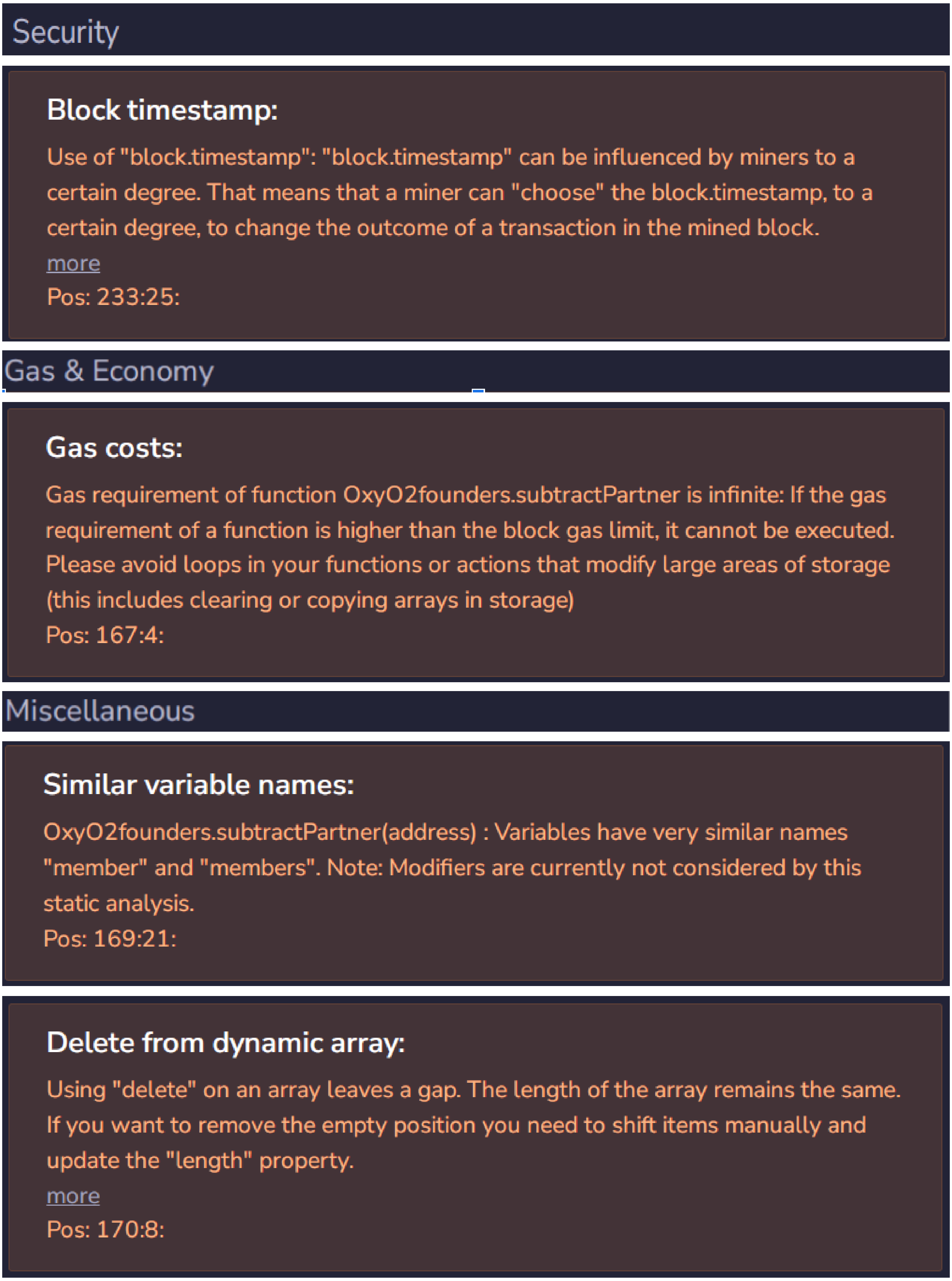
Pos: 170:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 85:57:

OxyO2founders1.sol



Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 233:25:

Gas & Economy

Gas costs:

Gas requirement of function OxyO2founders.subtractPartner is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 167:4:

Miscellaneous

Similar variable names:

OxyO2founders.subtractPartner(address) : Variables have very similar names "member" and "members". Note: Modifiers are currently not considered by this static analysis.

Pos: 169:21:

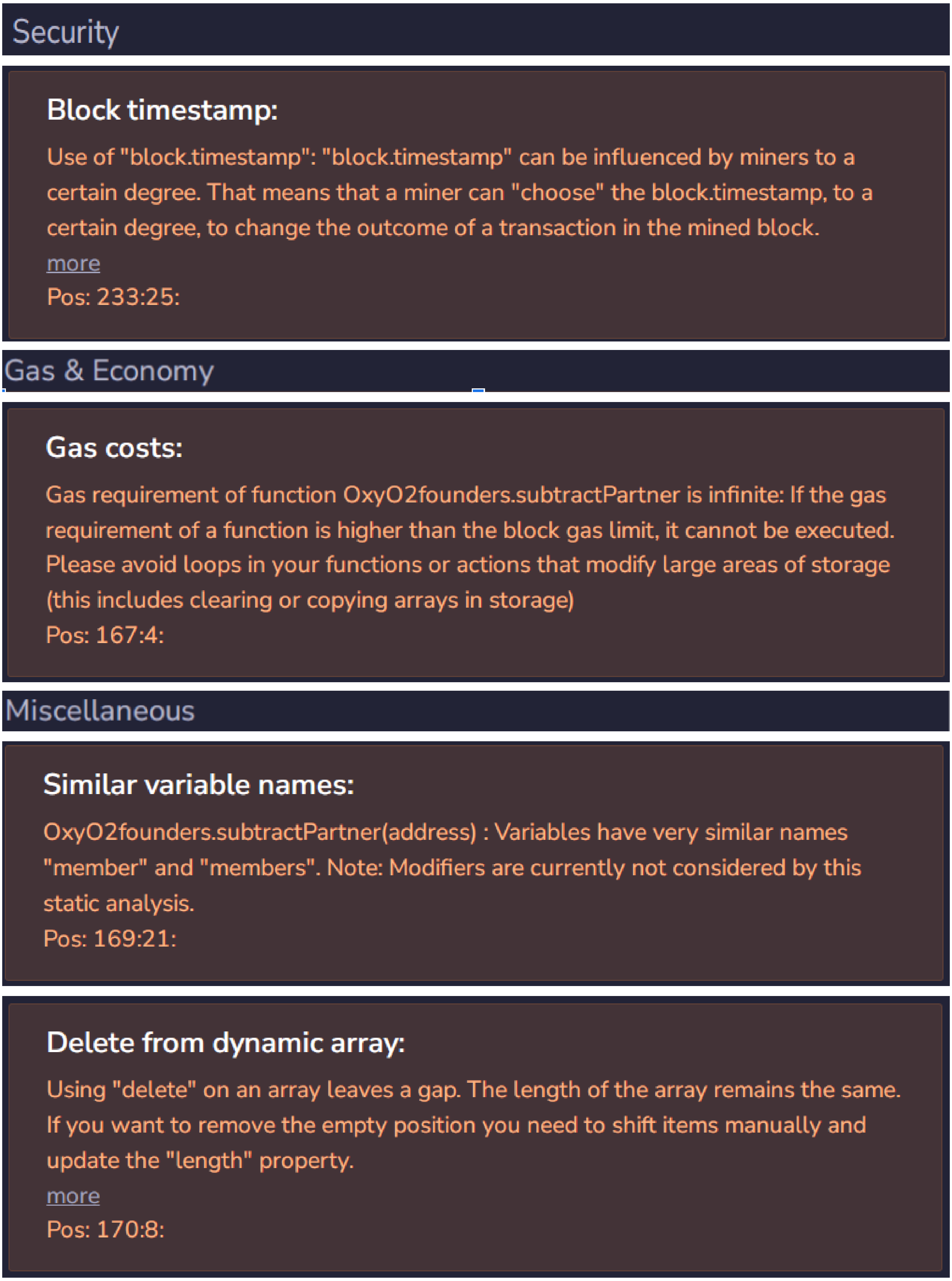
Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

more

Pos: 170:8:

OxyO2founders2.sol



Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 233:25:

Gas & Economy

Gas costs:

Gas requirement of function OxyO2founders.subtractPartner is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 167:4:

Miscellaneous

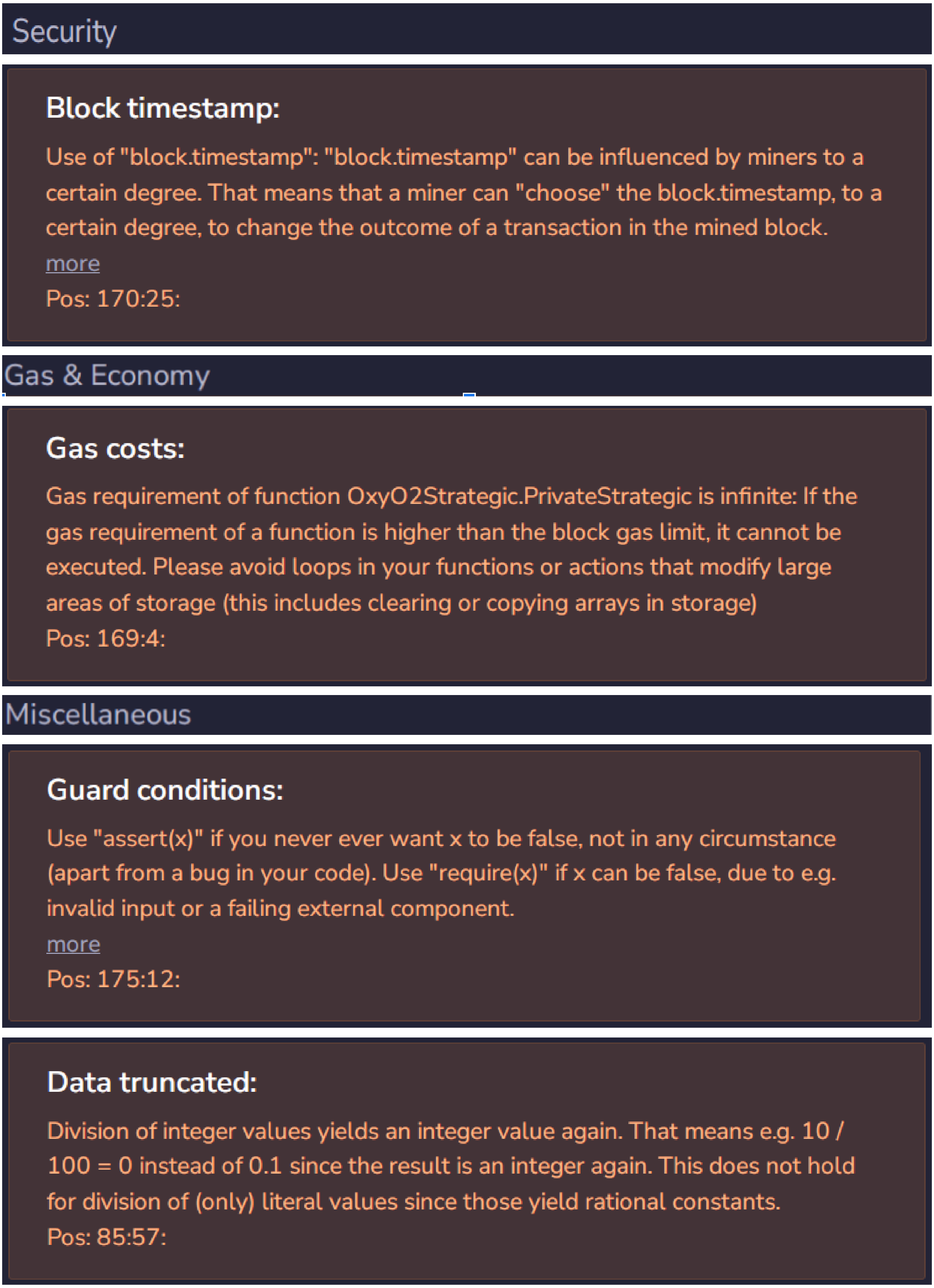
Similar variable names:

