Cyberscope

Audit Report

WONDER MINER

May 2022

Network BSC, AVAX ,CRO, FTM, MOVR

Audited by © cyberscope

Table of Contents

Contract Review

BSC

Contract Name	WonderMiner
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/address/0x0afc7F8f05747a95208B06a137079447B55A8Dcb

AVAX

Contract Name	WonderMiner
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://snowtrace.io/address/0x0afc7F8f05747a95208B06a137079447B55A8Dcb

CRO

Contract Name	WonderMiner
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://cronoscan.com/address/0x080E7C8bD5E9A2850e0689355aE2A2aba8FBe07a

FTM

Contract Name	WonderMiner
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://ftmscan.com/address/0x43A270c4B1CC2AAB315dE5F125148064e7C7b68d

MOVR

Contract Name	WonderMiner
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://moonriver.moonscan.io/address/0xc629929Fcd415A11A215B6337523b9c1E2d88661

Source Files

Filename	SHA256
contract.sol	ee48a20ea75b03fb828f526fa0225d0d287302d0136c54256e71b57ac6eb3f47

Audit Updates

Initial Audit	16th May 2022
Corrected	

Contract Analysis

- The users have the ability to buy eggs by paying in the native currency.
- The buying process is called "hireFarmers".
- During the hireFarmers process the referred user takes a proportional amount as a reward in the native token.
- The price of eggs depends on some variations like the current egg supply and the WonderMiner contract's native currency balance.
- The buy and sell amount is taxed. The taxed amount is moved directly to the dev wallet.
- The users gathered eggs in order to redeem rewards.
- The redeem process is called "sellCrops".

Contract Owner Privileges

The contract owner has the authority to manipulate the minimum amount of eggs that can be bought, the minimum value that can be set is 10.

The contract owner has the authority to manipulate the taxed amount. The maximum value that can be set is 1.5%.

The contract owner has the authority to manipulate the referral percentage. The value that can be configured is between 1% and 10%.

The contract owner has the authority to manipulate the early withdrawal funds. The value that can be configured is up to 90%.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CBD	Contract Balance Dependency
	IAD	Initial Amount Distribution
	BA	Blacklist Addresses
	ICBM	Insufficient Contract Balance Manipulation
	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
	L13	Divide before Multiply Operation

Contract Balance Dependency

Criticality	minor
Location	contract.sol#L179

Description

The calculation of the sell and buy price heavily depends on the WonderMiner contract's amount. 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 eggs supply:

Amount	Contract Balance	Supply	Result
1	1000000	144000000000	718204846.9
10	1000000	144000000000	7024116132
100	1000000	144000000000	57375089648.5

The following is the same amounts with different contract balance:

Amount	Contract Balance	Supply	Result
1	1000	144000000000	205743677668.2
10	1000	144000000000	276497695852.5
100	1000	144000000000	281359906213.3

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#L184

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.

```
uint256 eggsBought = calculateEggBuy(msg.value,
address(this).balance.sub(msg.value));
```

Recommendation

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

Blacklist Addresses

Criticality	medium
Location	contract.sol#L143

Description

The contract owner has the authority to massively stop contracts from claiming their rewards. The owner may take advantage of it by calling the `blackMultipleWallets`.

```
if (blacklistActive) {
    require(!Blacklisted[msg.sender], "Address is blacklisted.");
}
```

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.

Insufficient Contract Balance Manipulation

Criticality	minor
Location	contract.sol#L167

Description

If the contract balance is not sufficient to cover the user's rewards, then the user will receive the contract balance instead of the entire reward.

```
if(getBalance() < eggValue) {
    eggValue = getBalance();
}
```

Recommendation

The contract could keep in track that the user has not received the corresponding amount.

