ARBICHECK

Onyx DAO

Smart Contract Audit

arbicheck.com

arbicheck

Audited on Apr 25 2023

Table of Contents

Audit Summary

Project Name	Onyx DAO
Contract Address	0xB7cD6C8C4600AeD9985d2c0Eb174e0BEe56E8854
Deployer Address	0x26b5a4c280f140344e84d7f32de63958f6ed5760
Website	https://www.onyxdao.finance/
Language	Solidity
Blockchain	Arbitrum
Audit Date	Apr 25 2023

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

This audit report has been prepared by Arbicheck's experts at the request of the client. The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Audit Scope

Arbicheck team was commissioned by Onyx DAO to perform an audit based on the following smart contracts:

https://arbiscan.io/address/0xb7cd6c8c4600aed9985d2c0eb174e0bee56e8854

https://arbiscan.io/address/0xf9c83ff6cf1a9bf2584aa2d00a7297ca8f845cce

https://arbiscan.io/address/0x85aca003b2c6481ca2c4547420fa0e211b51df00

https://arbiscan.io/address/0x586C7D0ced41310C299AC47AdD5d0c3Df8651C0e

https://arbiscan.io/address/0x586C7D0ced41310C299AC47AdD5d0c3Df8651C0e

Note that we **ONLY** audited the code available to us on this URL at the time of the audit. If the URL is not from any block explorer (main net), it may be subject to change. Always check the contract address on this audit report and compare it to the token you are doing research for.

SWC Attack Analysis

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470.

ID	Description	Status
SWC-100	Function Default Visibility	Not Found
SWC-101	Integer Overflow and Underflow	Not Found
SWC-102	Outdated Compiler Version	Not Found
SWC-103	Floating Pragma	Not Found
SWC-104	Unchecked Call Return Value	Not Found
SWC-105	Unprotected Ether Withdrawal	Not Found
SWC-106	Unprotected SELF DESTRUCT Instruction	Not Found
SWC-107	Reentrancy	Not Found
SWC-108	State Variable Default Visibility	Not Found
SWC-109	Uninitialized Storage Pointer	Not Found
SWC-110	Assert Violation	Not Found
SWC-111	Use of Deprecated Solidity Functions	Not Found
SWC-112	Delegatecall to Untrusted Callee	Not Found
SWC-113	DoS with Failed Call	Not Found
SWC-114	Transaction Order Dependence	Not Found
SWC-115	Authorization through tx.origin	Not Found
SWC-116	Block values as a proxy for time	Not Found

SWC-117	Signature Malleability	Not Found
SWC-118	Incorrect Constructor Name	Not Found
SWC-119	Shadowing State Variables	Not Found
SWC-120	Weak Sources of Randomness from Chain Attributes	Not Found
SWC-121	Missing Protection against Signature Replay Attacks	Not Found
SWC-122	Lack of Proper Signature Verification	Not Found
SWC-123	Requirement Violation	Not Found
SWC-124	Write to Arbitrary Storage Location	Not Found
SWC-125	Incorrect Inheritance Order	Not Found
SWC-126	Insufficient Gas Grieng	Found
SWC-127	Arbitrary Jump with Function Type Variable	Not Found
SWC-128	DoS With Block Gas Limit	Not Found
SWC-129	Typographical Error	Not Found
SWC-130	Right-To-Left-Override control character (U+202E)	Not Found
SWC-131	Presence of unused variables	Not Found
SWC-132	Unexpected Ether balance	Not Found
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Not Found
SWC-134	Message call with hardcoded gas amount	Not Found
SWC-135	Code With No Effects	Found
SWC-136	Unencrypted Private Data On-Chain	Not Found

Owner Privileges

- Owner can update the dev fee.
- Owner can change the emission rate.
- Owner can change the dev address.
- Owner can withdraw rewards from the StakingPool contract if there is any emergency.

Risk Classification

Arbicheck uses certain vulnerability levels, these indicate how bad a certain issue is. The higher the risk, the more strictly it is recommended to correct the error before using the contract.

Vulnerability Level	Description
High Issues	These issues will cause the problems and SHOULD be adjusted
Medium Issues	These issues will likely cause the problems and recommended to be adjusted
Low Issues	These issues will not cause any problems, but can be adjusted for the improvement
Notes	Does not compromise the functionality of the contract and just the general recommendations

Gas Optimizations

Notes

Description

1. The MasterChefOnyx contract, which is intended to be OnyxToken's owner, can't use the mint(uint256) function. It can be removed from the OnyxToken contract to reduce the deployed bytecode.