OxyO2founders.subtractPartner(address) : Variables have very similar names "member" and "members". Note: Modifiers are currently not considered by this static analysis.
Pos: 169:21:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 170:8:

OxyO2Strategic.sol



Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 170:25:

Gas & Economy

Gas costs:

Gas requirement of function OxyO2Strategic.PrivateStrategic is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 169:4:

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 175:12:

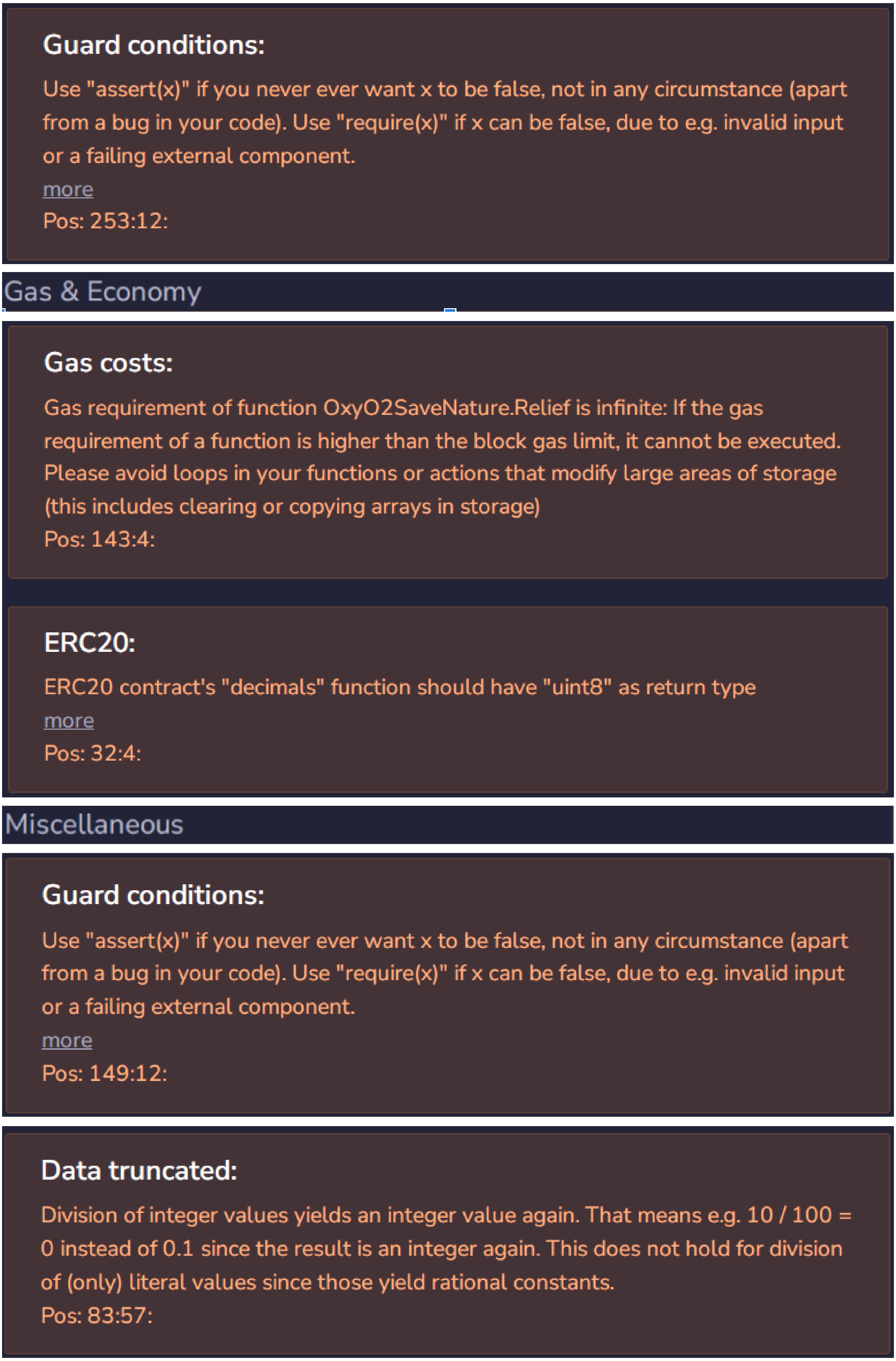
Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 85:57:

OxyO2SaveNature.sol

Security



Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 253:12:

Gas & Economy

Gas costs:

Gas requirement of function OxyO2SaveNature.Relief is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 143:4:

ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type

more

Pos: 32:4:

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

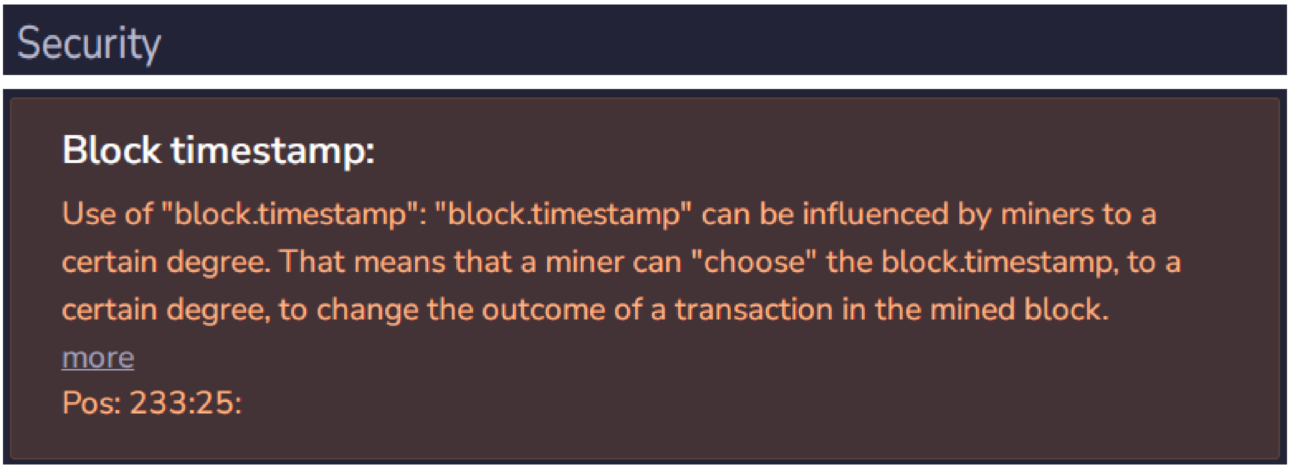
Pos: 149:12:

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 83:57:

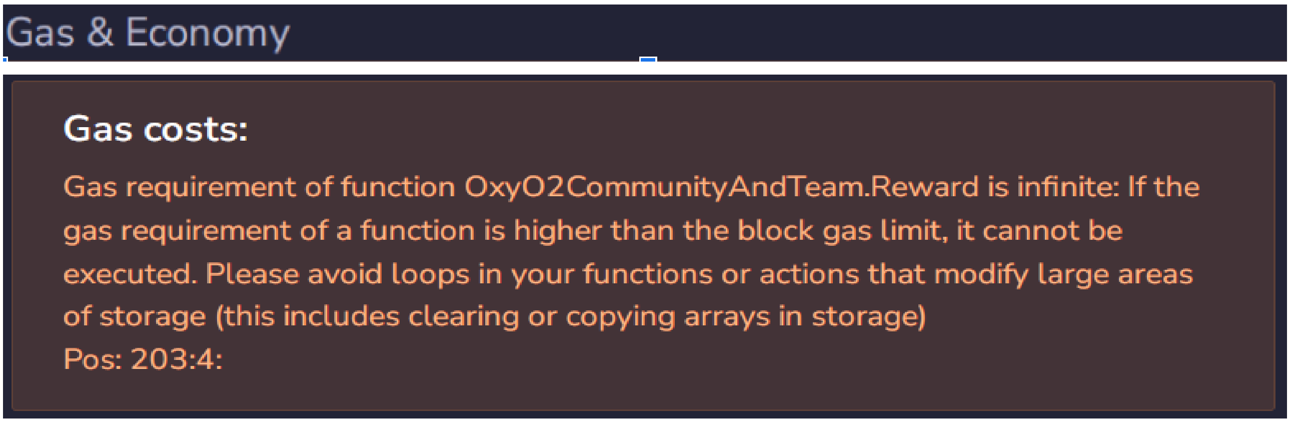
OxyO2CommunityAndTeam.sol



Security

Block timestamp:

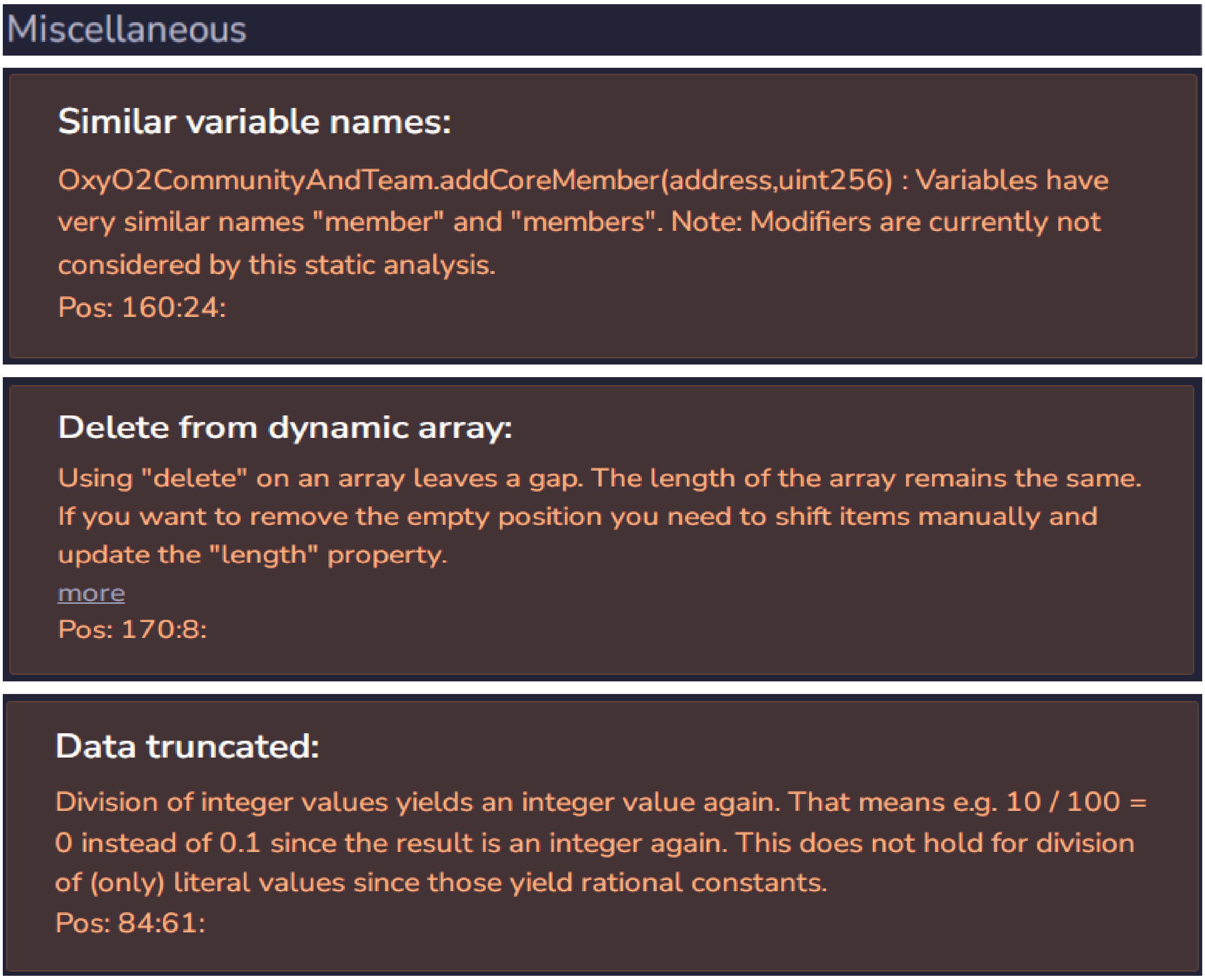
Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 233:25:



Gas & Economy

Gas costs:

Gas requirement of function OxyO2CommunityAndTeam.Reward is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 203:4:



Miscellaneous

Similar variable names:

OxyO2CommunityAndTeam.addCoreMember(address,uint256) : Variables have very similar names "member" and "members". Note: Modifiers are currently not considered by this static analysis.
Pos: 160:24:

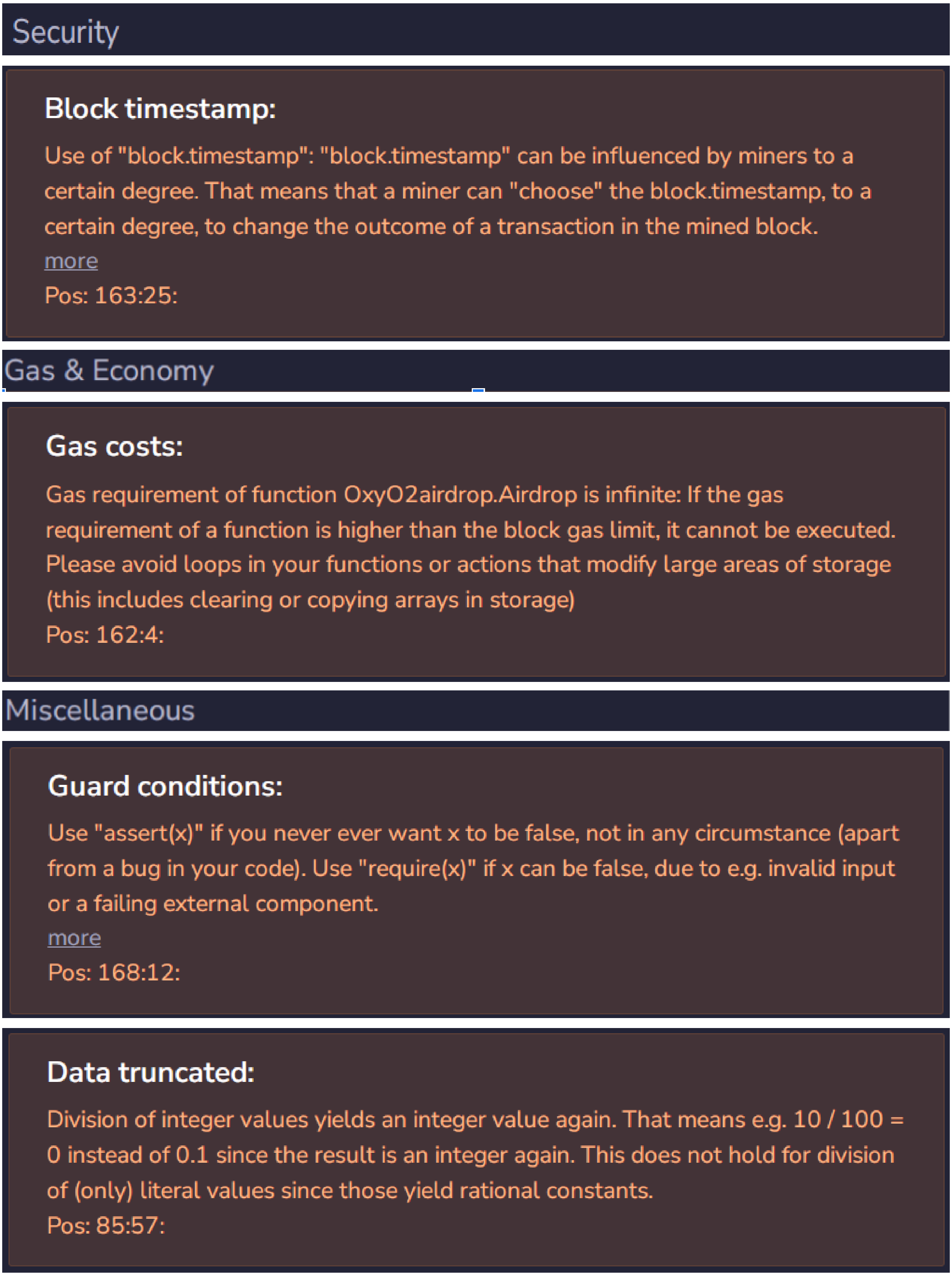
Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 170:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.
Pos: 84:61:

OxyO2airdrop.sol



Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 163:25:

Gas & Economy

Gas costs:

Gas requirement of function OxyO2airdrop.Airdrop is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 162:4:

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 168:12:

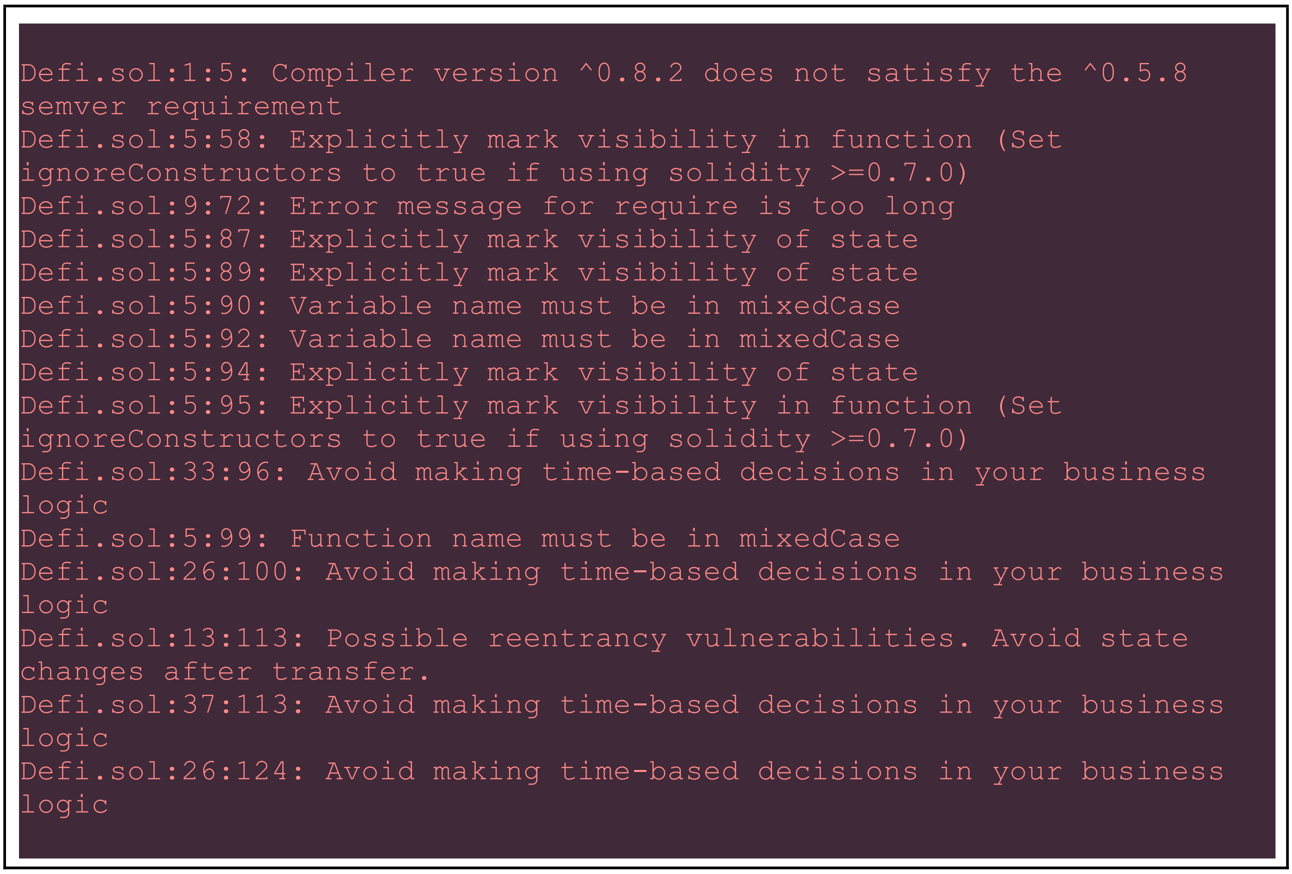
Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 85:57:

