COINSCOPE

Audit Report

Fractional Rocket Dividend Distributor

February 2022

Source	Github
commit	186bee2a13455ec944fb73b8922ea971a05aa52b
Audited by	© coinscope

Table of Contents

Contract Review

Github	https://github.com/Fr0ck/audit
Commit	186bee2a13455ec944fb73b8922ea971a05aa52b
File	DividenDistributor/DividenDistributorV1.sol

Audit Updates

Initial Audit	16h of February 2022
Corrected	

Contract Analysis

The dividend distributor is a mechanism that shares rewards to the token holders. The users that hold the “mainToken” are rewarded proportional to their holdings quantity. The reward price is paid in the main network’s coin. The reward process is taking place in rounds.

The contract owner is responsible to set the rewards rounds and the amount that is going to be shared. There is no limit in regards to the rewards amount. Users can claim their rewards manually. They can claim their rewards even if the rewards round has elapsed.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	DSS	Data Structure Suggestion

L01 - Public Function could be Declared External

Criticality	minor
Location	DividenDistributor/DividenDistributorV1.sol#L87,110,151,188,198

Description

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

```
excludedFromReward
setMainToken
claimReward
shareReward
swapAndShareReward
```

Recommendation

Use the external attribute for functions never called from the contract

Data Structure Suggestion

Criticality	minor
Location	DividenDistributor/DividenDistributorV1.sol#L87,110,151,188,198

Description

The variable rewardClaimed the Reward.isExcludedFromReward follows exactly the same convention. They are an array that maps the reward id to a reward array that contains an array with information per address. While they follow the same principle, they have a different structure.

```
mapping(uint256 => Reward) public rewards;
mapping(uint256 => mapping(address => bool)) rewardClaimed;
```

Recommendation

The Reward structure could contain an additional property that stores the “rewardClaimed” information.

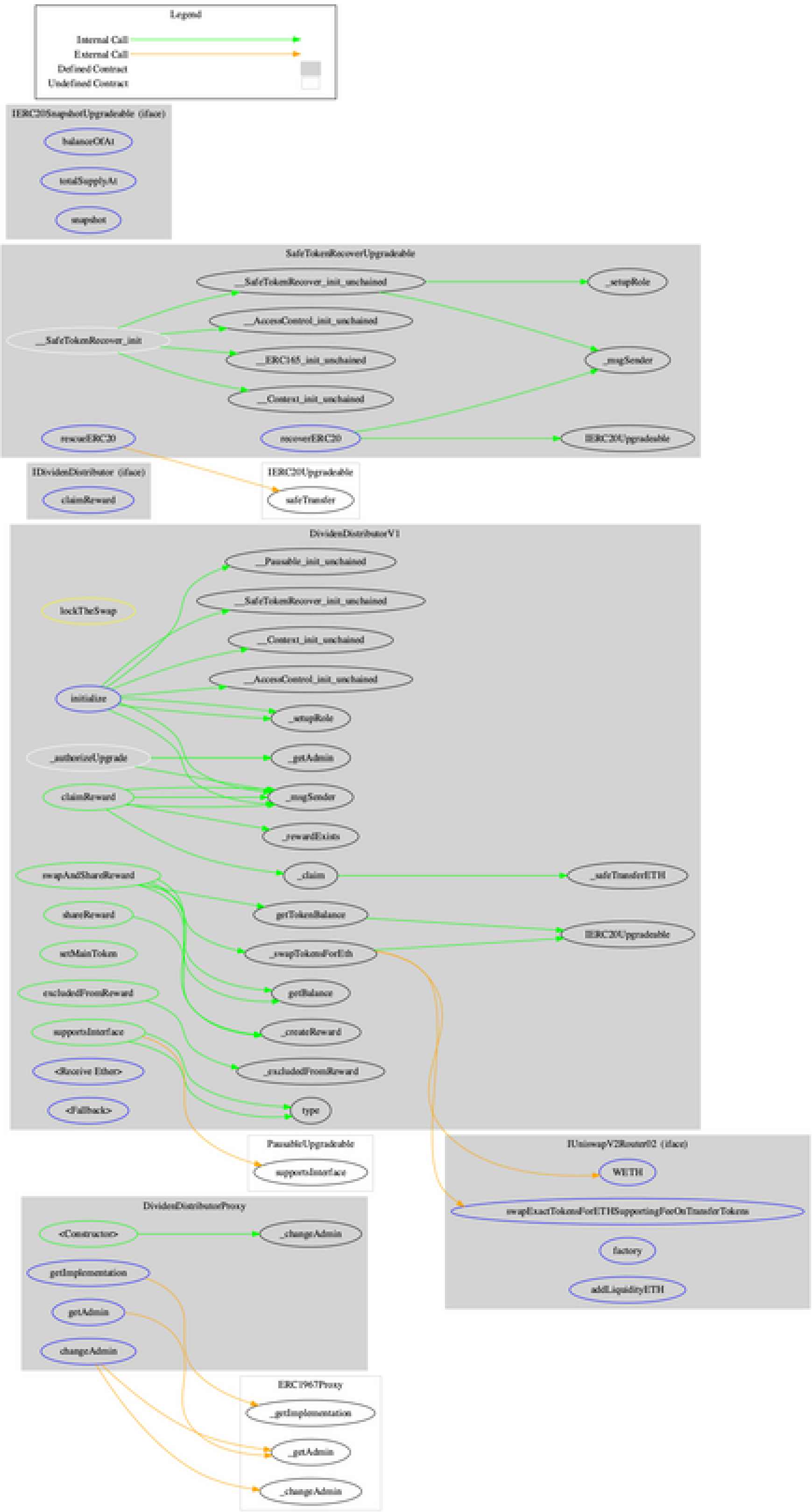
```
struct Reward {
    uint256 rewardAmount;
    uint256 totalClaimed;
    uint256 issuedAt;
    uint256 snapshotId;
    uint256 totalExcludedFromSupply;
    uint8 rewardSource;
    mapping(address => bool) isExcludedFromReward;
    mapping(address => bool) rewardClaimed;
}
```

