Cyberscope

Audit Report

Harmony Nodes
Node

April 2022

Github	https://github.com/harmonynodes/harmonynodes
Commit	71007f66ea6f560be6f9533aaeb0bbb4b0b84bfa
Audited by	© cyberscope

Table of Contents

Contract Review

Github	https://github.com/harmonynodes/harmonynodes
Commit	71007f66ea6f560be6f9533aaeb0bbb4b0b84bfa
Contract Name	HoneNode

Audit Updates

Initial Audit	26th April 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9
@openzeppelin/contracts/security/ReentrancyGuard.sol	aa73590d5265031c5bb64b5c0e7f84c44cf5f8539e6d8606b763adac784e8b2e
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadc0fcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/token/ERC721/IERC721.sol	a88e8e63c7a737436f7ec62542620609ab07bb9a772e77146ed4dc98539e03d3
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b306432580 1d73654847a5fb11c58b1e5
contracts/HarmonyLibrary.sol	fe394e4673030daa1fa2ee37d44f41c9a223fe2223f80401f3d59831c758d61e
contracts/Hone_Node.sol	744e7162cbf6213cfe7f44a054b0c2d5d76b2325268c7fe311d02c858136101c
contracts/interfaces/ICommonStruct.sol	747b4cac1a9d9313bbfee2b29add000dc8b36296037270cc5c3ffc874adefed7
contracts/interfaces/IHone_Node.sol	7d852ab677485fc387d2af128c38f3e4ecb41c47165c3e9627ae0db462813a89
contracts/interfaces/IHONE.sol	7c0640d6b05f69c78ece2a10afad3ea67703e4b5994c7f61e5a3e3508da45fdf

contracts/interfaces/IUniswapV2Factory.sol	4158fa477eb2e55aec14343d2e917ab085c71ed068ff2a56a51ee9fa6311879e
contracts/interfaces/IUniswapV2Pair.sol	6a7c6cf1bee1404140c33be5415d887c47a1433869a2f3763c46396e287a52eb
contracts/interfaces/IUniswapV2Router02.sol	8630a0478e76aca1807ded7d149e51b75c7f142e4ad1e3a32df1ea823dc801c5
hardhat/console.sol	27d7e349617dc857b040f2186bf577fe6169ede8bfc98be714ab4289b5793548

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	BLC	Business Logic Concern
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L13	Divide before Multiply Operation

BLC - Business Logic Concern

Criticality	critical
Location	contract.sol#L1

Description

The HoneNode contract implements a vesting-related functionality where users are renting nodes to each other. The implementation contains multiple issues that may cause the contract to not operate properly in the future. We are mentioning some segments.

The following code may underflow if the `brokenCount_` is more than 30 and not a multiplier of 30.

```
while (true) {
   if (brokenCount_ == 0) {
      break;
   }
   if (brokenCount_ >= 30) {
      brokenCount_ -= 30;
      reward = reward * 95**30 / 100**30;
   } else {
      reward = reward * 95**brokenCount_ / 100**brokenCount_;
      brokenCount_ = 0;
   }
}
```

There are arrays that are accessed via unchecked indexes. This may produce unexpectable instances.

```
userInfos[lendOwner_][lendNodeIndex_].rentStatus.rentTime = curTime_;
userInfos[lendOwner_][lendNodeIndex_].rentStatus.rentDeadline = curTime_ +
lendMonths_ * 30 days;
userInfos[lendOwner_][lendNodeIndex_].rentStatus.lendStatus = true;
userInfos[lendOwner_][lendNodeIndex_].rentStatus.offerStatus = false;
```

There are expressions and statements that could be simplified in order to make the business logic more clear and decrease the gas cost.

```
if (node.rentStatus.lendStatus == true) {
   if (node.rentStatus.rentDeadline < curTime) {
      userInfos[sender_][i].rentStatus.lendStatus = false;
   }
}
// could be simplified to
if (node.rentStatus.lendStatus && node.rentStatus.rentDeadline < curTime) {
    userInfos[sender_][i].rentStatus.lendStatus = false;
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

L09 - Dead Code Elimination

Criticality	minor
Location	contracts/HarmonyLibrary.sol#L10

Description

Functions that are not used in the contract, and make the code's size bigger.

