Audit Report

Paranodes

January 2022

Type	ERC20
Network	FTM
Address	0x3AB0482425c3a5BaC7Ef59f4a39A194FEE13a2c7
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	Paranodes
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	
Explorer	https://testnet.ftmscan.com/address/0x3AB0482425c3a5BaC7Ef59f4a39A194FEE13a2c7#code
Symbol	PARA
Decimals	18
Total Supply	20,456,743
Website	https://paranodes.finance/

Audit Updates

Initial Audit	1st of January 2022
Corrected	

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	medium
Location	contract.sol#L2167

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `maxTxAmount` to zero.

```
require(
    amount <= maxTxAmount,
    "Please transfer under the max transaction amount"
);
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#code#L2087,L2092,L2097

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the updateRewardsFee function with a high percentage value.

```
function updateRewardsFee(uint256 value) external onlyOwner {
     rewardsFee = value;
     totalFees = rewardsFee.add(liquidityPoolFee).add(futurFee);
 }
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	critical
Location	contract.sol#L2479

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been swiped from the swap & liquify feature. The owner may take advantage of it by calling the `manualsend` function after swapping tokens in `manualswap`.

```
    function manualsend(uint256 amount) public onlyOwner {
        if (amount > address(this).balance) amount = address(this).balance;
        payable(owner()).transfer(amount);
    }
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L2154

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `blacklistMalicious` function.

```
        require(
             !_isBlacklisted[from] && !_isBlacklisted[to],
             "Blacklisted address"
         );
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L2469,L2463,L2457 and 39 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
getTotalNodesCreated
getNodesLastClaims
getNodesRewards
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1961,L1610

Description

Constant state variables should be declared constant to save gas.

```
deadWallet
distribution
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1976,L1975,L2484 and 22 more

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_isExcluded
_isBlacklisted
_i
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L152,L136,L1278 and 10 more

Description

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

```
mod
safeTransferFrom
safeIncreaseAllowance
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1638

Description

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

```
require(bool,string)(_managers[msg.sender] == true,Only managers can call this function)
```

Recommendation

Remove the equality to the boolean constant.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L2413,L2110,L2106 and 5 more

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
maxNodesAmount = newMaxNodesAmount
maxTxAmount = value
rwSwap = value
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
SpookySwapRouter01	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		-
SpookySwapRouter02	Interface	SpookySwapRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	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	✓	-
SpookySwapFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	migrator	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
	setMigrator	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20	Implementation	IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		

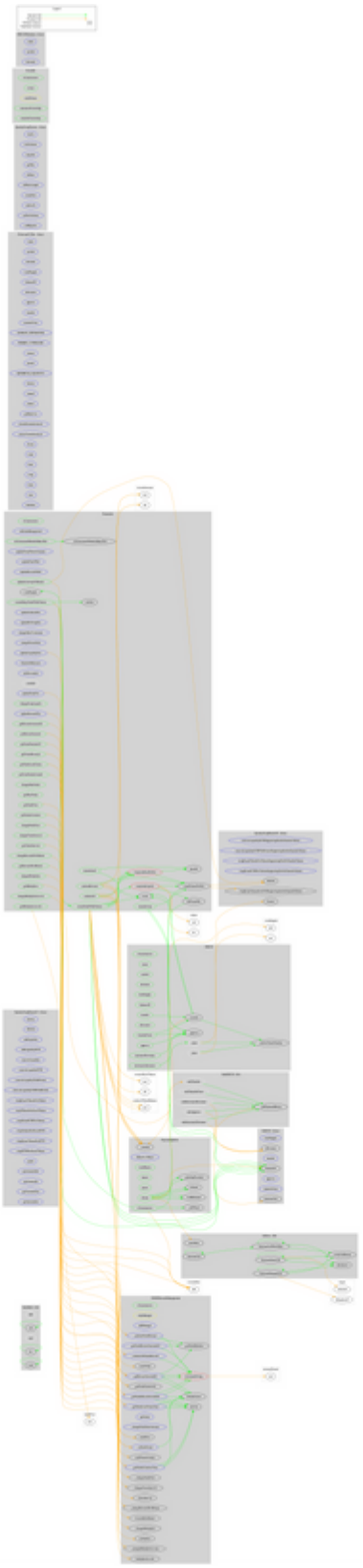
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
PaymentSplitter	Implementation			
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	totalShares	Public		-
	totalReleased	Public		-
	totalReleased	Public		-
	shares	Public		-
	released	Public		-
	released	Public		-
	payee	Public		-
	release	Public	✓	-
	release	Public	✓	-
	_pendingPayment	Private		
	_addPayee	Private	✓	
NODERewardManagement	Implementation			
	<Constructor>	Public	✓	-
	addManager	External	✓	onlyManager
	createNode	External	✓	onlyManager
	dividendsOwing	Private		
	_getNodeByIndex	Private		
	_cashoutNodeReward	External	✓	onlyManager