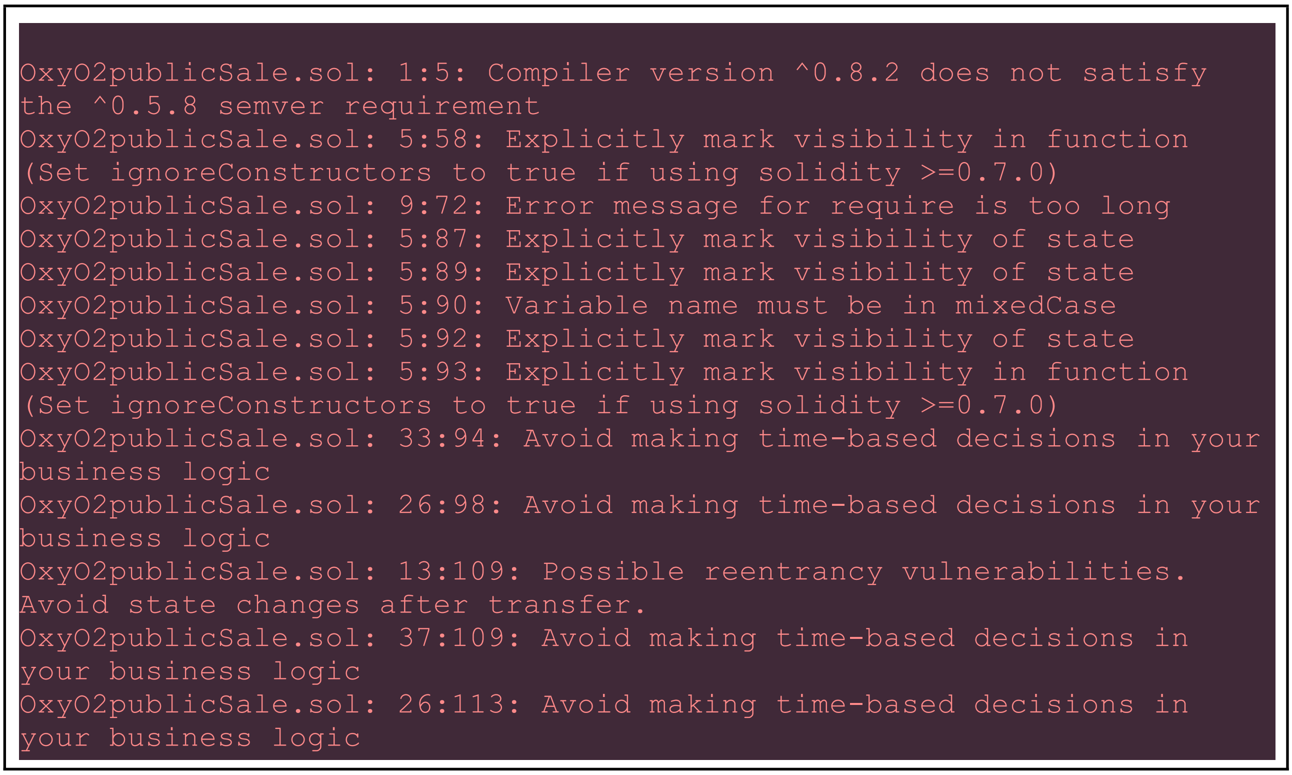
Solhint Linter

Defi.sol



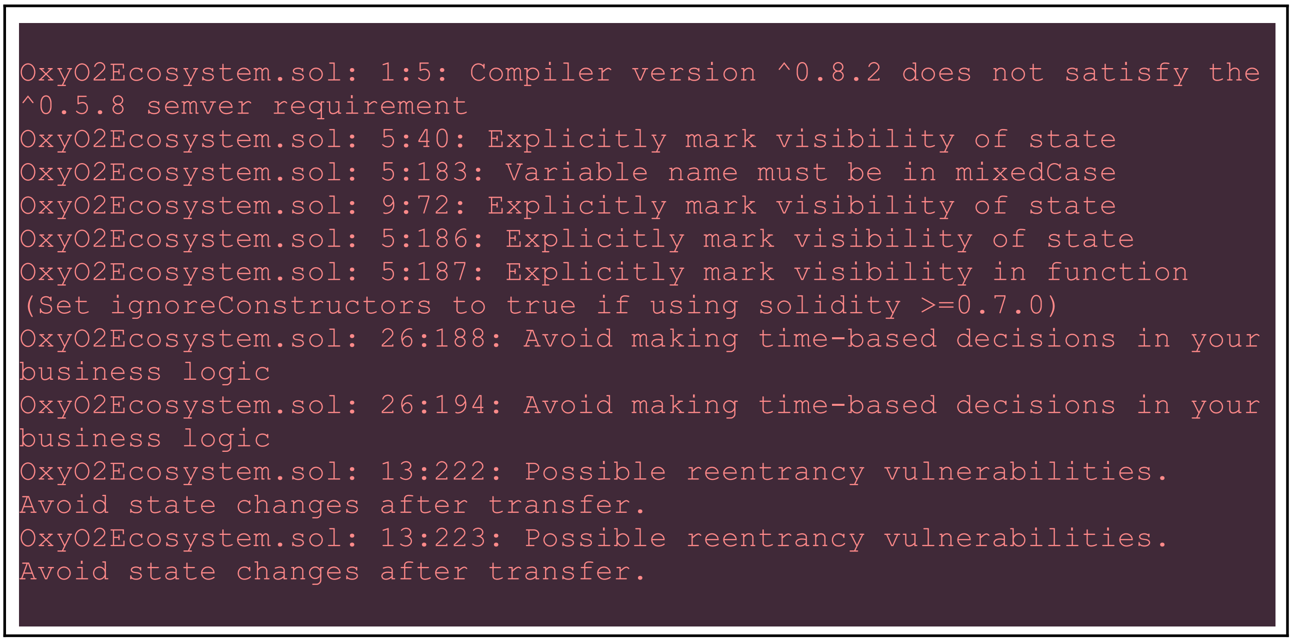
```
Defi.sol:1:5: Compiler version ^0.8.2 does not satisfy the ^0.5.8
semver requirement
Defi.sol:5:58: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
Defi.sol:9:72: Error message for require is too long
Defi.sol:5:87: Explicitly mark visibility of state
Defi.sol:5:89: Explicitly mark visibility of state
Defi.sol:5:90: Variable name must be in mixedCase
Defi.sol:5:92: Variable name must be in mixedCase
Defi.sol:5:94: Explicitly mark visibility of state
Defi.sol:5:95: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
Defi.sol:33:96: Avoid making time-based decisions in your business
logic
Defi.sol:5:99: Function name must be in mixedCase
Defi.sol:26:100: Avoid making time-based decisions in your business
logic
Defi.sol:13:113: Possible reentrancy vulnerabilities. Avoid state
changes after transfer.
Defi.sol:37:113: Avoid making time-based decisions in your business
logic
Defi.sol:26:124: Avoid making time-based decisions in your business
logic
```

OxyO2publicSale.sol



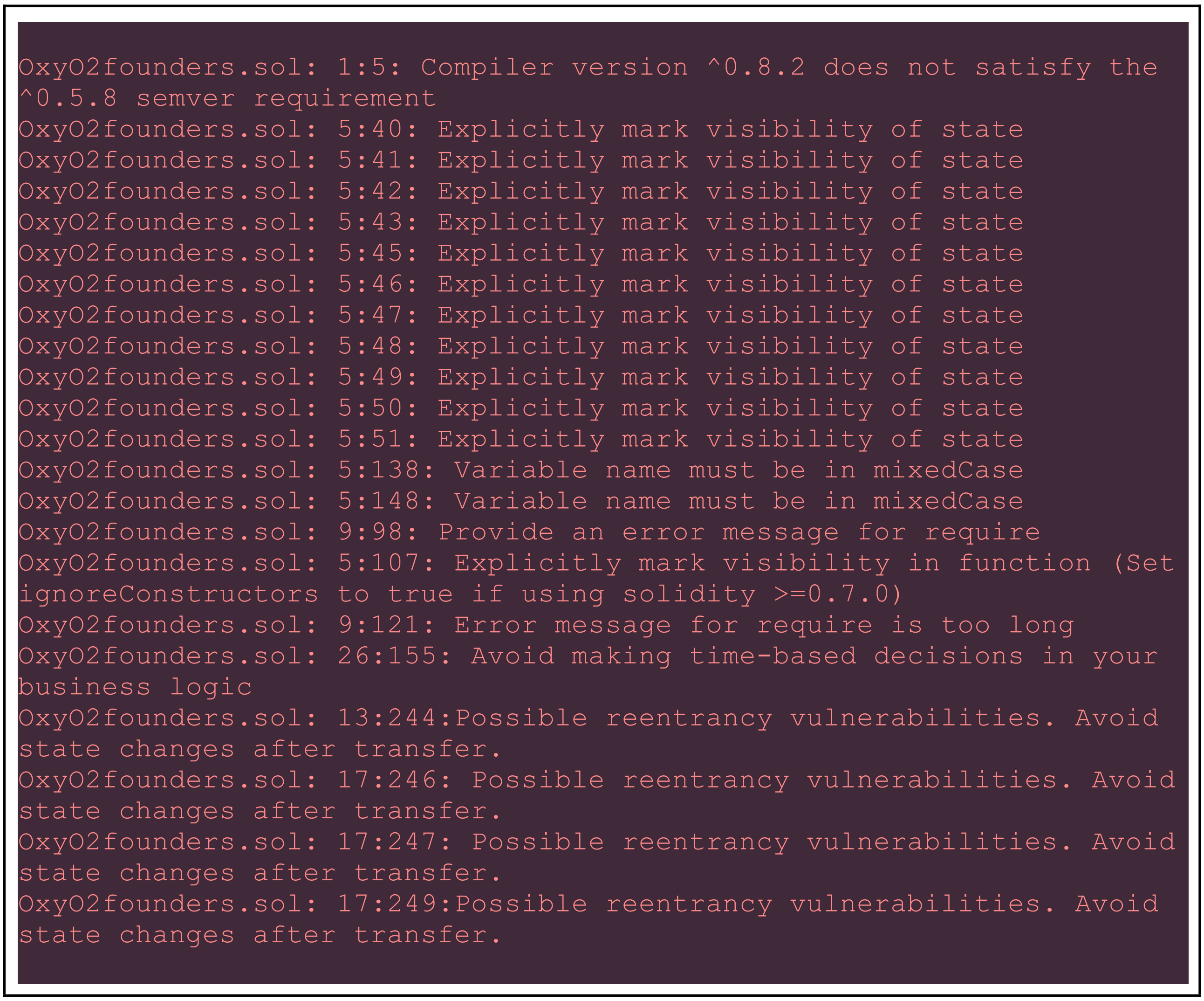
```
OxyO2publicSale.sol: 1:5: Compiler version ^0.8.2 does not satisfy
the ^0.5.8 semver requirement
OxyO2publicSale.sol: 5:58: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2publicSale.sol: 9:72: Error message for require is too long
OxyO2publicSale.sol: 5:87: Explicitly mark visibility of state
OxyO2publicSale.sol: 5:89: Explicitly mark visibility of state
OxyO2publicSale.sol: 5:90: Variable name must be in mixedCase
OxyO2publicSale.sol: 5:92: Explicitly mark visibility of state
OxyO2publicSale.sol: 5:93: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2publicSale.sol: 33:94: Avoid making time-based decisions in your
business logic
OxyO2publicSale.sol: 26:98: Avoid making time-based decisions in your
business logic
OxyO2publicSale.sol: 13:109: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2publicSale.sol: 37:109: Avoid making time-based decisions in
your business logic
OxyO2publicSale.sol: 26:113: Avoid making time-based decisions in
your business logic
```

