Cyberscope

Audit Report

BNB Park

April 2022

Type	BEP20
Network	BSC
Address	0x3837155448d85E9FE132a9fc721c5C417a7FFB07
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BNBPark
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x3837155448d85E9FE132a9fc721c5C417a7FFB07

Source Files

Filename	**SHA256**
contract.sol	a07436314697de0c88953dd2ecc60f7d491d6fc2f89ce890e93536123af9ec08

Audit Updates

Initial Audit	28th April 2022
Corrected	

Contract Analysis

- The users have the ability to buy eggs by paying in the native currency.
- The price of eggs depends on some variations like the current egg supply and the Contract's native currency balance.
- The buy and sell amount is taxed by an admin fee, the taxed amount is moved directly to dev’s wallet.
- The users gathered eggs in order to redeem miners.
- The redeem process is called "hatch".
- During the hatch process the referred user takes a percentage of the user's eggs as a reward.

Contract Owner Privileges

- The contract owner has the authority to manipulate the admin fee.
- The contract owner has the authority to manipulate referral fees.
- The contract owner has the authority to change the minimum period for the hatched vesting.
- The contract owner has the authority to change the dev’s wallet.

Contract Diagnostics

🔴 Critical 🟠 Medium ● Minor

Severity	Code	Description
●	CBD	Contract Balance Dependency
●	IAD	Initial Amount Distribution
🟠	MC	Missing Check
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic

Contract Balance Dependency

Criticality	minor
Location	contract.sol#L351

Description

The calculation of the sell and buy price heavily depends on the Contract's balance. That means that the same amount of eggs can be bought and sold at quite different prices according to the contract's balance. This calculation may be abused by the users and produce unexpected results in the financial ecosystem.

Below is the calculated eggs quantity as a result of the amount, contract balance and egg supply:

Amount	Contract Balance	Supply	Result
1	1000000	108000000000	107999.8
10	1000000	108000000000	1079989.2
100	1000000	108000000000	107892107.8

The following is the same amounts with different contract balance:

Amount	Contract Balance	Supply	Result
1	1000	108000000000	107892107.8
10	1000	108000000000	857142857.1
100	1000	108000000000	9818181818.1

Recommendation

The contract could exclude the contract’s balance from the price calculations or use a weight in the calculations so it cannot heavily affect the prices.

Initial Amount Distribution

Criticality	minor
Location	contract.sol#L341

Description

The price calculations depend on the initial contract’s funds.

For instance, if the contract’s funds are less than the acquisition funds, then the purchase will not be able to complete since the calculation will underflow.

```
SafeMath.sub(address(this).balance,msg.value)
```

Recommendation

The contract should check if the contract’s amount is sufficient in order to proceed with the buy and sell methods.

MC - Missing Check

Criticality	medium
Location	contract.sol#L377,382

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The adminFeeVal and refFeeVal are used as percentage variables. If the variables are configured with a high value, the transaction will not be able to proceed.

```
function setAdminFeeVal(uint256 _adminFeeVal) public onlyOwner {
    require(_adminFeeVal > 0);
    adminFeeVal = _adminFeeVal;
}

function setRefFeeVal(uint256 _refFeeVal) public onlyOwner {
    require(_refFeeVal > 0);
    refFeeVal = _refFeeVal;
}
```

Recommendation

The contract should properly check the variables according to the required specifications

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L331,346

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
recAdd.transfer(fee);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L255,265,270,323,335,341,363,371,377,382,387,392,396,400

Description

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

```
getMyMiners
getBalance
changeAdmin
setRewardRate
setRefFeeVal
setAdminFeeVal
seedMarket
calculateEggBuySimple
buyEggs
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L377,382,387,392,284

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.

```
EGGS_TO_HATCH_1MINERS
_admin
_rate
_refFeeVal
_adminFeeVal
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L387

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.

```
EGGS_TO_HATCH_1MINERS = _rate
```