	_cashoutAllNodesReward	External	✓	onlyManager
	_getRewardAmountOf	External		-
	_getRewardAmountOf	External		-
	_getNodeRewardAmountOf	External		-
	_getNodesCreationTime	External		-
	_getNodesRewardAvailable	External		-
	_getNodesLastClaimTime	External		-
	getNodes	External		-
	uint2str	Internal		
	_changeNodeStartAmount	External	✓	onlyManager
	_changeNodePrice	External	✓	onlyManager
	_changeRewardsPerMinute	External	✓	onlyManager
	_changeMultipliers	External	✓	onlyManager
	_changeMultipliersLevels	External	✓	onlyManager
	_changeClaimInterval	External	✓	onlyManager
	_getNodeNumberOf	Public		-
	isNodeOwner	Private		
	_isNodeOwner	External		-
Paranodes	Implementation	ERC20, Ownable, PaymentSplitter		
	<Constructor>	Public	✓	ERC20 PaymentSplitter
	setNodeManagement	External	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	updateSwapTokensAmount	External	✓	onlyOwner
	updateFuturWall	External	✓	onlyOwner
	updateRewardsWall	External	✓	onlyOwner
	updateRewardsFee	External	✓	onlyOwner
	updateLiquidityFee	External	✓	onlyOwner
	updateFuturFee	External	✓	onlyOwner
	updateCashoutFee	External	✓	onlyOwner
	updateRwSwapFee	External	✓	onlyOwner
	changeMaxTxAmount	External	✓	onlyOwner

	changeProtectSale	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	blacklistMalicious	External	✓	onlyOwner
	setIsExcluded	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	_transfer	Internal	✓	
	swapAndSendToFee	Private	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	createNodeWithTokens	Public	✓	-
	createManyNodeWithTokens	Public	✓	-
	cashoutReward	Public	✓	-
	cashoutAll	Public	✓	-
	changeSwapLiquify	Public	✓	onlyOwner
	getNodeNumberOf	Public		-
	getRewardAmountOf	Public		onlyOwner
	getRewardAmount	Public		-
	changeNodePrice	Public	✓	onlyOwner
	getNodePrice	Public		-
	changeClaimInterval	Public	✓	onlyOwner
	getClaimInterval	Public		-
	changeMaxNodes	Public	✓	onlyOwner
	getMaxNodes	Public		-
	changeRewardsPerMinute	Public	✓	onlyOwner
	getRewardsPerMinute	Public		-
	changeMultipliers	Public	✓	onlyOwner
	getMultipliers	Public		-
	changeMultipliersLevels	Public	✓	onlyOwner
	getMultipliersLevels	Public		-
	getNodesCreatime	Public		-
	getNodesRewards	Public		-
	getNodesLastClaims	Public		-
	getTotalNodesCreated	Public		-
	manualswap	Public	✓	onlyOwner

	manualsend	Public	✓	onlyOwner
	uint2str	Internal		

Contract Flow



Domain Info

Domain Name	paranodes.finance
Registry Domain ID	cbfad2467b6648c695b67a94dd48a748-DONUTS
Creation Date	2021-12-12T22:40:48Z
Updated Date	2021-12-28T20:23:18Z
Registry Expiry Date	2022-12-12T22:40:48Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 19 days before the creation of the audit. It will expire in 12 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Paranodes is a Phantom powered DAAS token that allows users to gain passive income. The users can create nodes that generate $PARA every minute of the day. The more nodes the user has, the higher daily $PARA reward is. The token has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees, blacklisting addresses and transferring funds to the team's wallet. There are some informative comments that do not affect the contract security. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Coinscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Coinscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Coinscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

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

The Coinscope.co team

https://www.coinscope.co