



Audit Report

Fractional Rocket Dividend Distributor

February 2022

Source Github
commit 573b1c6536c35e0fb9e8b03bd00339ee839cc645
Audited by © coinscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Contract Analysis	2
Contract Diagnostics	4
CO - Code Optimization	5
Description	5
Recommendation	5
L14 - Uninitialized Variables in Local Scope	6
Description	6
Recommendation	6
Contract Functions	7
Contract Flow	9
Summary	10
Disclaimer	11
About Coinscope	12

Contract Review

Github	https://github.com/Fr0ck/audit
Commit	573b1c6536c35e0fb9e8b03bd00339ee839cc645
File	DividenDistributor/DividenDistributorV1.sol

Audit Updates

Initial Audit	16h of February 2022
Corrected	18h of February 2022

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
●	CO	Code Optimization
●	L04	Conformance to Solidity Naming Conventions

CO - Code Optimization

Criticality	minor
Location	contract.sol#L299,325

Description

There are segments where the *storage* keyword is defined even if the variable is used for read-only purposes.

```
function getTotalUnclaimedReward(address holder, uint8 rewardSource) external
view returns (uint256 totalUnclaimedReward) {
    uint256[] memory rewardIdsUnclaimed = getRewardIdsUnclaimed(holder,
rewardSource);
    for(uint i = 0 ; i < rewardIdsUnclaimed.length; i++) {
        uint256 rewardId = rewardIdsUnclaimed[i];
        Reward storage reward = rewards[rewardId];
        (,uint256 rewardAmount) = _calculateRewardAmount(
            reward.snapshotId,
            reward.totalExcludedFromSupply,
            reward.rewardAmount,
            holder
        );
        totalUnclaimedReward += rewardAmount;
    }
}
```

Recommendation

The memory keyword could be used when accessing state variables for read-only purposes.

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

build/contract/DividenDistributor/DividenDistributorV1.sol#L160

Description

These are variables that are defined in the local scope and are not initialized.

```
totalRewardAmount
```

Recommendation

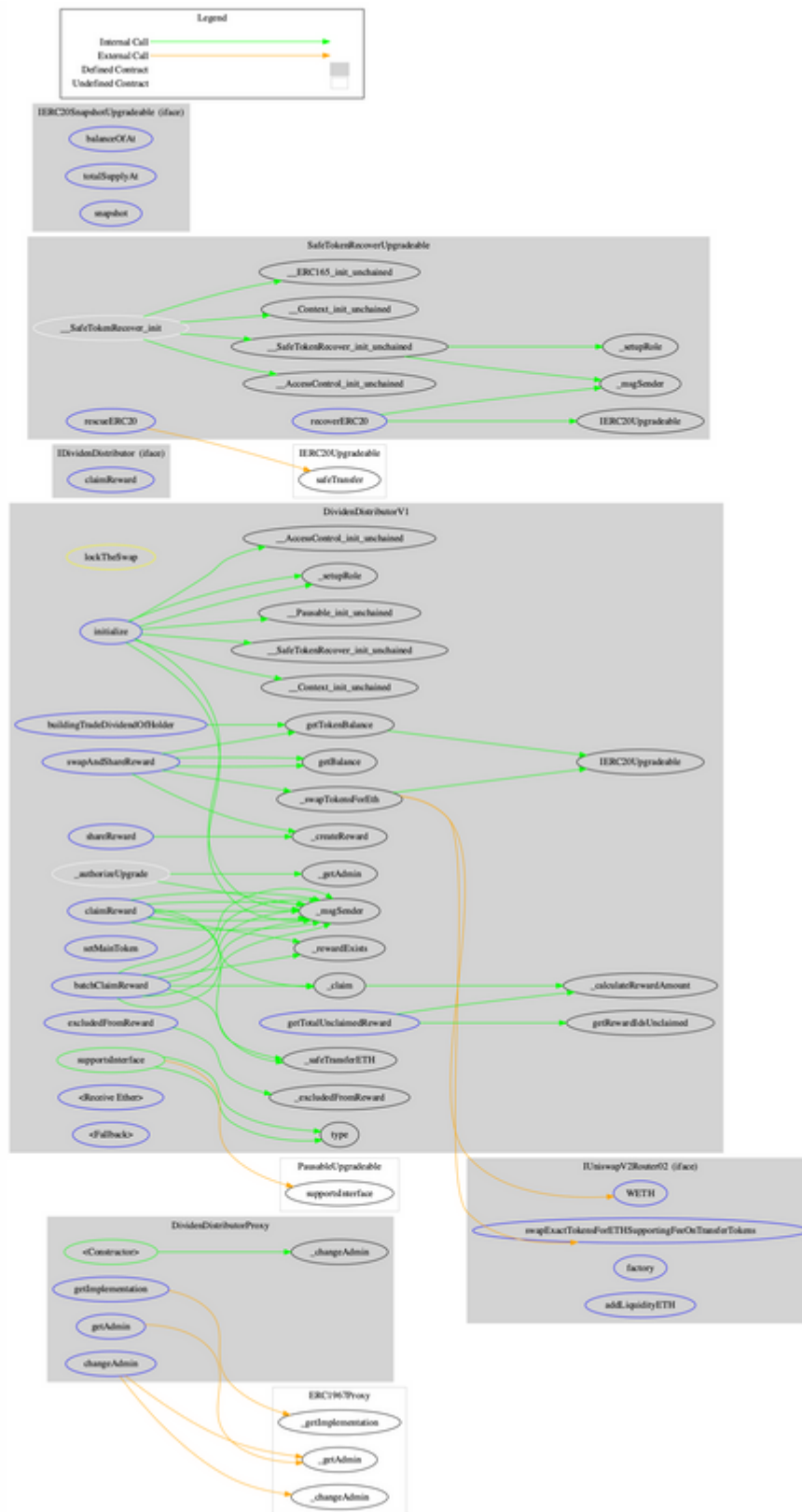
All the local scoped variables should be initialized.

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	External	✓	onlyRole
	shareReward	External	Payable	onlyRole
	_createReward	Internal	✓	
	_swapTokensForEth	Private	✓	lockTheSwap
	claimReward	External	✓	-
	batchClaimReward	External	✓	-
	_claim	Internal	✓	
	_calculateRewardAmount	Internal		
	_rewardExists	Internal		
	getTokenBalance	Public		-
	getBalance	Public		-

	setMainToken	External	✓	onlyRole
	_safeTransferETH	Internal	✓	
	excludedFromReward	External	✓	onlyRole
	_excludedFromReward	Internal	✓	
	buildingTradeDividendOfHolder	External		-
	getRewardIdsUnclaimed	Public		-
	getTotalUnclaimedReward	External		-
	supportsInterface	Public		-
	<Receive Ether>	External	Payable	-
	<Fallback>	External	Payable	-
IDividendDistributor	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>