Cyberscope

Audit Report

UTA INU

April 2022

Type	BEP20
Network	BSC
Address	0x64600D9C8Df1ab026E0a90BCf04Dc22ca9B7DA36
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	UTA_BEP20Token
Compiler Version	v0.5.16+commit.9c3226ce
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x64600D9C8Df1ab026E0a90BCf04Dc22ca9B7DA36
Symbol	UTA
Decimals	18
Total Supply	100,000,000,000
Domain	utainu.club

Source Files

Filename	**SHA256**
contract.sol	8e192e3a28a716f4a6d04b50ab30745ce11fbcf4c16fbbf0f22216a0c817357c

Audit Updates

Initial Audit	22nd April 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#L513,517,522

Description

The contract owner has the authority to stop the sales for all users. The owner may take advantage of it by setting the `taxAddress` to the router address and the `tax` to a high number so the calculated amount will be insufficient.

```
if(taxStatus==true && sender==taxAddress && tax>0){
    _balances[sender] = _balances[sender].sub(amount, "BEP20: transfer amount
exceeds balance");
    _balances[masterWallet] = _balances[masterWallet].add(amount/100*tax);
    _balances[recipient] = _balances[recipient].add(amount/100*(100-tax));
```

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

```
if(sender==taxAddress){ // user buy Token
  require(amount<=maxBuyAmount, "You can not buy token larger than maximum
amount");
}

if(recipient==taxAddress){ // user sell Token
  require(amount<=maxSellAmount, "You can not sell token larger than maximum
amount");
}
```

Recommendation

The contract could embody a check for not allowing setting the maxSellAmount, maxBuyAmount 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.

Read more about tax manipulation in the corresponding section.

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#L551

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 `update_tax` function with a high percentage value.

```
function update_tax(uint newTax ) public onlyOwner{
    tax = newTax;
}
```

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
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L13	Divide before Multiply Operation
	L15	Local Scope Variable Shadowing

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L316,325,469,488,532,537,542,547,551

Description

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

```
update_tax
update_tax_Status
update_master_wallet
update_tax_Address
update_Max_Token_Amount
decreaseAllowance
increaseAllowance
transferOwnership
renounceOwnership
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L339,532,537,542,547,551

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.

```
update_tax
update_tax_Status
update_master_wallet
_newAddress
update_tax_Address
_maxSell
_maxBuy
update_Max_Token_Amount
UTA_BEP20Token
```

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#L532,551

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.

```
tax = newTax
maxSellAmount = _maxSell
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L582,617,564

Description

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

```
_mint
_burnFrom
_burn
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L508

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
taxStatus == true && sender == taxAddress && tax > 0
```

Recommendation

Remove the equality to the boolean constant.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L508

Description

Performing divisions before multiplications may cause lose of prediction.

```
_balances[recipient] = _balances[recipient].add(amount / 100 * (100 - tax))
_balances[masterWallet] = _balances[masterWallet].add(amount / 100 * tax)
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L355

Description

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

```
totalSupply
symbol
name
```

Recommendation

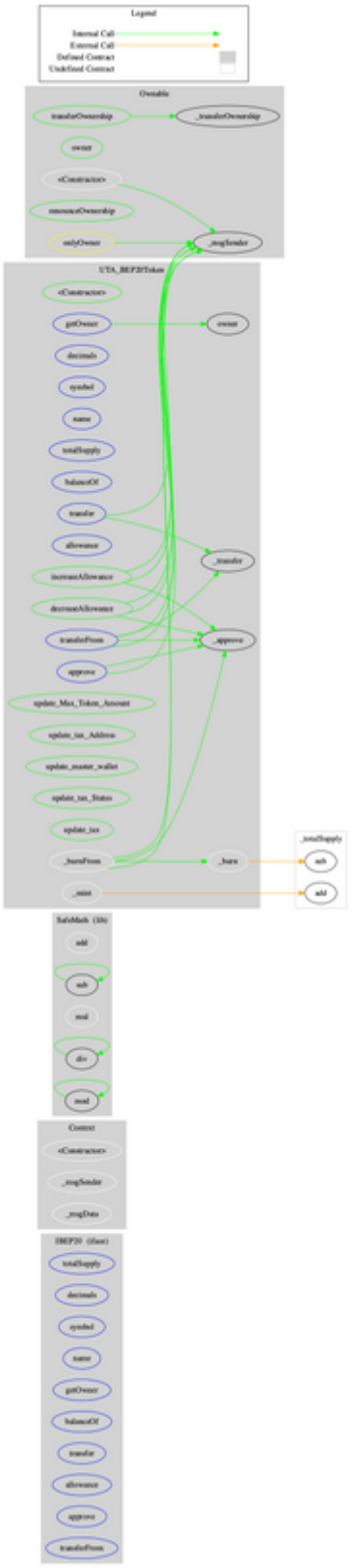
The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	<Constructor>	Internal	✓	
	_msgSender	Internal		
	_msgData	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	

	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
UTA_BEP20Token	Implementation	Context, IBEP20, Ownable		
	<Constructor>	Public	✓	-
	getOwner	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	update_Max_Token_Amount	Public	✓	onlyOwner
	update_tax_Address	Public	✓	onlyOwner
	update_master_wallet	Public	✓	onlyOwner
	update_tax_Status	Public	✓	onlyOwner
	update_tax	Public	✓	onlyOwner
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	

Contract Flow



Domain Info

Domain Name	utainu.club
Registry Domain ID	D7B546B9D295440AD9495CA0550C0F532-GDREG
Creation Date	2022-04-21T15:05:46Z
Updated Date	2022-04-21T15:05:57Z
Registry Expiry Date	2023-04-21T15:05:46Z
Registrar WHOIS Server	whois.registrar.eu
Registrar URL	www.openprovider.com
Registrar	Hosting Concepts B.V. d/b/a Registrar.eu
Registrar IANA ID	1647

The domain has been created 1 day before the creation of the audit. It will expire in 12 months.

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

Summary

UTA INU is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees and stopping transactions. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. 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