OxyO2Ecosystem.sol



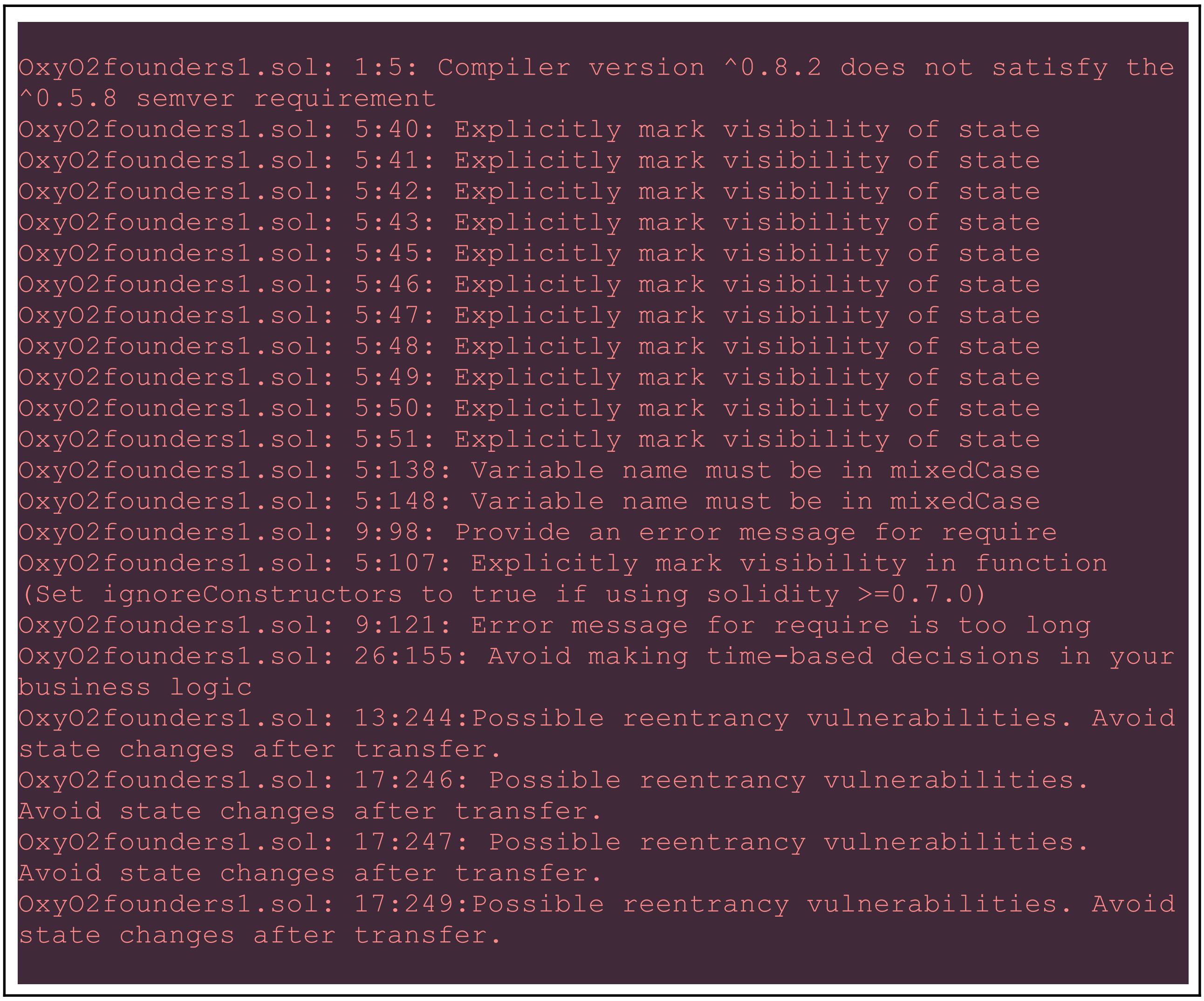
```
OxyO2Ecosystem.sol: 1:5: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2Ecosystem.sol: 5:40: Explicitly mark visibility of state
OxyO2Ecosystem.sol: 5:183: Variable name must be in mixedCase
OxyO2Ecosystem.sol: 9:72: Explicitly mark visibility of state
OxyO2Ecosystem.sol: 5:186: Explicitly mark visibility of state
OxyO2Ecosystem.sol: 5:187: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2Ecosystem.sol: 26:188: Avoid making time-based decisions in your
business logic
OxyO2Ecosystem.sol: 26:194: Avoid making time-based decisions in your
business logic
OxyO2Ecosystem.sol: 13:222: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2Ecosystem.sol: 13:223: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
```

OxyO2founders.sol



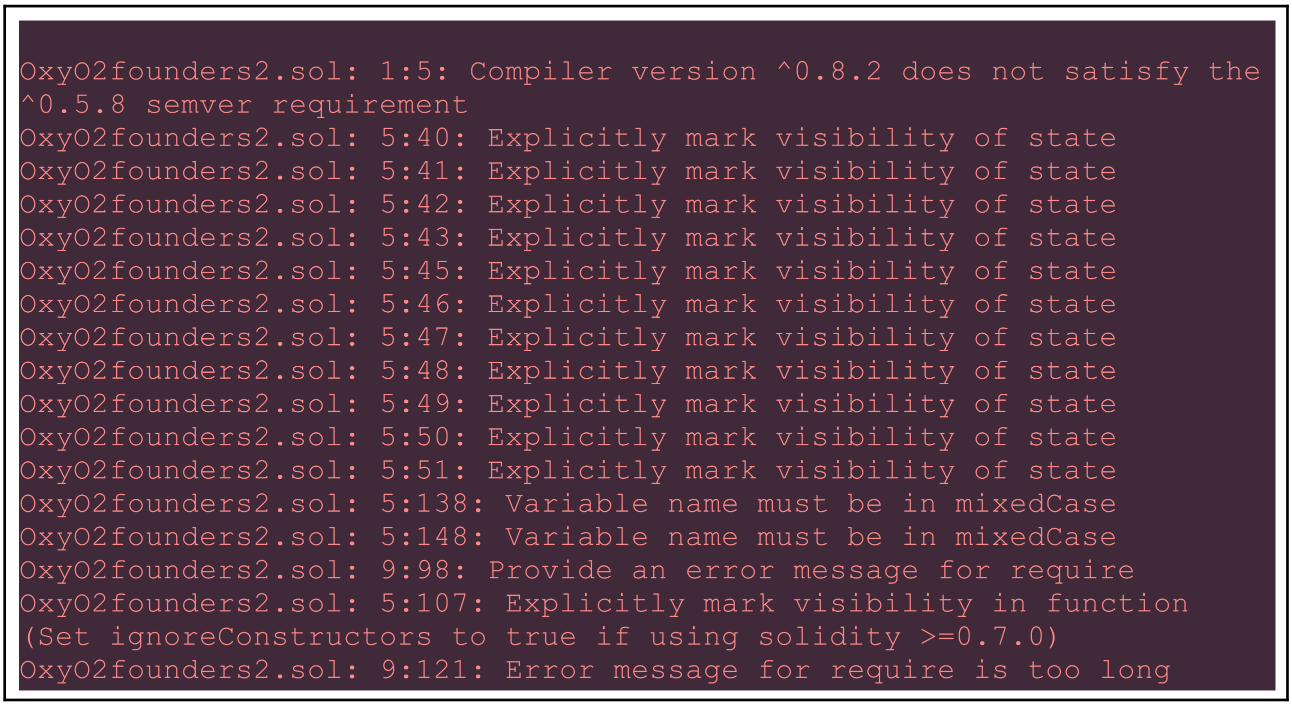
```
OxyO2founders.sol: 1:5: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2founders.sol: 5:40: Explicitly mark visibility of state
OxyO2founders.sol: 5:41: Explicitly mark visibility of state
OxyO2founders.sol: 5:42: Explicitly mark visibility of state
OxyO2founders.sol: 5:43: Explicitly mark visibility of state
OxyO2founders.sol: 5:45: Explicitly mark visibility of state
OxyO2founders.sol: 5:46: Explicitly mark visibility of state
OxyO2founders.sol: 5:47: Explicitly mark visibility of state
OxyO2founders.sol: 5:48: Explicitly mark visibility of state
OxyO2founders.sol: 5:49: Explicitly mark visibility of state
OxyO2founders.sol: 5:50: Explicitly mark visibility of state
OxyO2founders.sol: 5:51: Explicitly mark visibility of state
OxyO2founders.sol: 5:138: Variable name must be in mixedCase
OxyO2founders.sol: 5:148: Variable name must be in mixedCase
OxyO2founders.sol: 9:98: Provide an error message for require
OxyO2founders.sol: 5:107: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
OxyO2founders.sol: 9:121: Error message for require is too long
OxyO2founders.sol: 26:155: Avoid making time-based decisions in your
business logic
OxyO2founders.sol: 13:244:Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
OxyO2founders.sol: 17:246: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
OxyO2founders.sol: 17:247: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
OxyO2founders.sol: 17:249:Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
```