2. The onyx variable should be declared immutable in the MasterChefOnyx contract.

3. No need to restrict duplicated pools since PoolInfo.totalStaked is stored and used during the pool updating in the MasterChefOnyx contract.

4. Unnecessary reads of pool.lastRewardTime and pool.accOnyxPerShare from storage in the updatePool() function in the MasterChefOnyx contract.

5. Multiple reads of user.amount, pool.stakeToken, pool.depositFeeBP, and pool.accOnyxPerShare from storage in the deposit() function in the MasterChefOnyx contract.

6. Multiple reads of user.amount and pool.accOnyxPerShare from storage in the withdraw() function in the MasterChefOnyx contract.

7. Multiple reads of user.amount from storage in the emergencyWithdraw() function in the MasterChefOnyx contract.

8. The _beneficiary and _releaseTime variables should be declared immutable in the OnyxLockedVesting contract.

9. No need in using the public functions releaseTime() and beneficiary() to access private variables for internal calculations in the OnyxLockedVesting contract.

10. The totalAllocPoint and poolInfo.allocPoint are not used since the contract has only one

pool in the StakingPool contract.

11. Unnecessary reads from the storage of the bonusEndTime variable in the setBonusEndTime() function in the StakingPool contract.

12. Multiple reads from the storage of poolInfo.lastRewardTime and totalStaked variables in the pendingReward() function in the StakingPool contract.

13. Multiple reads from the storage of user.amount, poolInfo.accRewardTokenPerShare and STAKE_TOKEN variables in the _depositTo() function in the StakingPool contract.

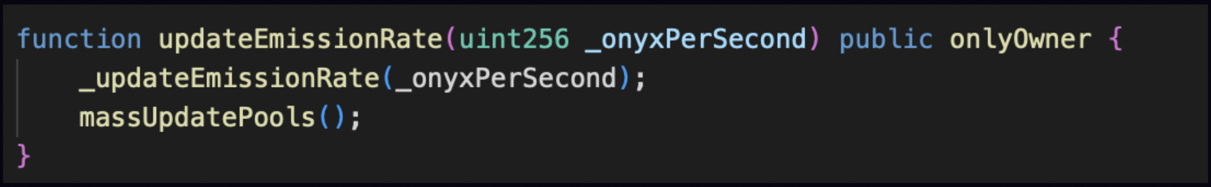
14. Multiple reads from the storage of user.amount and poolInfo.accRewardTokenPerShare variables in the withdraw() function in the StakingPool contract.

Emission rate updating without updating pools

Medium Issue

Description

An update of the pools should be performed prior to changing the emission rate in the updateEmissionRate() function in the MasterChefOnyx contract.



```
function updateEmissionRate(uint256 _onyxPerSecond) public onlyOwner {
    _updateEmissionRate(_onyxPerSecond);
    massUpdatePools();
}
```

Recommendation

Consider updating pools before changing the emission rate to ensure fair reward distribution.

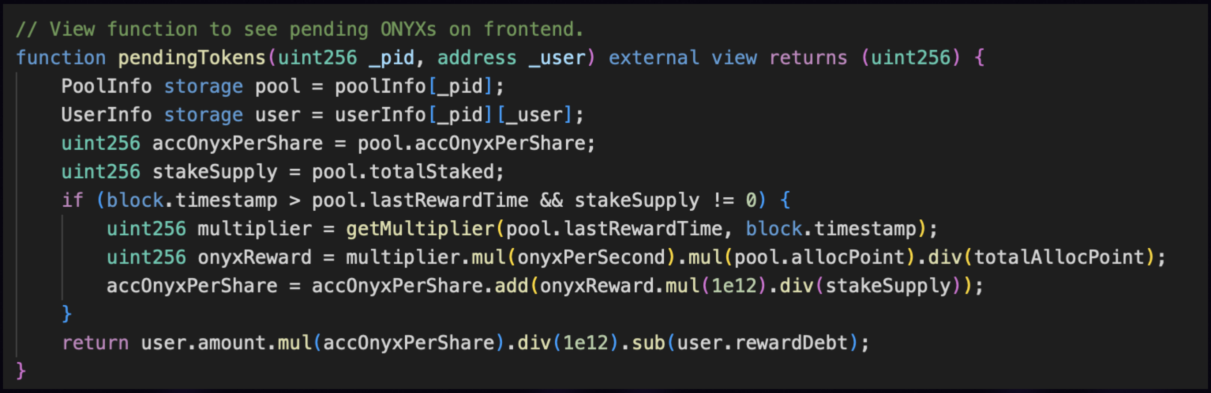
Wrong pending tokens calculation

Medium Issue

Description

In MasterChefOnyx contract function pendingTokens() calculates a user's not claimed rewards.