```
_calcAmount
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contracts/HarmonyLibrary.sol#L130 contracts/Hone_Node.sol#L24,71,96,114,136,200,214,236

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
node.ownable == true
node.rentStatus.lendStatus == true
node.rentStatus.borrowStatus == true
userInfos[userAddress_][i].burned == false
require(bool,string)(userInfos[sender_][nodeIndex_].rentStatus.offerStatus ==
false,already listed)
userInfos[sender_][i].burned == false &&
HarmonyLibrary._getLendStatus(userInfos[sender_][i].rentStatus,block.timestamp)
== false
require(bool,string)(userInfos[sender_].length > nodeID_ ||
HarmonyLibrary._getLendStatus(userInfos[sender_][nodeID_].rentStatus,block.times
tamp) == false,wrong node)
require(bool,string)(userInfos[sender_][useNodes_[i]].rentStatus.lendStatus ==
false && userInfos[sender_][useNodes_[i]].rentStatus.offerStatus == false &&
userInfos[sender_][useNodes_[i]].rentStatus.borrowStatus == false,rent node
exist)
rentStatus_.lendStatus == false || (rentStatus_.lendStatus == true &&
rentStatus_.rentDeadline <= curTime_)
...
```

Recommendation

Remove the equality to the boolean constant.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contracts/HarmonyLibrary.sol#L10,23

Description

Performing divisions before multiplications may cause lose of prediction.

```
reward = reward * 95 ** 30 / 100 ** 30
reward = uint256(nodePrice_) * 1e18 * uint256(nodePercent_) / 1e4
liquidity = honeAmount_ / 10
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ReentrancyGuard	Implementation			
	<Constructor>	Public	✓	-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-

	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC165	Interface			
	supportsInterface	External		-
HarmonyLibrary	Library			
	_calcAmount	Internal		
	_getRewardAmount	Internal		
	_getGeneralReward	Internal		
	_getLendReward	Internal		
	_getLendStatus	Internal		
HoneNode	Implementation	Ownable, ReentrancyGuard, IHoneNode		
	claimRewards	External	✓	onlyOwner
	createNode	External	✓	onlyOwner
	upgradeNode	External	✓	onlyOwner
	payMaintenanceFee	External	✓	onlyOwner
	payAllMaintenanceFee	External	✓	onlyOwner
	listLendOffer	External	✓	onlyOwner
	closeLendOffer	External	✓	onlyOwner
	acceptLendOffer	External	✓	onlyOwner
	getClaimableRewards	External		onlyOwner
	getNodeCount	Public		onlyOwner
	getNodes	External		onlyOwner
	_calcClaimableRewards	Internal		
	_createNode	Internal	✓	
ICommonStruc	Interface			

t				
IHoneNode	Interface	ICommonStruct		
	claimRewards	External	✓	-
	getClaimableRewards	External		-
	createNode	External	✓	-
	upgradeNode	External	✓	-
	payMaintenanceFee	External	✓	-
	payAllMaintenanceFee	External	✓	-
	listLendOffer	External	✓	-
	closeLendOffer	External	✓	-
	acceptLendOffer	External	✓	-
	getNodeCount	External		-
	getNodes	External		-
IHONE	Interface	IERC20		
	setSaleFee	External	✓	-
	setTransferFee	External	✓	-
	setFeeCollectWallet	External	✓	-
	setNodeManagementContract	External	✓	-
	enableBlacklist	External	✓	-
	disableBlacklist	External	✓	-
	isBlacklisted	External		-
	mint	External	✓	-
	burn	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-