OxyO2founders1.sol

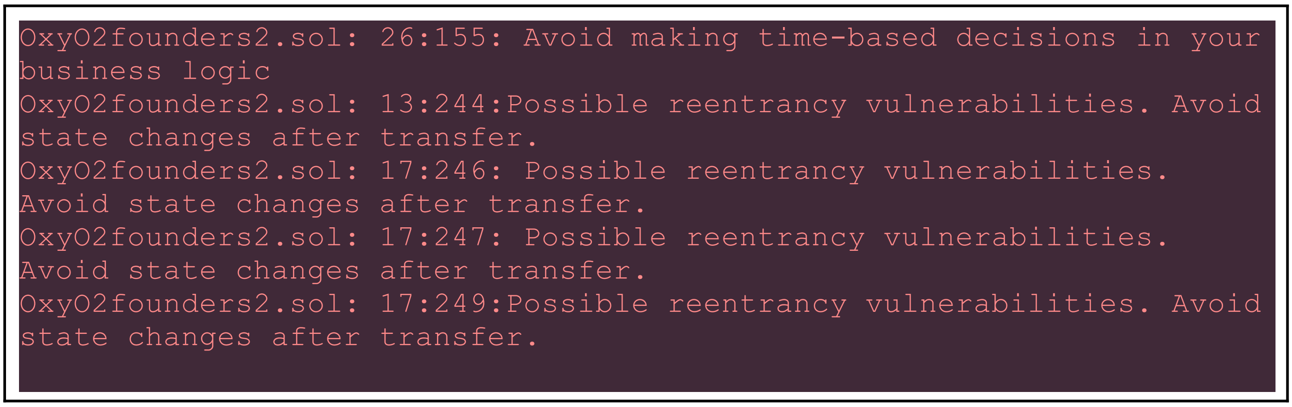


```
OxyO2founders1.sol: 1:5: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2founders1.sol: 5:40: Explicitly mark visibility of state
OxyO2founders1.sol: 5:41: Explicitly mark visibility of state
OxyO2founders1.sol: 5:42: Explicitly mark visibility of state
OxyO2founders1.sol: 5:43: Explicitly mark visibility of state
OxyO2founders1.sol: 5:45: Explicitly mark visibility of state
OxyO2founders1.sol: 5:46: Explicitly mark visibility of state
OxyO2founders1.sol: 5:47: Explicitly mark visibility of state
OxyO2founders1.sol: 5:48: Explicitly mark visibility of state
OxyO2founders1.sol: 5:49: Explicitly mark visibility of state
OxyO2founders1.sol: 5:50: Explicitly mark visibility of state
OxyO2founders1.sol: 5:51: Explicitly mark visibility of state
OxyO2founders1.sol: 5:138: Variable name must be in mixedCase
OxyO2founders1.sol: 5:148: Variable name must be in mixedCase
OxyO2founders1.sol: 9:98: Provide an error message for require
OxyO2founders1.sol: 5:107: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2founders1.sol: 9:121: Error message for require is too long
OxyO2founders1.sol: 26:155: Avoid making time-based decisions in your
business logic
OxyO2founders1.sol: 13:244:Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
OxyO2founders1.sol: 17:246: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2founders1.sol: 17:247: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2founders1.sol: 17:249:Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
```

OxyO2founders2.sol

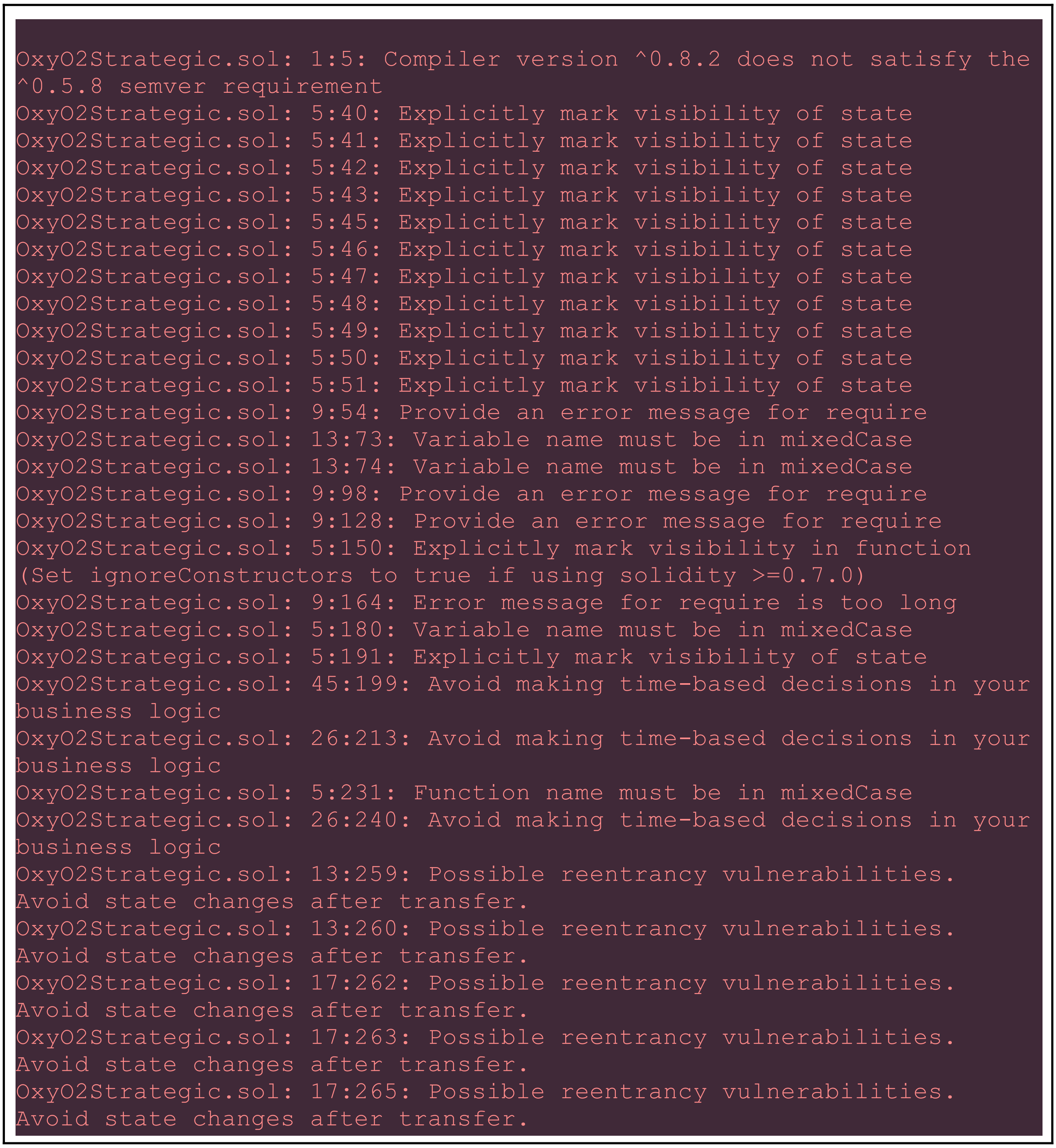


```
OxyO2founders2.sol: 1:5: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2founders2.sol: 5:40: Explicitly mark visibility of state
OxyO2founders2.sol: 5:41: Explicitly mark visibility of state
OxyO2founders2.sol: 5:42: Explicitly mark visibility of state
OxyO2founders2.sol: 5:43: Explicitly mark visibility of state
OxyO2founders2.sol: 5:45: Explicitly mark visibility of state
OxyO2founders2.sol: 5:46: Explicitly mark visibility of state
OxyO2founders2.sol: 5:47: Explicitly mark visibility of state
OxyO2founders2.sol: 5:48: Explicitly mark visibility of state
OxyO2founders2.sol: 5:49: Explicitly mark visibility of state
OxyO2founders2.sol: 5:50: Explicitly mark visibility of state
OxyO2founders2.sol: 5:51: Explicitly mark visibility of state
OxyO2founders2.sol: 5:138: Variable name must be in mixedCase
OxyO2founders2.sol: 5:148: Variable name must be in mixedCase
OxyO2founders2.sol: 9:98: Provide an error message for require
OxyO2founders2.sol: 5:107: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2founders2.sol: 9:121: Error message for require is too long
```



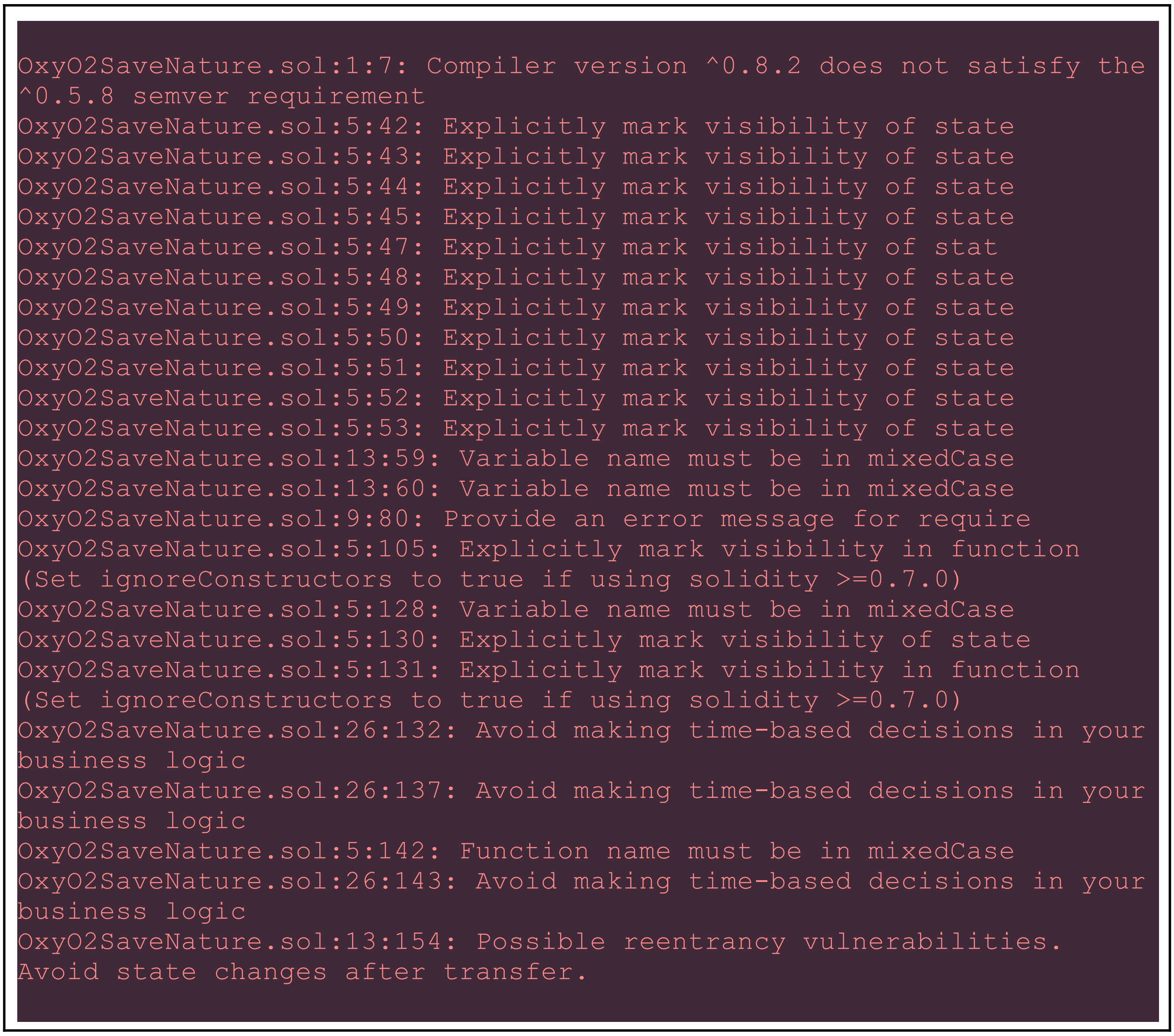
```
OxyO2founders2.sol: 26:155: Avoid making time-based decisions in your
business logic
OxyO2founders2.sol: 13:244:Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
OxyO2founders2.sol: 17:246: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2founders2.sol: 17:247: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2founders2.sol: 17:249:Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
```