The function calculates with the assumption that all minted tokens are used for the rewards, but 10% of them are minted to the devaddr address.



```
// View function to see pending ONYXs on frontend.
function pendingTokens(uint256 _pid, address _user) external view returns (uint256) {
    PoolInfo storage pool = poolInfo[_pid];
    UserInfo storage user = userInfo[_pid][_user];
    uint256 accOnyxPerShare = pool.accOnyxPerShare;
    uint256 stakeSupply = pool.totalStaked;
    if (block.timestamp > pool.lastRewardTime && stakeSupply != 0) {
        uint256 multiplier = getMultiplier(pool.lastRewardTime, block.timestamp);
        uint256 onyxReward = multiplier.mul(onyxPerSecond).mul(pool.allocPoint).div(totalAllocPoint);
        accOnyxPerShare = accOnyxPerShare.add(onyxReward.mul(1e12).div(stakeSupply));
    }
    return user.amount.mul(accOnyxPerShare).div(1e12).sub(user.rewardDebt);
}
```

The actual token distribution made in the function updatePool() uses 90% of the minted tokens for the rewards and 10% are minted to the developers' address.

The issue results in a wrong value shown to a user and may implicate integration with other services like farm aggregators.

Recommendation

Fix the calculation for pending tokens using the same formulas as in the updatePool() function.

Lack of checks for input token in the add pool function

Low Issue

Description

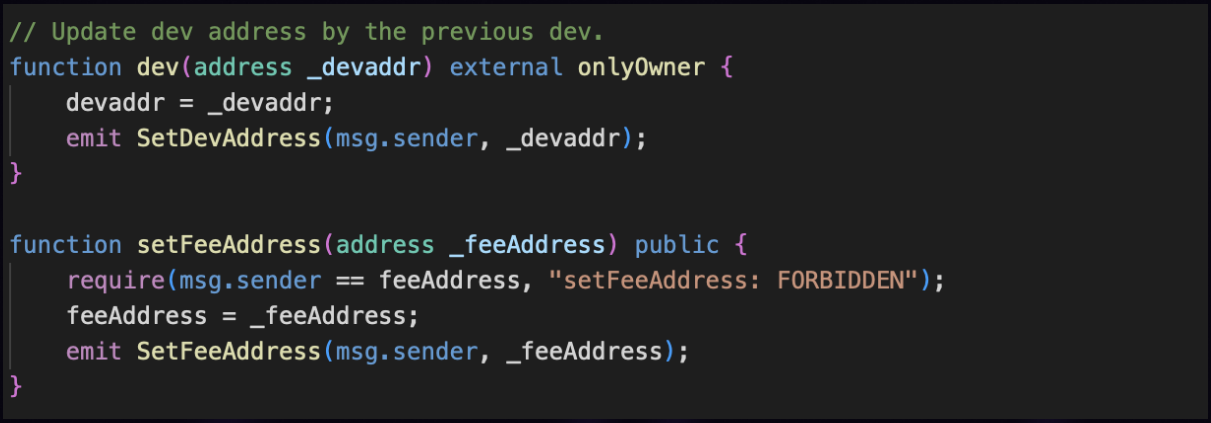
Adding an address to a pool that does not implement IERC20 will break the massUpdate() function. If a new pool is added without massUpdate() the unclaimed user's rewards will be recalculated and decreased.

