Audit Report

Wenabis

February 2022

Type	BEP20
Network	BSC
Address	0xAfe91443bd6A8ceFD50d3095D8E24a4F0D205aA0
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	Token6
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	
Explorer	https://bscscan.com/token/0xde90431f56596C24002A4E6fEFA3052de7D619dA
Symbol	
Decimals	18
Total Supply	-

Source	contracts/token6.sol, @openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol, @openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20BurnableUpgradeable.sol, @openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol, @openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol, @openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol, @openzeppelin/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol, @openzeppelin/contracts-upgradeable/utils/math/SafeMathUpgradeable.sol, @openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol, @openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol, @openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol, @openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol, @openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol, @openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol, @openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol, @openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol, @openzeppelin/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol, @openzeppelin/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol, @openzeppelin/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol, @openzeppelin/contracts-upgradeable/utils/StorageSlotUpgradeable.sol
Domain	

Audit Updates

Initial Audit	26th February 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	medium
Location	contract.sol#L158,162

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `_maxTxAmount` or `_maxWalletAmount` to zero.

```
require(amount < _maxTxAmount, "AntiWhale 1% transfer 5% WalletHold.");
```

```
require((balanceOf(recipient) + amount) < _maxWalletAmount, "AntiWhale: 1%
transfer 5% WalletHold.");
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount and _maxWalletAmount 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#L351

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 `updateFee` function with a high percentage value.

```
    function updateFee(uint256 _txFee,uint256 _burnFee,uint256 _charityFee)
public  onlyRole(UPGRADER_ROLE) {
        require(_txFee < 100 && _burnFee < 100 && _charityFee < 100);
        _taxFees = _txFee* 100;
        _burnFees = _burnFee * 100;
        _charityFees = _charityFee* 100;
        origTaxFees =_taxFees;
        origburnFees =_burnFees;
        origCharityFees = _charityFees;

    }

...

uint256 tFee = ((tAmount.mul(taxFee)).div(granularity)).div(100);
uint256 tBurn = ((tAmount.mul(burnFee)).div(granularity)).div(100);
uint256 tCharity = ((tAmount.mul(charityFee)).div(granularity)).div(100);
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L154

Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by calling the `enableBlacklist` function.

```
require(!isBlacklisted(msg.sender), "sender en lista negra");
require(!isBlacklisted(recipient), "receptor en  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.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L07	Missing Events Arithmetic
●	L15	Local Scope Variable Shadowing
●	L14	Uninitialized Variables in Local Scope
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#L136,149,167 @openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#L67,75,92,141,163,186,206 @openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20BurnableUpgradeable.sol#L26,41 contracts/token6.sol#L131,311

Description

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

```
updateFee
enableBlacklist
burnFrom
burn
decreaseAllowance
increaseAllowance
transferFrom
approve
decimals
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contracts/token6.sol#L16

Description

Constant state variables should be declared constant to save gas.

```
_owner
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	@openzeppelin/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#L107 contracts/token6.sol#L20

Description

There are segments that contain unused state variables.

```
_allowances
__gap
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#L51,54,235 @openzeppelin/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#L21,24,211 and 7 more files

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.

```
_charityFees
_burnFees
_taxFees
_charityFee
_burnFee
_txFee
_amount
_feeaddress
_supply
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#L54,200,191 @openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#L85,95,114,128,147,157,60 @openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#L18,21,27 @openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#L24,27 and 7 more files

Description

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

```
_authorizeUpgrade
__UUPSUpgradeable_init_unchained
toString
toHexString
getUint256Slot
getBytes32Slot
trySub
tryMul
tryMod
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	@openzeppelin/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#L98

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
slot
```

Recommendation

The variables should be declared before any usage of them.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contracts/token6.sol#L95

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.

```
_tTotal -= _amount
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contracts/token6.sol#L45

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
decimals
```

Recommendation

The local variables should have different names from the upper scoped variables.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	@openzeppelin/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#L98

Description

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

```
slot
```

Recommendation

All the local scoped variables should be initialized.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contracts/token6.sol#L45

Description

Performing divisions before multiplications may cause lose of prediction.