MC - Missing Check

Criticality	minor
Location	contract.sol#L356

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 MARKET_EGGS_DIVISOR could cause a zero division error if the contract owner set the zero value.

```
function PRC_MARKET_EGGS_DIVISOR(uint256 value) external {
    require(msg.sender == owner, "Admin use only.");
    require(value <= 50);
    MARKET_EGGS_DIVISOR = value;
}
```

```
marketEggs = marketEggs.add(eggsUsed.div(MARKET_EGGS_DIVISOR));
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L74,79,84,91,96,140,232,253,257,281,286,299,303

Description

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

```
getMyMiners
getSiteInfo
getEggsYield
calculateEggBuySimple
getAvailableEarnings
getTimeStamp
getUserInfo
sellCrops
startFarm
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L11,32,33

Description

Constant state variables should be declared constant to save gas.

```
PSNH
PSN
PERCENTS_DIVIDER
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L79,84,91,222,232,257,323,328,338,344,350,356,362,368,374,380,386,391,396,402,408,9,10,11,12,13,15,16,18,19,20,22,23,32,33,36,38,39

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.

```
WITHDRAW_COOLDOWN
CUTOFF_STEP
Blacklisted
PSNH
PSN
COMPOUND_FOR_NO_TAX_WITHDRAWAL
WITHDRAWAL_TAX
COMPOUND_STEP
COMPOUND_BONUS_MAX_TIMES
...
```

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#L179,338,344,368,391

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.

```
CUTOFF_STEP = value * 60 * 60
COMPOUND_BONUS = value
TAX = value
EGGS_TO_HIRE_1MINERS = value
totalRefBonus = totalRefBonus.add(refRewards)
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L286

Description

Performing divisions before multiplications may cause lose of prediction.

```
miners = eggsAmount.div(EGGS_TO_HIRE_1MINERS)
```