OxyO2Strategic.sol



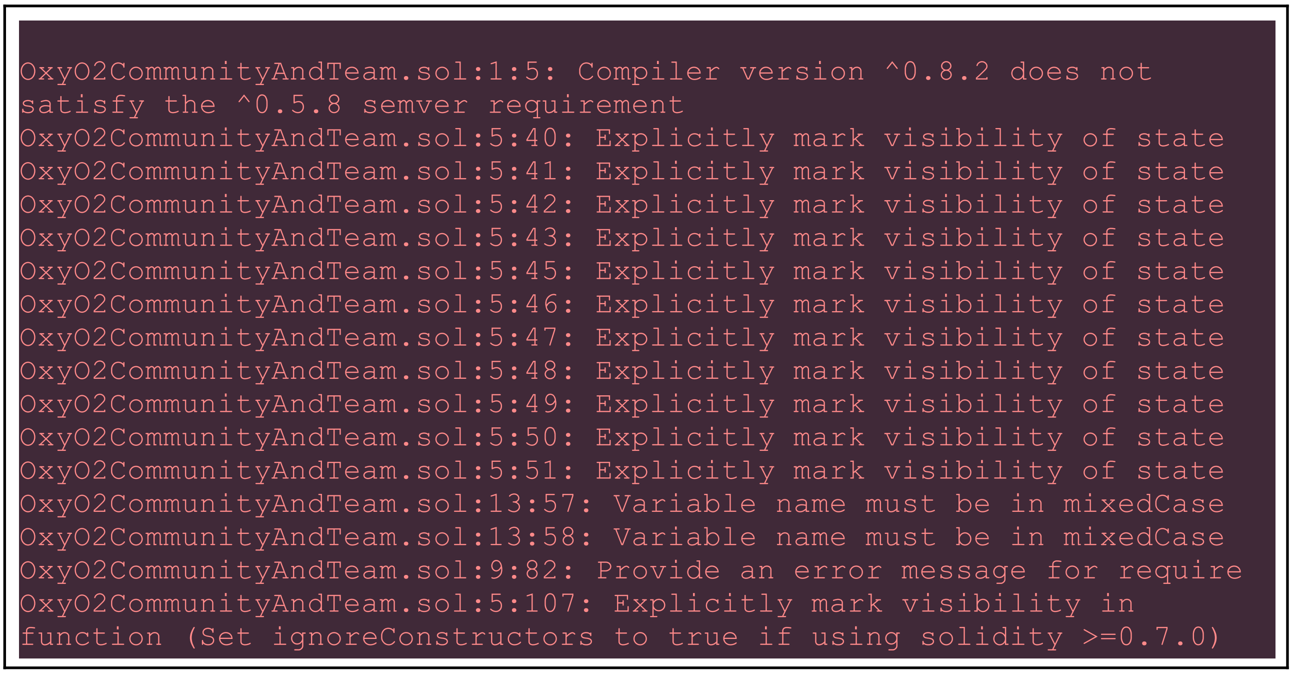
```
OxyO2Strategic.sol: 1:5: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2Strategic.sol: 5:40: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:41: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:42: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:43: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:45: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:46: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:47: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:48: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:49: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:50: Explicitly mark visibility of state
OxyO2Strategic.sol: 5:51: Explicitly mark visibility of state
OxyO2Strategic.sol: 9:54: Provide an error message for require
OxyO2Strategic.sol: 13:73: Variable name must be in mixedCase
OxyO2Strategic.sol: 13:74: Variable name must be in mixedCase
OxyO2Strategic.sol: 9:98: Provide an error message for require
OxyO2Strategic.sol: 9:128: Provide an error message for require
OxyO2Strategic.sol: 5:150: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2Strategic.sol: 9:164: Error message for require is too long
OxyO2Strategic.sol: 5:180: Variable name must be in mixedCase
OxyO2Strategic.sol: 5:191: Explicitly mark visibility of state
OxyO2Strategic.sol: 45:199: Avoid making time-based decisions in your
business logic
OxyO2Strategic.sol: 26:213: Avoid making time-based decisions in your
business logic
OxyO2Strategic.sol: 5:231: Function name must be in mixedCase
OxyO2Strategic.sol: 26:240: Avoid making time-based decisions in your
business logic
OxyO2Strategic.sol: 13:259: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2Strategic.sol: 13:260: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2Strategic.sol: 17:262: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2Strategic.sol: 17:263: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
OxyO2Strategic.sol: 17:265: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
```

OxyO2SaveNature.sol

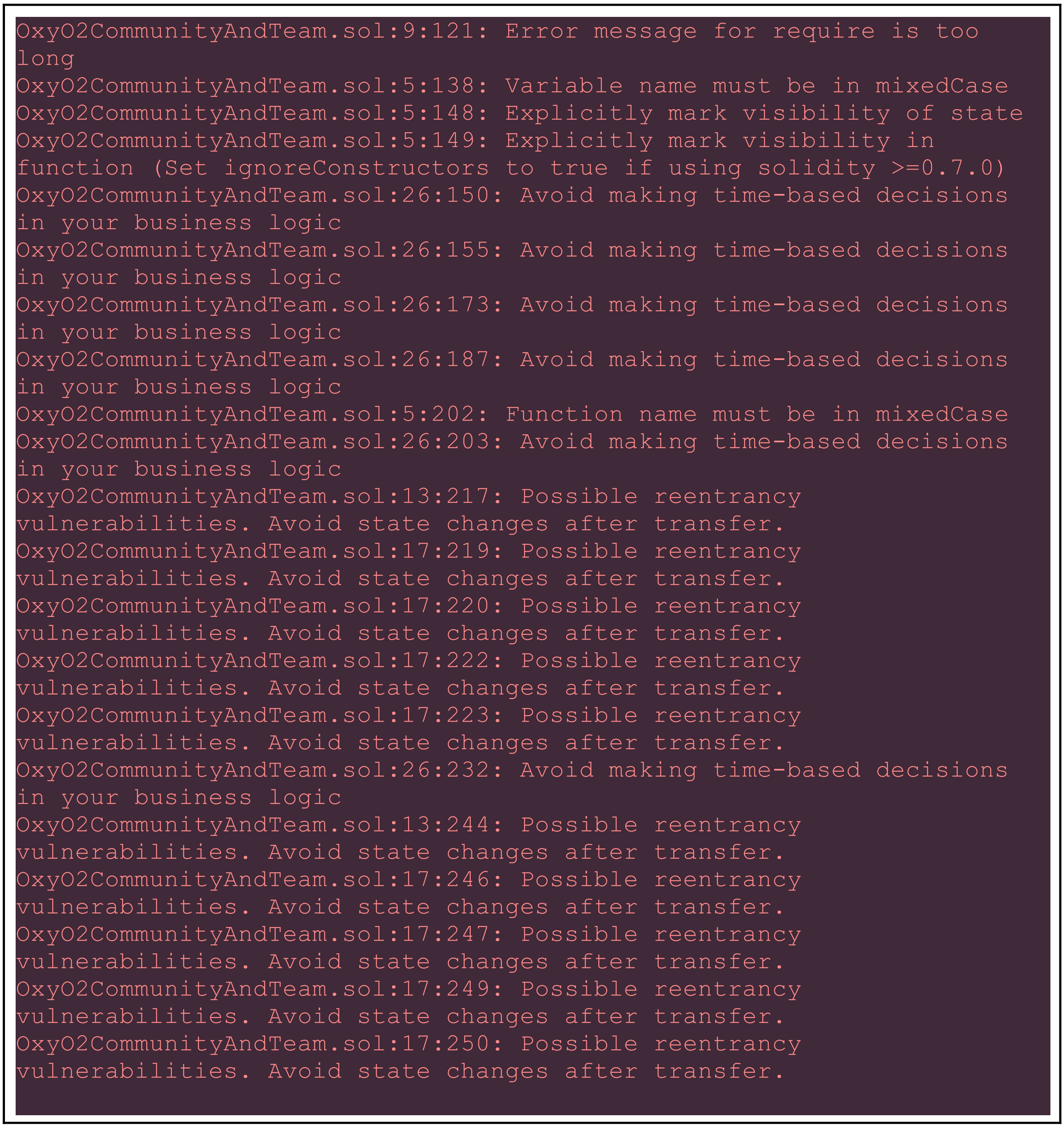


```
OxyO2SaveNature.sol:1:7: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2SaveNature.sol:5:42: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:43: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:44: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:45: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:47: Explicitly mark visibility of stat
OxyO2SaveNature.sol:5:48: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:49: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:50: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:51: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:52: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:53: Explicitly mark visibility of state
OxyO2SaveNature.sol:13:59: Variable name must be in mixedCase
OxyO2SaveNature.sol:13:60: Variable name must be in mixedCase
OxyO2SaveNature.sol:9:80: Provide an error message for require
OxyO2SaveNature.sol:5:105: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2SaveNature.sol:5:128: Variable name must be in mixedCase
OxyO2SaveNature.sol:5:130: Explicitly mark visibility of state
OxyO2SaveNature.sol:5:131: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2SaveNature.sol:26:132: Avoid making time-based decisions in your
business logic
OxyO2SaveNature.sol:26:137: Avoid making time-based decisions in your
business logic
OxyO2SaveNature.sol:5:142: Function name must be in mixedCase
OxyO2SaveNature.sol:26:143: Avoid making time-based decisions in your
business logic
OxyO2SaveNature.sol:13:154: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
```