```
_minSupply = _tTotal.div(2) * _decimalsFactor
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AccessControlUpgradeable	Implementation	Initializable, ContextUpgradeable, IAccessControlUpgradeable, ERC165Upgradeable		
	__AccessControl_init	Internal	✓	onlyInitializing
	__AccessControl_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControlUpgradeable	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-

IERC1822ProxiableUpgradeable	Interface			
	proxiableUUID	External		-
IBeaconUpgradeable	Interface			
	implementation	External		-
ERC1967UpgradeUpgradeable	Implementation	Initializable		
	__ERC1967Upgrade_init	Internal	✓	onlyInitializing
	__ERC1967Upgrade_init_unchained	Internal	✓	onlyInitializing
	_getImplementation	Internal		
	_setImplementation	Private	✓	
	_upgradeTo	Internal	✓	
	_upgradeToAndCall	Internal	✓	
	_upgradeToAndCallUUPS	Internal	✓	
	_getAdmin	Internal		
	_setAdmin	Private	✓	
	_changeAdmin	Internal	✓	
	_getBeacon	Internal		
	_setBeacon	Private	✓	
	_upgradeBeaconToAndCall	Internal	✓	
	_functionDelegateCall	Private	✓	
Initializable	Implementation			
	_isConstructor	Private		
UUPSUpgradeable	Implementation	Initializable, IERC1822ProxiableUpgradeable, ERC1967UpgradeUpgradeable		
	__UUPSUpgradeable_init	Internal	✓	onlyInitializing
	__UUPSUpgradeable_init_unchained	Internal	✓	onlyInitializing

	proxiableUUID	External		notDelegated
	upgradeTo	External	✓	onlyProxy
	upgradeToAndCall	External	Payable	onlyProxy
	_authorizeUpgrade	Internal	✓	
PausableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	__Pausable_init	Internal	✓	onlyInitializing
	__Pausable_init_unchained	Internal	✓	onlyInitializing
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		
	__ERC20_init	Internal	✓	onlyInitializing
	__ERC20_init_unchained	Internal	✓	onlyInitializing
	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	✓	
	_burn	Internal	✓	

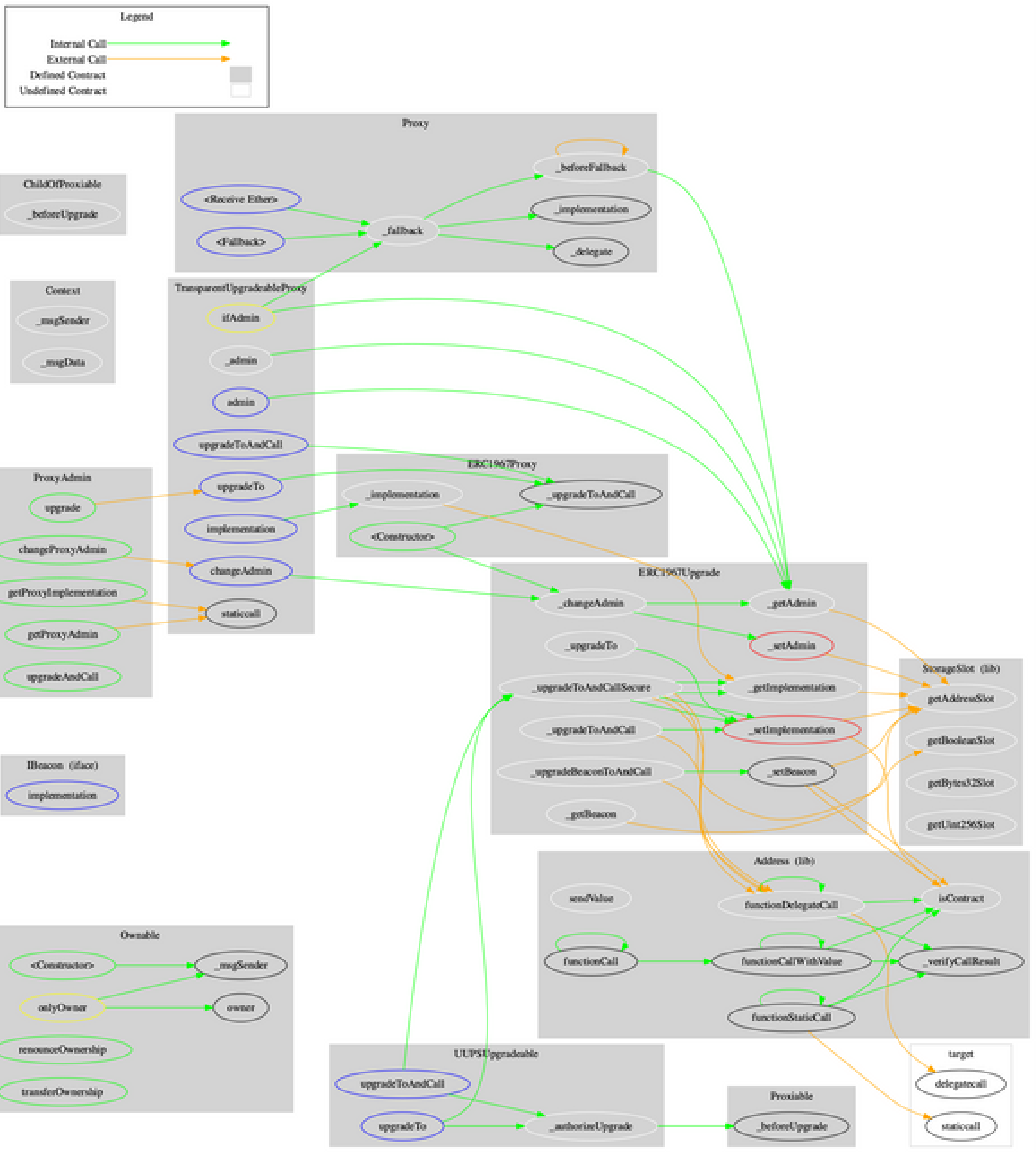
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20BurnableUpgradeable	Implementation	Initializable, ContextUpgradeable, ERC20Upgradeable		
	__ERC20Burnable_init	Internal	✓	onlyInitializing
	__ERC20Burnable_init_unchained	Internal	✓	onlyInitializing
	burn	Public	✓	-
	burnFrom	Public	✓	-
IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20Upgradeable	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
AddressUpgradeable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	