Recommendation

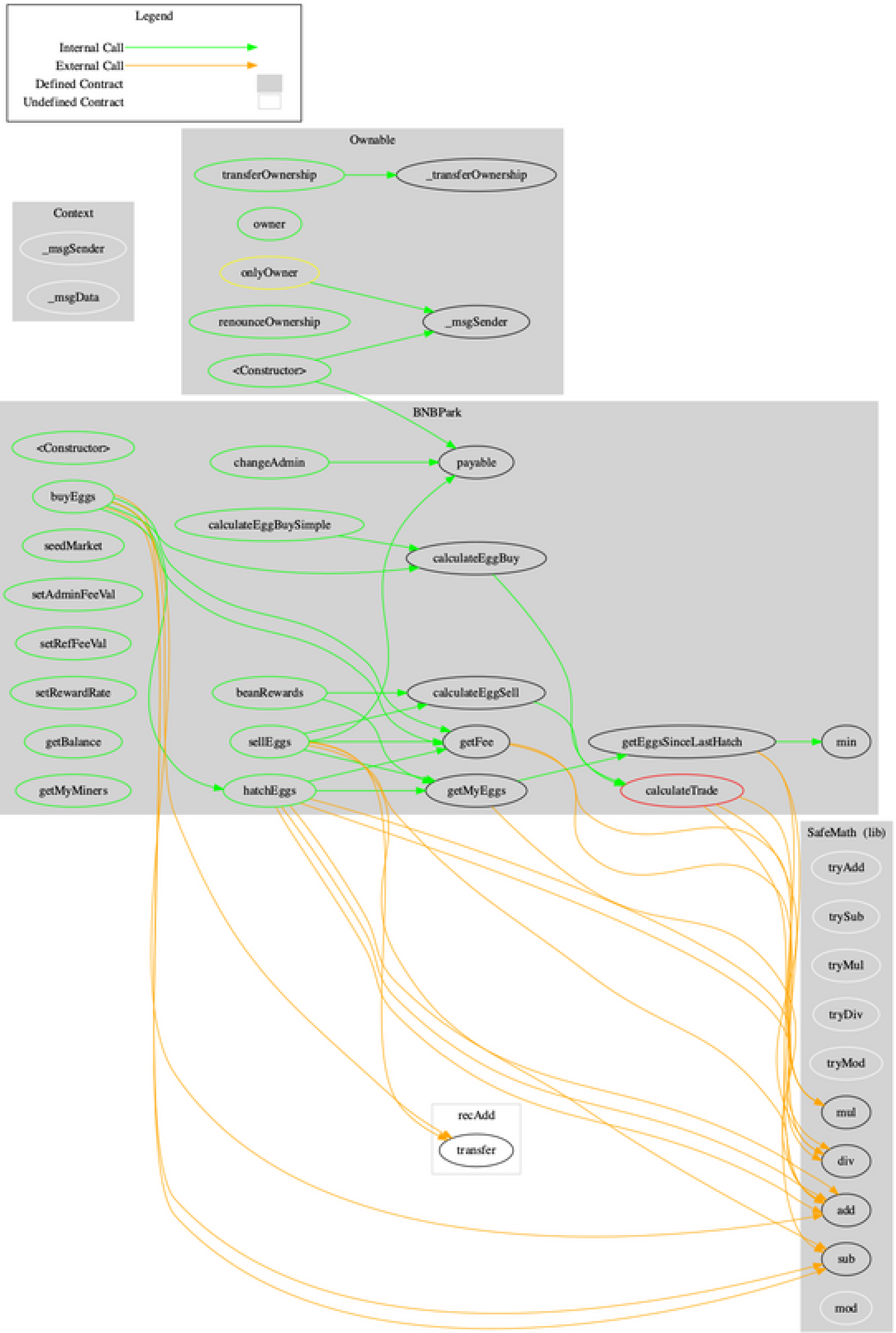
Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
BNBPark	Implementation	Context, Ownable		
	<Constructor>	Public	✓	-

	hatchEggs	Public	✓	-
	sellEggs	Public	✓	-
	beanRewards	Public		-
	buyEggs	Public	Payable	-
	calculateTrade	Private		
	calculateEggSell	Public		-
	calculateEggBuy	Public		-
	calculateEggBuySimple	Public		-
	getFee	Private		
	seedMarket	Public	Payable	onlyOwner
	setAdminFeeVal	Public	✓	onlyOwner
	setRefFeeVal	Public	✓	onlyOwner
	setRewardRate	Public	✓	onlyOwner
	changeAdmin	Public	✓	onlyOwner
	getBalance	Public		-
	getMyMiners	Public		-
	getMyEggs	Public		-
	getEggsSinceLastHatch	Public		-
	min	Private		

Contract Flow



Summary

BNB Park is a novel project where users have the ability to buy eggs in order to redeem minters. The users can later claim the awarded amount that is based on the time period that has elapsed, the number of eggs/minters and the contract's balance. This audit focuses on the business logic, the security concerns and performance 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.

Cyberscope 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 Cyberscope 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 Cyberscope 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 Cyberscope team disclaims any liability for the resulting losses.

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