Lack of zero address checks

Low Issue

Description

The functions dev() , setFeeAddress() lack zero address checks for the input values in the MasterChefOnyx contract.



```
// Update dev address by the previous dev.
function dev(address _devaddr) external onlyOwner {
    devaddr = _devaddr;
    emit SetDevAddress(msg.sender, _devaddr);
}

function setFeeAddress(address _feeAddress) public {
    require(msg.sender == feeAddress, "setFeeAddress: FORBIDDEN");
    feeAddress = _feeAddress;
    emit SetFeeAddress(msg.sender, _feeAddress);
}
```

Rewards source and withdrawals locking

Medium Issue

Description

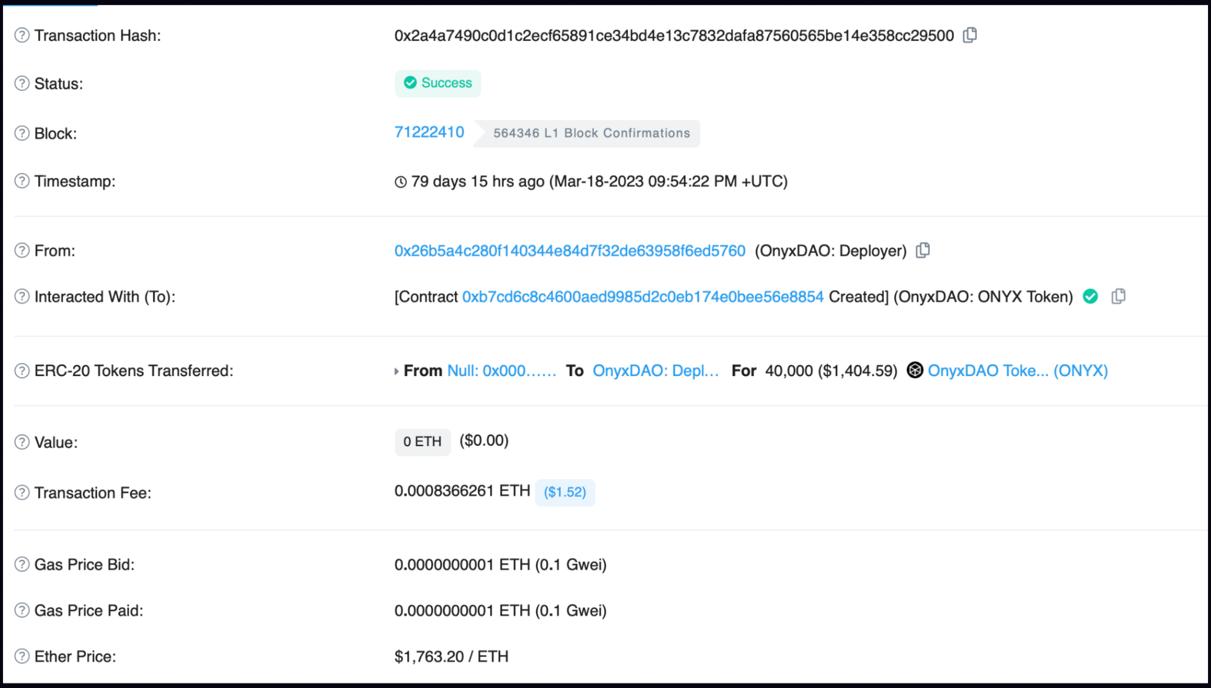
In the StakingPool contract the source of rewards is out of the scope, therefore rewards aren't guaranteed. At the same time the withdraw() function requires a successful transfer of pending rewards to complete a withdrawal.

