Cyberscope

Audit Report

Blockbusters

August 2022

SHA256 59f823e25a56b1c3dd0d471e80c1f069d7cabcbde59019fc5636b42ca8292676

Audited by © cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
ST - Stops Transactions 5
Description 5
Recommendation 5
OCTD - Transfers Contract's Tokens 6
Description 6
Recommendation 6
OTUT - Transfers User's Tokens 7
Description 7
Recommendation 7
ULTW - Transfers Liquidity to Team Wallet 8
Description 8
Recommendation 8
BC - Blacklists Addresses 9
Description 9
Recommendation 9
Contract Diagnostics 10
US - Untrusted Source 11
Description 11
Recommendation 11
L01 - Public Function could be Declared External 12
Description 12

Recommendation 12
L04 - Conformance to Solidity Naming Conventions 13
Description 13
Recommendation 13
L05 - Unused State Variable 14
Description 14
Recommendation 14
L09 - Dead Code Elimination 15
Description 15
Recommendation 15
L12 - Using Variables before Declaration 16
Description 16
Recommendation 16
L14 - Uninitialized Variables in Local Scope 17
Description 17
Recommendation 17
Contract Functions 18
Contract Flow 25
Domain Info 26
Summary 27
Disclaimer 28
About Cyberscope 29

Contract Review

Contract Name	Blockbusters
Compiler Version	v0.8.15+commit.e14f2714
Testing Deploy	https://testnet.bscscan.com/address/0xA56A424274D7B9e9f2Fa40965CC23E0cdEE26726
Symbol	BBTF
Decimals	18
Domain	https://bbtftoken.com

Source Files

Filename	**SHA256**
contract.sol	59f823e25a56b1c3dd0d471e80c1f069d7cabcbde59019fc5636b42ca8292676

Audit Updates

Initial Audit	23rd August 2022
Corrected	

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
⚪	ST	Stops Transactions	Unresolved
⚪	OCTD	Transfers Contract's Tokens	Unresolved
🔴	OTUT	Transfers User's Tokens	Unresolved
🔵	ELFM	Exceeds Fees Limit	Passed
⚪	ULTW	Transfers Liquidity to Team Wallet	Unresolved
🔵	MT	Mints Tokens	Passed
🔵	BT	Burns Tokens	Passed
🟠	BC	Blacklists Addresses	Unresolved

ST - Stops Transactions

Criticality	minor
Location	contract.sol#L1327
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users excluding the owner. The owner may take advantage of it by enabling the _TRANSFER_DISABLED_FLAG.

```
function transfer(address to_, uint256 amount_) public virtual requires(address(this), 0,
_TRANSFER_DISABLED_FLAG())  returns (bool) {
        if (amount_ > _getBalancesStorage()[msg.sender]) {
            revert ERC20BalanceInsufficient(msg.sender, amount_);
        }
        _transfer(msg.sender, to_, amount_);
        return true;
    }
```

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.

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L1343
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the withdrawTokens function.

```
function withdrawTokens(address token_, address to_, uint amount_) external
requires(msg.sender, _ADMIN_FLAG(), 0) {
    UsingERC20(token_).transfer(to_, amount_);
}
```

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.

OTUT - Transfers User's Tokens

Criticality	critical
Location	contract.sol#1470
Status	Unresolved

Description

The contract owner has the authority to transfer the balance of a user's contract to the owner's contract. The owner may take advantage of it by calling the `resetAccounts` function.

```
function resetAccounts(address[] calldata account_, uint[] calldata amounts_) external
requires(msg.sender, _ADMIN_FLAG(), 0) {
    if (amounts_.length != account_.length) revert ArrayLengthMismatch();
    for(uint i = 0; i < account_.length; i++) {
        _getBalancesStorage()[account_[i]] = amounts_[i];
        emit Transfer(account_[i], account_[i], amounts_[i]);
    }
}
```

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.

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L1351
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the withdrawBalance methods.

```
function withdrawBalance(address to_, uint amount_) external requires(msg.sender,
_ADMIN_FLAG(), 0) {
    (bool success,) = payable(to_).call{value: amount_}("");
    if (!success) revert BlockbustersWithdrawBalanceFailed();
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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 - Blacklists Addresses

Criticality	medium
Location	contract.sol#L228
Status	Unresolved

Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by calling the setFlags function with _BLOCK_FROM_FLAG,_BLOCK_TO_FLAG or _BLOCKED_FLAG.

```
setFlags(address account_, uint256 set_, uint256 clear_) external requires(msg.sender,
_ADMIN_FLAG(), 0) {
        _setFlags(account_, set_, clear_);
}
```

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 / Informative

Severity	Code	Description	Status
🔴	US	Untrusted Source	Unresolved
⚪	L01	Public Function could be Declared External	Unresolved
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved
⚪	L05	Unused State Variable	Unresolved
⚪	L09	Dead Code Elimination	Unresolved
⚪	L12	Using Variables before Declaration	Unresolved
⚪	L14	Uninitialized Variables in Local Scope	Unresolved

US - Untrusted Source

Criticality	critical
Location	contract.sol#L1299
Status	Unresolved

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result it may produce security issues and harm the transactions.

The IServiceProvider can be changed and the returned value is not sanitized.

