



Audit Report

NUTGAIN Staking

April 2022

Type	BEP20
Network	BSC
Address	0xF1477207249C4429a0B872dFE91E276176DCedff
Audited by	© cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Diagnostics	4
Notes	5
Data Structure Optimization	6
Description	6
Recommendation	6
CO - Code Optimization	7
Description	7
Recommendation	7
STC - Succeeded Transfer Check	8
Description	8
Recommendation	8
CR - Code Repetition	9
Description	9
Recommendation	9
MC - Missing Check	10
Description	10
Recommendation	10
L01 - Public Function could be Declared External	11
Description	11
Recommendation	11
L02 - State Variables could be Declared Constant	12
Description	12

Recommendation	12
L04 - Conformance to Solidity Naming Conventions	13
Description	13
Recommendation	13
L07 - Missing Events Arithmetic	14
Description	14
Recommendation	14
L09 - Dead Code Elimination	15
Description	15
Recommendation	15
L13 - Divide before Multiply Operation	16
Description	16
Recommendation	16
L14 - Uninitialized Variables in Local Scope	17
Description	17
Recommendation	17
Contract Functions	18
Contract Flow	21
Summary	22
Disclaimer	23
About Cyberscope	24



Contract Review

Contract Name	NutGainSingleStaking
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xf1477207249c4429a0b872dfe91e276176dcedff

Source Files

Filename	SHA256
contract.sol	1d7015d97935ed3fc35fbd55165d192b66105ecde365fbd4642989491e0dcca8

Audit Updates

Initial Audit	21st April 2022
Corrected	

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	DSO	Data Structure Optimization
●	CO	Code Optimization
●	STC	Succeeded Transfer Check
●	CR	Code Repetition
●	MC	Missing Check
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

Notes

- if $\text{tokenSupply} * \text{apy}$ is less than 1051200000, then no rewards will be distributed since the division will be less than 1.
- A user can withdraw the deposited amount with two ways. The `emergencyWithdraw()` that receives the entire amount, and the `withdraw()` that receives the amount redacted by an early withdrawal tax. The users have no reason to withdraw tokens using the `withdraw()` method since they have the option to call the `emergencyWithdraw()` method