Recommendation

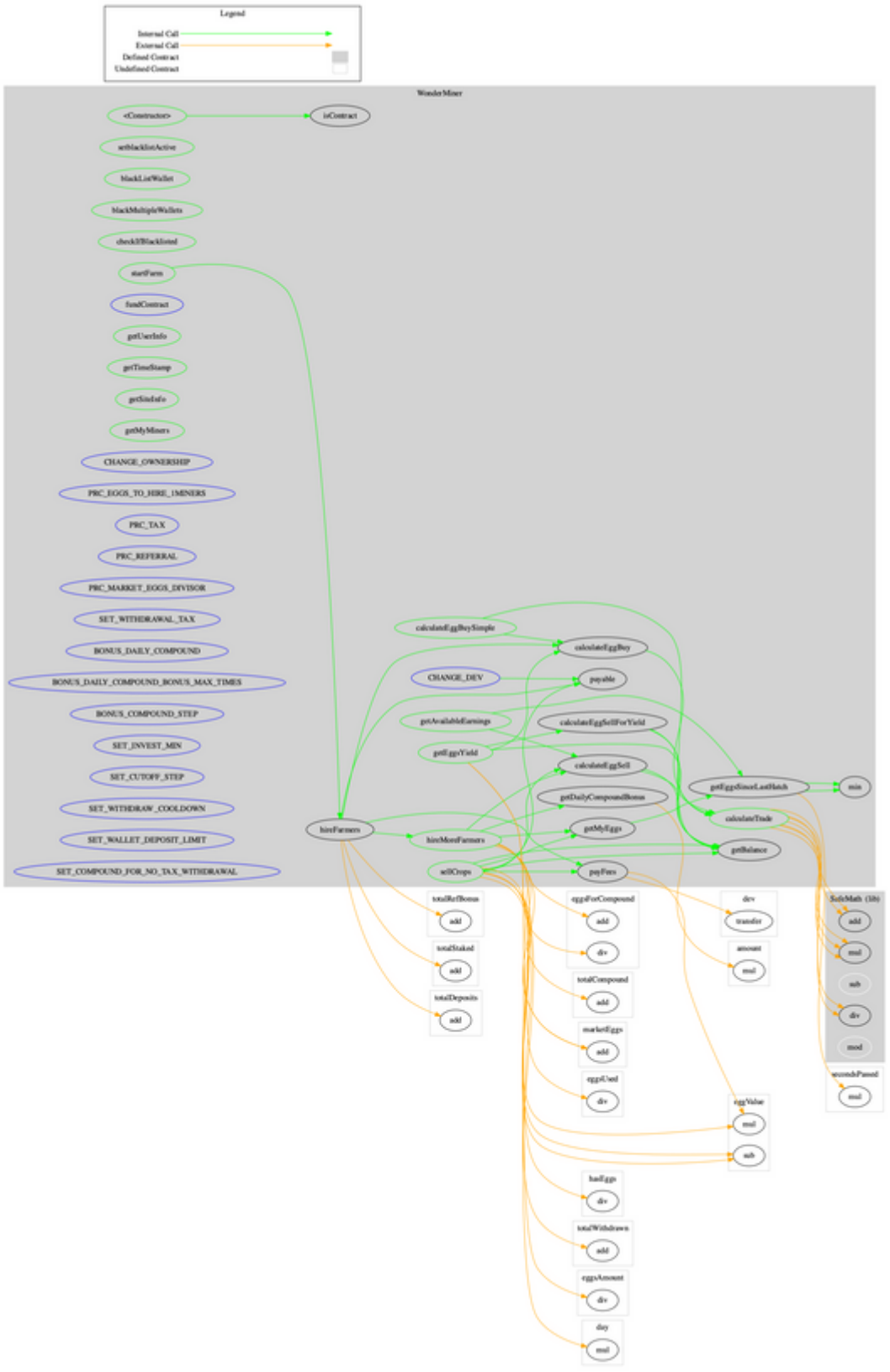
The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
WonderMiner	Implementation			
	<Constructor>	Public	✓	-
	isContract	Internal		
	setblacklistActive	Public	✓	-
	blackListWallet	Public	✓	-
	blackMultipleWallets	Public	✓	-
	checkIfBlacklisted	Public		-
	startFarm	Public	Payable	-
	fundContract	External	Payable	-
	hireMoreFarmers	Public	✓	-
	sellCrops	Public	✓	-
	hireFarmers	Public	Payable	-
	payFees	Internal	✓	
	getDailyCompoundBonus	Public		-
	getUserInfo	Public		-
	getBalance	Public		-
	getTimeStamp	Public		-
	getAvailableEarnings	Public		-
	calculateTrade	Public		-
	calculateEggSell	Public		-
	calculateEggBuy	Public		-
	calculateEggBuySimple	Public		-
	getEggsYield	Public		-
	calculateEggSellForYield	Public		-
	getSiteInfo	Public		-
	getMyMiners	Public		-
	getMyEggs	Public		-
	getEggsSinceLastHatch	Public		-
	min	Private		

	CHANGE_OWNERSHIP	External	✓	-
	CHANGE_DEV	External	✓	-
	PRC_EGGS_TO_HIRE_1MINERS	External	✓	-
	PRC_TAX	External	✓	-
	PRC_REFERRAL	External	✓	-
	PRC_MARKET_EGGS_DIVISOR	External	✓	-
	SET_WITHDRAWAL_TAX	External	✓	-
	BONUS_DAILY_COMPOUND	External	✓	-
	BONUS_DAILY_COMPOUND_BONUS_MAX_TIMES	External	✓	-
	BONUS_COMPOUND_STEP	External	✓	-
	SET_INVEST_MIN	External	✓	-
	SET_CUTOFF_STEP	External	✓	-
	SET_WITHDRAW_COOLDOWN	External	✓	-
	SET_WALLET_DEPOSIT_LIMIT	External	✓	-
	SET_COMPOUND_FOR_NO_TAX_WITHDRAWAL	External	✓	-
SafeMath	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	mod	Internal		

Contract Flow



Summary

WONDER MINER is a novel project where users have the ability to buy eggs in order to redeem rewards. The users can later claim the awarded amount that is based on the time period that has elapsed, the number of eggs 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