	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	onlyInitializing
	__Context_init_unchained	Internal	✓	onlyInitializing
	_msgSender	Internal		
	_msgData	Internal		
ERC165Upgradeable	Implementation	Initializable, IERC165Upgradeable		
	__ERC165_init	Internal	✓	onlyInitializing
	__ERC165_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
IERC165Upgradeable	Interface			
	supportsInterface	External		-
SafeMathUpgradeable	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		
StorageSlotUpgradeable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
StringsUpgradeable	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
Token6	Implementation	Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, AccessControlUpgradeable, UUPSUpgradeable		
	initialize	External	✓	initializer
	setMaxTxPercent	External	✓	onlyRole
	setMaxWalletPercent	External	✓	onlyRole
	totalSupply	Public		-
	balanceOf	Public		-
	tokenFromReflection	Public		-
	burn	Public	✓	-
	excludeAccount	External	✓	onlyRole
	includeAccount	External	✓	onlyRole
	enableBlacklist	Public	✓	onlyRole
	disableBlacklist	External	✓	onlyRole
	isBlacklisted	Public		-

	_transfer	Internal	✓	
	_transferStandard	Private	✓	
	_standardTransferContent	Private	✓	
	_transferToExcluded	Private	✓	
	_excludedFromTransferContent	Private	✓	
	_transferFromExcluded	Private	✓	
	_excludedToTransferContent	Private	✓	
	_transferBothExcluded	Private	✓	
	_bothTransferContent	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTBasics	Private		
	getTTransferAmount	Private		
	_getRBasics	Private		
	_getRTransferAmount	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_sendToCharity	Private	✓	
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	updateFee	Public	✓	onlyRole
	pause	External	✓	onlyRole
	unpause	External	✓	onlyRole
	_beforeTokenTransfer	Internal	✓	whenNotPaused
	_authorizeUpgrade	Internal	✓	onlyRole

Contract Flow



Domain Info

Domain Name	wenabis.co
Registry Domain ID	D2BEC39862EE14CFDAAD19477EE0D6C32-NSR
Creation Date	2021-05-06T03:33:40Z
Updated Date	2021-07-17T01:32:25Z
Registry Expiry Date	2022-05-06T03:33:40Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	whois.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 9 months before the creation of the audit. It will expire in 3 months.

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

Summary

Wenabis is aiming to build massive cannabis production for medical and industry use. There are some functions that can be abused by the owner, like manipulating fees, blacklisting contracts and stopping transactions. 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.

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