Recommendation

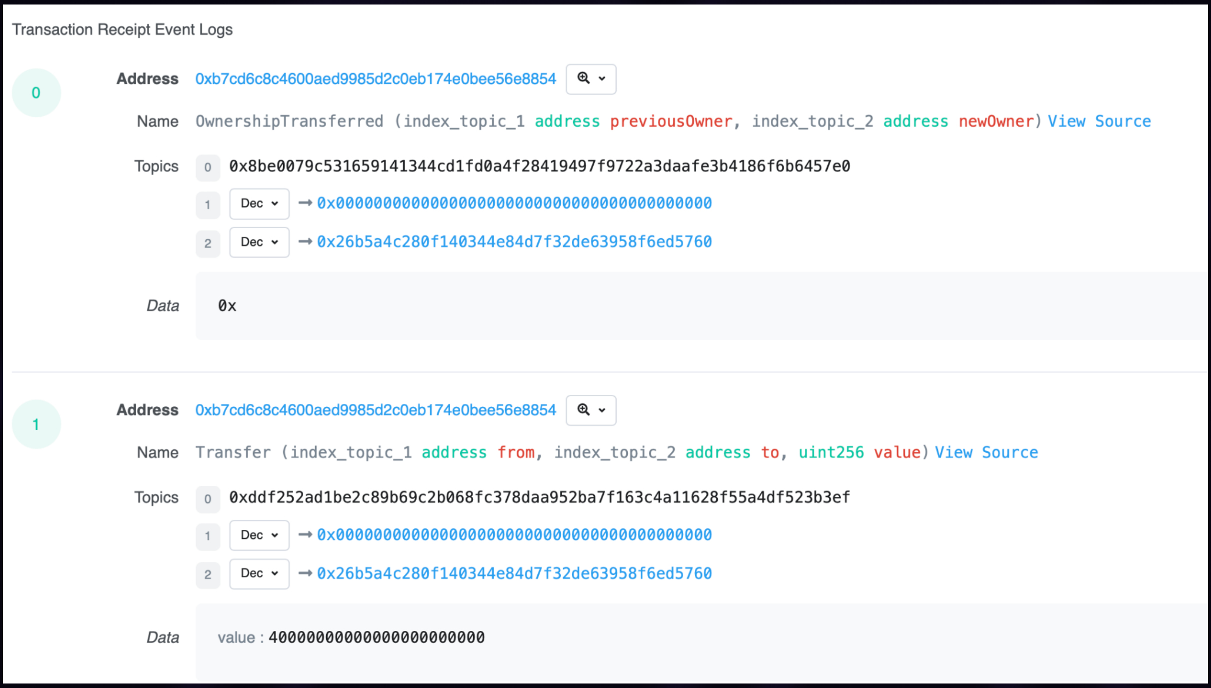
Consider allowing user withdrawals in case of insufficient reward balance by storing pending rewards to be claimed later.

ARBICHECK

Deploy Snapshot



Transaction Hash:	0x2a4a7490c0d1c2ecf65891ce34bd4e13c7832dafa87560565be14e358cc29500
Status:	Success
Block:	71222410 564346 L1 Block Confirmations
Timestamp:	79 days 15 hrs ago (Mar-18-2023 09:54:22 PM +UTC)
From:	0x26b5a4c280f140344e84d7f32de63958f6ed5760 (OnyxDAO: Deployer)
Interacted With (To):	[Contract 0xb7cd6c8c4600aed9985d2c0eb174e0bee56e8854 Created] (OnyxDAO: ONYX Token)
ERC-20 Tokens Transferred:	**From** Null: 0x000…… **To** OnyxDAO: Depl… **For** 40,000 ($1,404.59) OnyxDAO Toke... (ONYX)
Value:	0 ETH ($0.00)
Transaction Fee:	0.0008366261 ETH ($1.52)
Gas Price Bid:	0.0000000001 ETH (0.1 Gwei)
Gas Price Paid:	0.0000000001 ETH (0.1 Gwei)
Ether Price:	$1,763.20 / ETH



Transaction Receipt Event Logs

```
0  Address  0xb7cd6c8c4600aed9985d2c0eb174e0bee56e8854
   Name     OwnershipTransferred (index_topic_1 address previousOwner, index_topic_2 address newOwner) View Source
   Topics   0 0x8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0
            1 Dec → 0x0000000000000000000000000000000000000000
            2 Dec → 0x26b5a4c280f140344e84d7f32de63958f6ed5760
   Data     0x

1  Address  0xb7cd6c8c4600aed9985d2c0eb174e0bee56e8854
   Name     Transfer (index_topic_1 address from, index_topic_2 address to, uint256 value) View Source
   Topics   0 0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef
            1 Dec → 0x0000000000000000000000000000000000000000
            2 Dec → 0x26b5a4c280f140344e84d7f32de63958f6ed5760
   Data     value : 40000000000000000000000
```

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn’t say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Arbicheck and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Arbicheck) owe no duty of care towards you or any other person, nor does Arbicheck make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties, or other terms of any kind except as set out in this disclaimer, and Arbicheck hereby excludes all representations, warranties, conditions, and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Arbicheck hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Arbicheck, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of the use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed. Arbicheck owns all copyright rights to the text, images, photographs, and other content provided in the following document. When using or sharing partly or in full, third parties must provide a direct link to the original document mentioning the author (arbicheck.com).