OxyO2CommunityAndTeam.sol

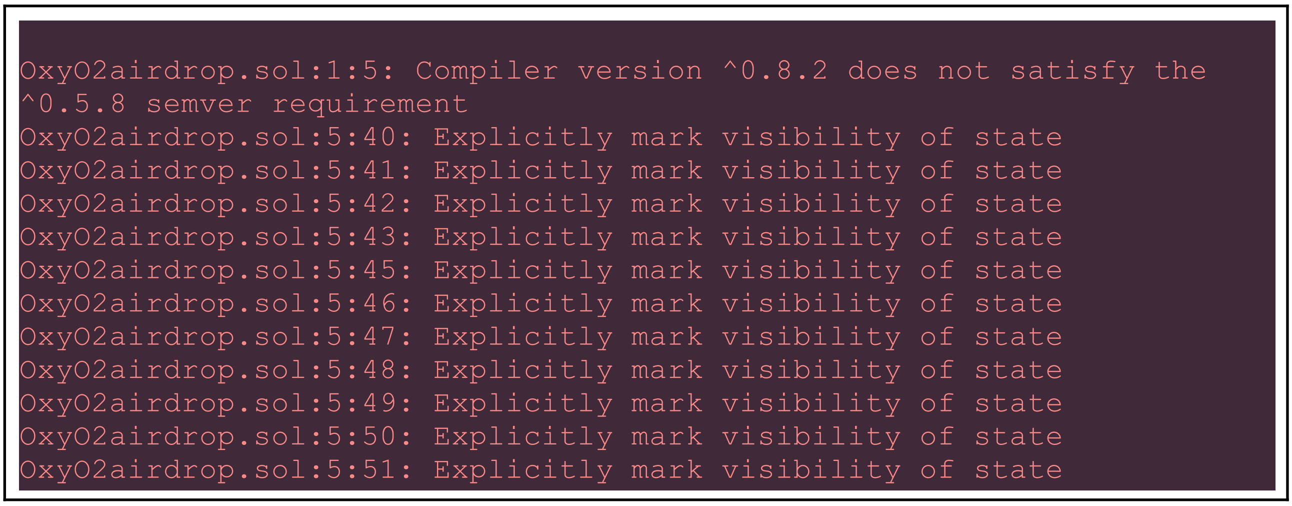


```
OxyO2CommunityAndTeam.sol:1:5: Compiler version ^0.8.2 does not
satisfy the ^0.5.8 semver requirement
OxyO2CommunityAndTeam.sol:5:40: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:41: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:42: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:43: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:45: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:46: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:47: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:48: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:49: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:50: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:51: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:13:57: Variable name must be in mixedCase
OxyO2CommunityAndTeam.sol:13:58: Variable name must be in mixedCase
OxyO2CommunityAndTeam.sol:9:82: Provide an error message for require
OxyO2CommunityAndTeam.sol:5:107: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
```

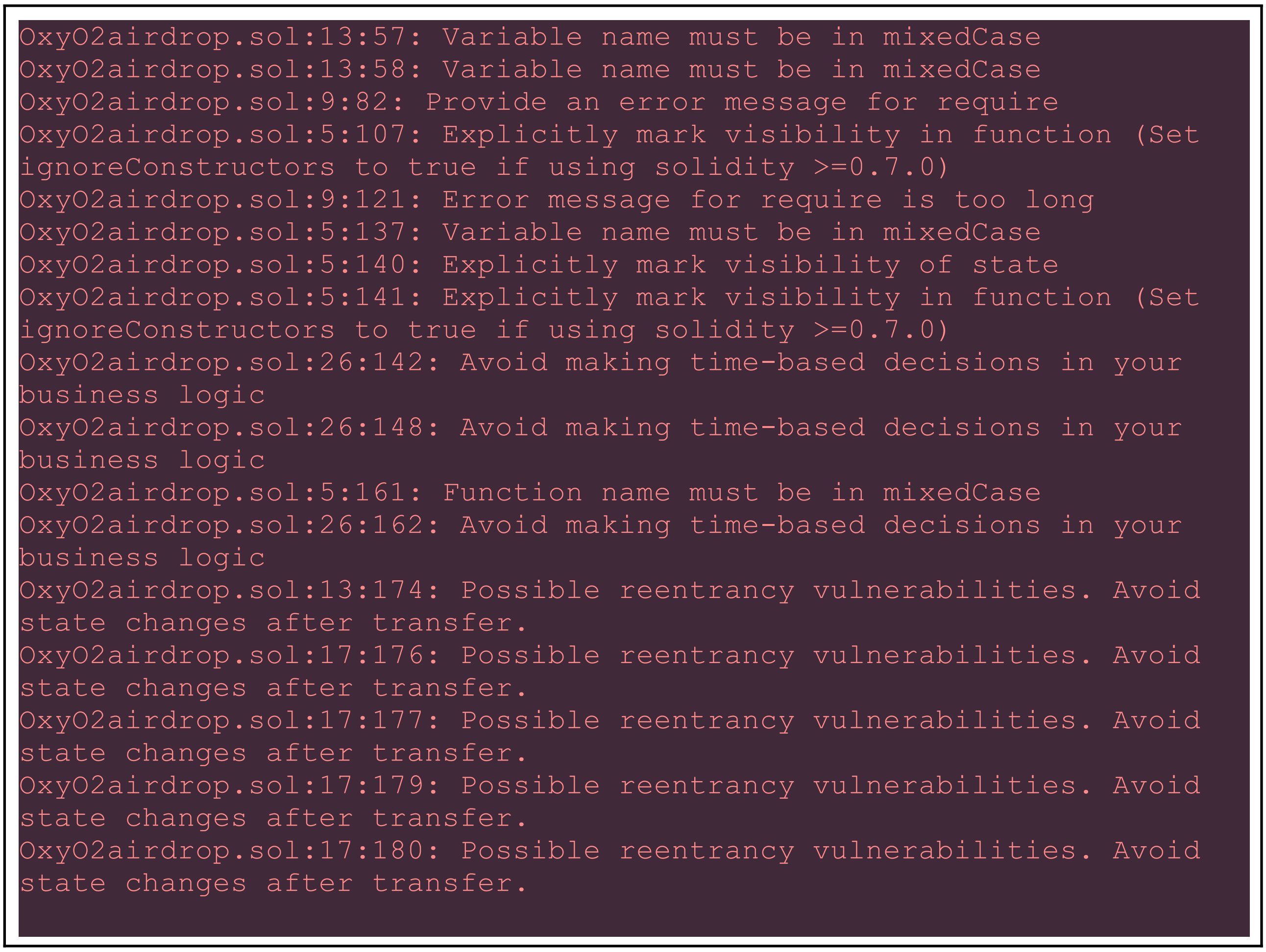


```
OxyO2CommunityAndTeam.sol:9:121: Error message for require is too
long
OxyO2CommunityAndTeam.sol:5:138: Variable name must be in mixedCase
OxyO2CommunityAndTeam.sol:5:148: Explicitly mark visibility of state
OxyO2CommunityAndTeam.sol:5:149: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
OxyO2CommunityAndTeam.sol:26:150: Avoid making time-based decisions
in your business logic
OxyO2CommunityAndTeam.sol:26:155: Avoid making time-based decisions
in your business logic
OxyO2CommunityAndTeam.sol:26:173: Avoid making time-based decisions
in your business logic
OxyO2CommunityAndTeam.sol:26:187: Avoid making time-based decisions
in your business logic
OxyO2CommunityAndTeam.sol:5:202: Function name must be in mixedCase
OxyO2CommunityAndTeam.sol:26:203: Avoid making time-based decisions
in your business logic
OxyO2CommunityAndTeam.sol:13:217: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:219: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:220: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:222: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:223: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:26:232: Avoid making time-based decisions
in your business logic
OxyO2CommunityAndTeam.sol:13:244: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:246: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:247: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:249: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
OxyO2CommunityAndTeam.sol:17:250: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
```

OxyO2airdrop.sol



```
OxyO2airdrop.sol:1:5: Compiler version ^0.8.2 does not satisfy the
^0.5.8 semver requirement
OxyO2airdrop.sol:5:40: Explicitly mark visibility of state
OxyO2airdrop.sol:5:41: Explicitly mark visibility of state
OxyO2airdrop.sol:5:42: Explicitly mark visibility of state
OxyO2airdrop.sol:5:43: Explicitly mark visibility of state
OxyO2airdrop.sol:5:45: Explicitly mark visibility of state
OxyO2airdrop.sol:5:46: Explicitly mark visibility of state
OxyO2airdrop.sol:5:47: Explicitly mark visibility of state
OxyO2airdrop.sol:5:48: Explicitly mark visibility of state
OxyO2airdrop.sol:5:49: Explicitly mark visibility of state
OxyO2airdrop.sol:5:50: Explicitly mark visibility of state
OxyO2airdrop.sol:5:51: Explicitly mark visibility of state
```



```
Oxy02airdrop.sol:13:57: Variable name must be in mixedCase
Oxy02airdrop.sol:13:58: Variable name must be in mixedCase
Oxy02airdrop.sol:9:82: Provide an error message for require
Oxy02airdrop.sol:5:107: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
Oxy02airdrop.sol:9:121: Error message for require is too long
Oxy02airdrop.sol:5:137: Variable name must be in mixedCase
Oxy02airdrop.sol:5:140: Explicitly mark visibility of state
Oxy02airdrop.sol:5:141: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
Oxy02airdrop.sol:26:142: Avoid making time-based decisions in your
business logic
Oxy02airdrop.sol:26:148: Avoid making time-based decisions in your
business logic
Oxy02airdrop.sol:5:161: Function name must be in mixedCase
Oxy02airdrop.sol:26:162: Avoid making time-based decisions in your
business logic
Oxy02airdrop.sol:13:174: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
Oxy02airdrop.sol:17:176: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
Oxy02airdrop.sol:17:177: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
Oxy02airdrop.sol:17:179: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
Oxy02airdrop.sol:17:180: Possible reentrancy vulnerabilities. Avoid
state changes after transfer.
```

Software analysis result:

These software reported many false positive results and some are informational issues. So, those issues can be safely ignored.

EtherAuthority