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
DividenDistributorProxy	Implementation	ERC1967Proxy		
	<Constructor>	Public	Payable	ERC1967Proxy
	getImplementation	External		-
	getAdmin	External		-
	changeAdmin	External	✓	-
DividenDistributorV1	Implementation	IDividenDistributor, Initializable, UUPSUpgradeable, AccessControlUpgradeable, SafeTokenRecoverUpgradeable, PausableUpgradeable		
	initialize	External	✓	initializer
	_authorizeUpgrade	Internal	✓	
	swapAndShareReward	Public	✓	onlyRole
	shareReward	Public	Payable	onlyRole
	_createReward	Internal	✓	
	_swapTokensForEth	Private	✓	lockTheSwap
	claimReward	Public	✓	-
	_claim	Internal	✓	
	_rewardExists	Internal		
	getTokenBalance	Public		-
	getBalance	Public		-
	setMainToken	Public	✓	onlyRole
	_safeTransferETH	Internal	✓	

	excludedFromReward	Public	✓	onlyRole
	_excludedFromReward	Internal	✓	
	supportsInterface	Public		-
	<Receive Ether>	External	Payable	-
	<Fallback>	External	Payable	-
IDividenDistributor	Interface			
	claimReward	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
SafeTokenRecoverUpgradeable	Implementation	Initializable, AccessControlUpgradeable		
	__SafeTokenRecover_init	Internal	✓	initializer
	__SafeTokenRecover_init_unchained	Internal	✓	initializer
	recoverERC20	External	✓	onlyRole
	rescueERC20	External	✓	onlyRole
IERC20SnapshotUpgradeable	Interface			
	balanceOfAt	External		-
	totalSupplyAt	External		-
	snapshot	External	✓	-

Contract Flow



Summary

The Smart Contract analysis reported no compiler error or critical issues. The contract cannot interrupt the user’s transactions. The main functionality of the contract is the reward distribution. In this audit we focus on the business logic of the distributor, security investigation and suggest some improvements.

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