Cyberscope

Audit Report

Stoned Metaverse

March 2022

SHA256 6d6636eeac2dfe96a092b0cd4403289c3a6921b958a49c5412393a8fc5834968

Audited by © cyberscope

Table of Contents

Contract Review

SHA256	6d6636eeac2dfe96a092b0cd4403289c3a6921b958a49c5412393a8fc5834968
Source	SMT.sol
Domain	nutgain.io

Audit Updates

Initial Audit	18th March 2022
Corrected	

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L310

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the dexTaxFee to a high value. This will cause the contract to operate as a honeypot

```
if(recipient==pairAddress&&takeFee){
    uint256 taxFee = amount.mul(dexTaxFee).div(10000);
    _balances[taxAddress] = _balances[taxAddress].add(taxFee);
    emit Transfer(sender, taxAddress, taxFee);
    amount = amount.sub(taxFee);
}
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L102

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setTax` function with a high percentage value.

```
function setTax(uint256 _taxFee) public onlyOwner{
    dexTaxFee = _taxFee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	FSA	Fixed Swap Address
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L75

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
//IPancakeRouter _router =
IPancakeRouter(0xECC5428A66808FC40A464e5B3F4D265Df985E3E8); //for test
IPancakeRouter _router =
IPancakeRouter(0x10ED43C718714eb63d5aA57B78B54704E256024E);


pairAddress = IPancakeFactory(_router.factory())
    .createPair(address(this), _router.WETH());

// set the rest of the contract variables
routerAddress = address(_router);
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L65,90,94,98,102,106,110,67,75,92 and 9 more

Description

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

```
transferOwnership
renounceOwnership
decreaseAllowance
increaseAllowance
transferFrom
approve
transfer
balanceOf
totalSupply
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L23,98,102,29,33,87,55,59,394,18 and 2 more

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.

```
__gap
__Context_init_unchained
__Context_init
__ERC20_init_unchained
__ERC20_init
__Ownable_init_unchained
__Ownable_init
_taxFee
_taxAddress
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L87

Description

There are segments that contain unused state variables.

```
__gap
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L102

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.

```
dexTaxFee = _taxFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L85,95,114,128,147,157,60,174,18,21 and 15 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
safeTransferFrom
safeTransfer
safeIncreaseAllowance
safeDecreaseAllowance
safeApprove
_callOptionalReturn
_transfer
_spendAllowance
_mint
...
```

Recommendation

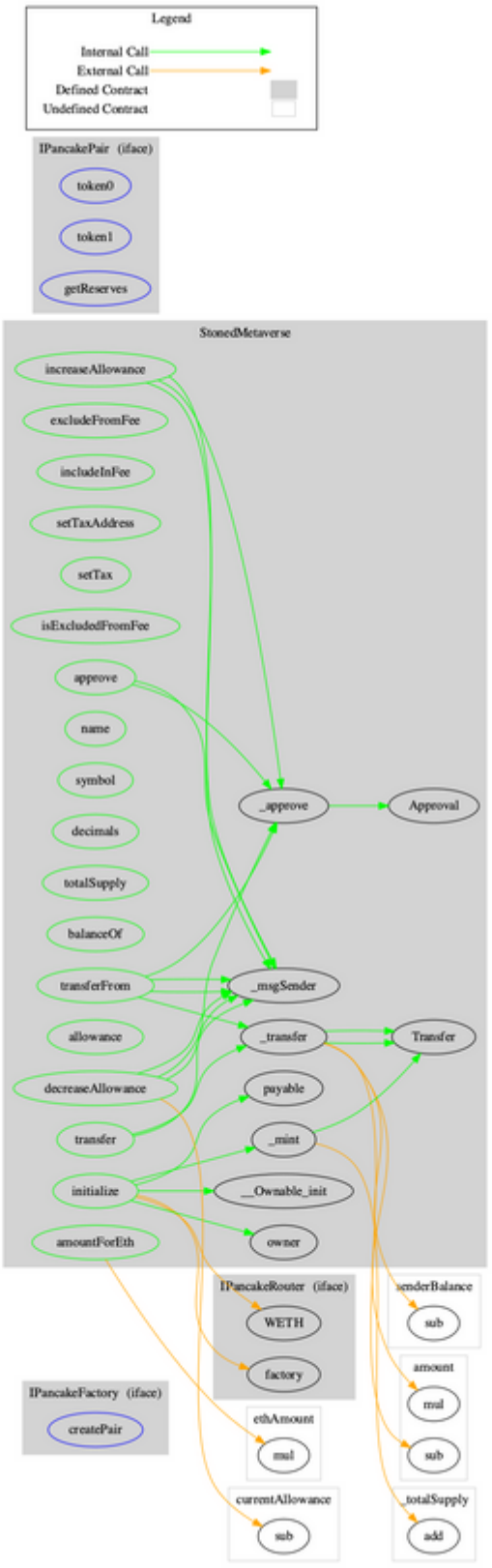
Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IPancakeFactory	Interface			
	createPair	External	✓	-
IPancakeRouter	Interface			
	WETH	External		-
	factory	External		-
IPancakePair	Interface			
	token0	External		-
	token1	External		-
	getReserves	External		-
StonedMetaverse	Implementation	Initializable, ERC20Upgradeable, OwnableUpgradeable		
	initialize	Public	✓	initializer
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setTaxAddress	Public	✓	onlyOwner
	setTax	Public	✓	onlyOwner
	isExcludedFromFee	Public		-
	amountForEth	Public		-
	name	Public		-
	symbol	Public		-
	decimals	Public		-

	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_approve	Internal	✓	

Contract Flow



Domain Info

Domain Name	stonedverse.network
Registry Domain ID	350be2c747aa40ea84f6f21ac7e046ff-DONUTS
Creation Date	2022-02-21T16:47:14Z
Updated Date	2022-02-26T16:48:04Z
Registry Expiry Date	2023-02-21T16:47:14Z
Registrar WHOIS Server	whois.donuts.co
Registrar URL	http://domains.google.com
Registrar	Google Inc.
Registrar IANA ID	895

The domain has been created 24 days before the creation of the audit. It will expire in 11 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

The Smart Contract analysis reported a critical issue. The contract Owner has the ability to increase the sale fees without limit. If the fees increase to a high value, then the contract can be converted into a honeypot and prevent users from selling. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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