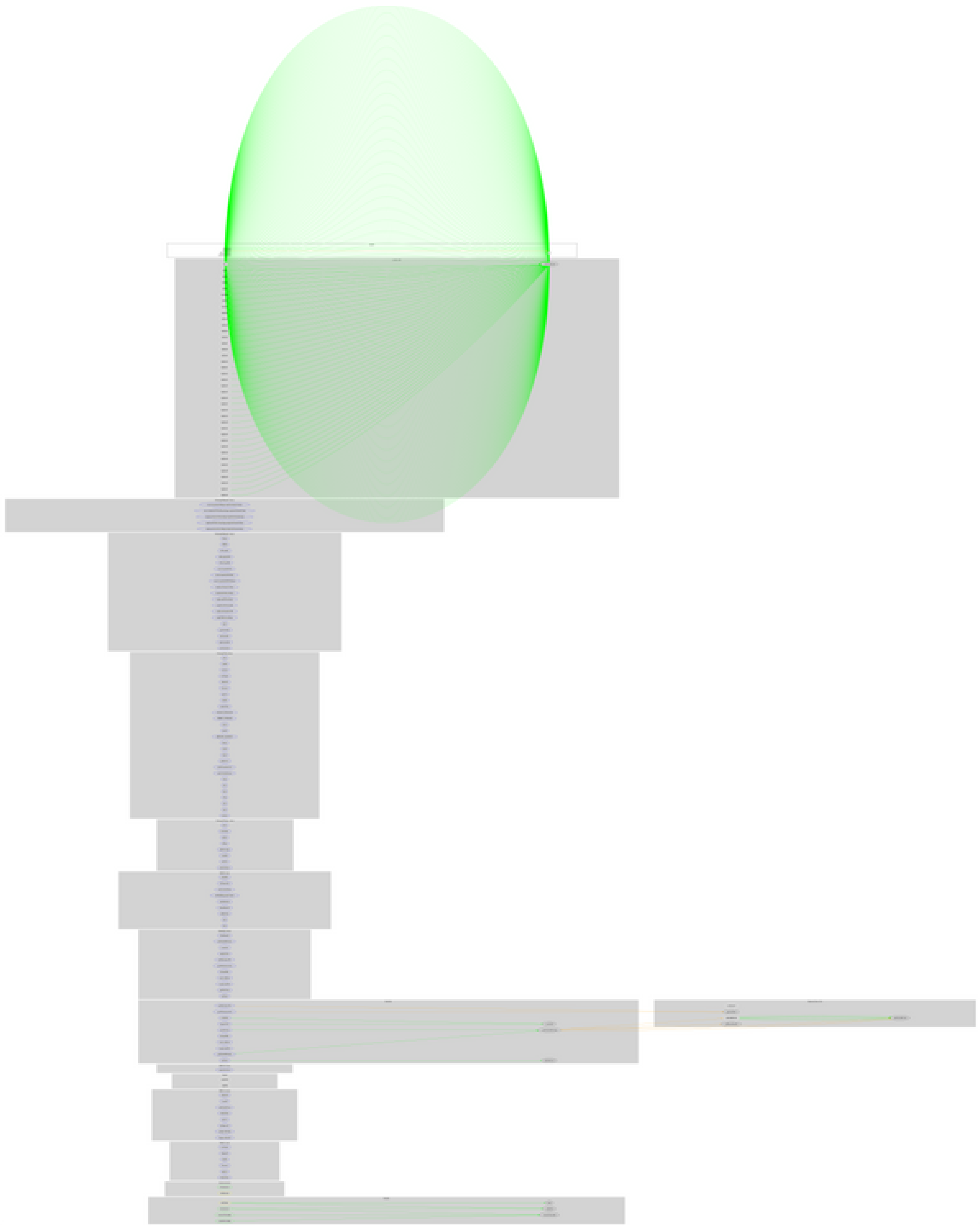
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-

	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
console	Library			
	_sendLogPayload	Private		
	log	Internal		
	logInt	Internal		
	logUint	Internal		
	logString	Internal		

	logBool	Internal		
	logAddress	Internal		
	logBytes	Internal		
	logBytes1	Internal		
	logBytes2	Internal		
	logBytes3	Internal		
	logBytes4	Internal		
	logBytes5	Internal		
	logBytes6	Internal		
	logBytes7	Internal		
	logBytes8	Internal		
	logBytes9	Internal		
	logBytes10	Internal		
	logBytes11	Internal		
	logBytes12	Internal		
	logBytes13	Internal		
	logBytes14	Internal		
	logBytes15	Internal		
	logBytes16	Internal		
	logBytes17	Internal		
	logBytes18	Internal		
	logBytes19	Internal		
	logBytes20	Internal		
	logBytes21	Internal		
	logBytes22	Internal		
	logBytes23	Internal		
	logBytes24	Internal		
	logBytes25	Internal		
	logBytes26	Internal		
	logBytes27	Internal		
	logBytes28	Internal		
	logBytes29	Internal		
	logBytes30	Internal		
	logBytes31	Internal		
	logBytes32	Internal		

	log	Internal		

Contract Flow



Summary

Harmony Nodes Node implements a vesting-related functionality. All the essential contract methods are only accessed by the contract owner. The contract owner has the ability to attach nodes on the users. The users can rent nodes to each other. The users receive rewards proportionally to their holdings and the time period that has elapsed. This audit mentions some business logic and implementation conserns.

Disclaimer

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io