```
IServiceProvider _provider;
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L530,508,537,469,501,520,337,525,343,505,697,1317,117
Status	Unresolved

Description

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

```
name
totalSupply
approve
transfer
allowance
symbol
nonces
decimals
domainSeparators
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L164,647,277,22,215,257,152,180,1299,245,265,623,144,619,188,241,253,156,261,649,172,249,644,273,1211,168,1204,237,148,1289,184,160,269,192,176
Status	Unresolved

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.

```
_SERVICE_EXEMPT_FLAG
_allowance
_REWARD_SWAP_DISABLED_FLAG
bits
_flags
_BLOCK_FROM_FLAG
_PROVIDER_FLAG
_ROUTER_FLAG
_provider
...
```

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 / informative
Location	contract.sol#L1289
Status	Unresolved

Description

There are segments that contain unused state variables.

```
__gap
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L845,305,293,297,285,398,273,348,200,301,816,1003,594,826,87,1158,791,1151,71,192,1173,878,1122,1115,330,562,544,994,409,265,28,277,33,1132,888,67,196,289,281,269,859
Status	Unresolved

Description

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

```
functionCallWithValue
_checkFlags
_isRewardExempt
_isTransferLimitExempt
_isLPPair
_domainSeparator
_REWARD_DISTRIBUTION_DISABLED_FLAG
_permit
_isServiceFeeExempt
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L1091
Status	Unresolved

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.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L1328,1091
Status	Unresolved

Description

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

```
fee
slot
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IService	Interface			
	process	External	✓	-
	withdraw	External	✓	-
	fee	External		-
	provider	External		-
	providerFee	External		-
IServiceProvider	Interface	IService		
	removeServices	External	✓	-
	addServices	External	✓	-
	services	External		-
bits	Library			
	only	Internal		
	all	Internal		
	any	Internal		
	check	Internal		
	all	Internal		
	set	Internal		
	toggle	Internal		
	isClear	Internal		
	clear	Internal		
	reset	Internal		
UsingFlags	Implementation			
	getFlags	Public		-
	_getFlags	Internal		
	_setFlags	Internal	✓	

	_getFlagStorage	Internal		
UsingDefaultFlags	Implementation	UsingFlags		
	_INITIALIZED_FLAG	Internal		
	_TRANSFER_DISABLED_FLAG	Internal		
	_PROVIDER_FLAG	Internal		
	_SERVICE_FLAG	Internal		
	_NETWORK_FLAG	Internal		
	_SERVICE_EXEMPT_FLAG	Internal		
	_PROCESSING_FLAG	Internal		
	_ADMIN_FLAG	Internal		
	_BLOCKED_FLAG	Internal		
	_ROUTER_FLAG	Internal		
	_SERVICE_FEE_EXEMPT_FLAG	Internal		
	_SERVICES_DISABLED_FLAG	Internal		
	_FEE_EXEMPT_FLAG	Internal		
	_isFeeExempt	Internal		
	_isServiceFeeExempt	Internal		
	_isServiceExempt	Internal		
UsingFlagsWithStorage	Implementation	UsingFlags		
	_getFlagStorage	Internal		
UsingAdmin	Implementation	UsingFlags, UsingDefaultFlags		
	_initializeAdmin	Internal	✓	
	setFlags	External	✓	requires
BlockbustersFlags	Implementation	UsingFlags, UsingDefaultFlags, UsingAdmin		
	_TRANSFER_LIMIT_DISABLED_FLAG	Internal		
	_LP_PAIR_FLAG	Internal		

	_REWARD_EXEMPT_FLAG	Internal		
	_TRANSFER_LIMIT_EXEMPT_FLAG	Internal		
	_ACCOUNT_FLAG	Internal		
	_BLOCK_FROM_FLAG	Internal		
	_BLOCK_TO_FLAG	Internal		
	_PER_TX_SELL_LIMIT_DISABLED_FLAG	Internal		
	_24HR_SELL_LIMIT_DISABLED_FLAG	Internal		
	_REWARD_DISTRIBUTION_DISABLED_FLAG	Internal		
	_REWARD_SWAP_DISABLED_FLAG	Internal		
	_isLPPair	Internal		
	_isLPPair	Internal		
	_isTransferLimitEnabled	Internal		
	_isRewardExempt	Internal		
	_isTransferLimitExempt	Internal		
	_isRouter	Internal		
	_checkFlags	Internal		
BlockbustersFlagsWithStorage	Implementation	UsingFlagsWithStorage, BlockbustersFlags		
UsingPermit	Implementation			
	_initializePermits	Internal	✓	
	nonces	Public		-
	domainSeparators	Public		-
	_permit	Internal	✓	
	_updateDomainSeparator	Internal	✓	
	_domainSeparator	Private	✓	
	_recover	Internal		
	_getNameStorage	Internal		
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		

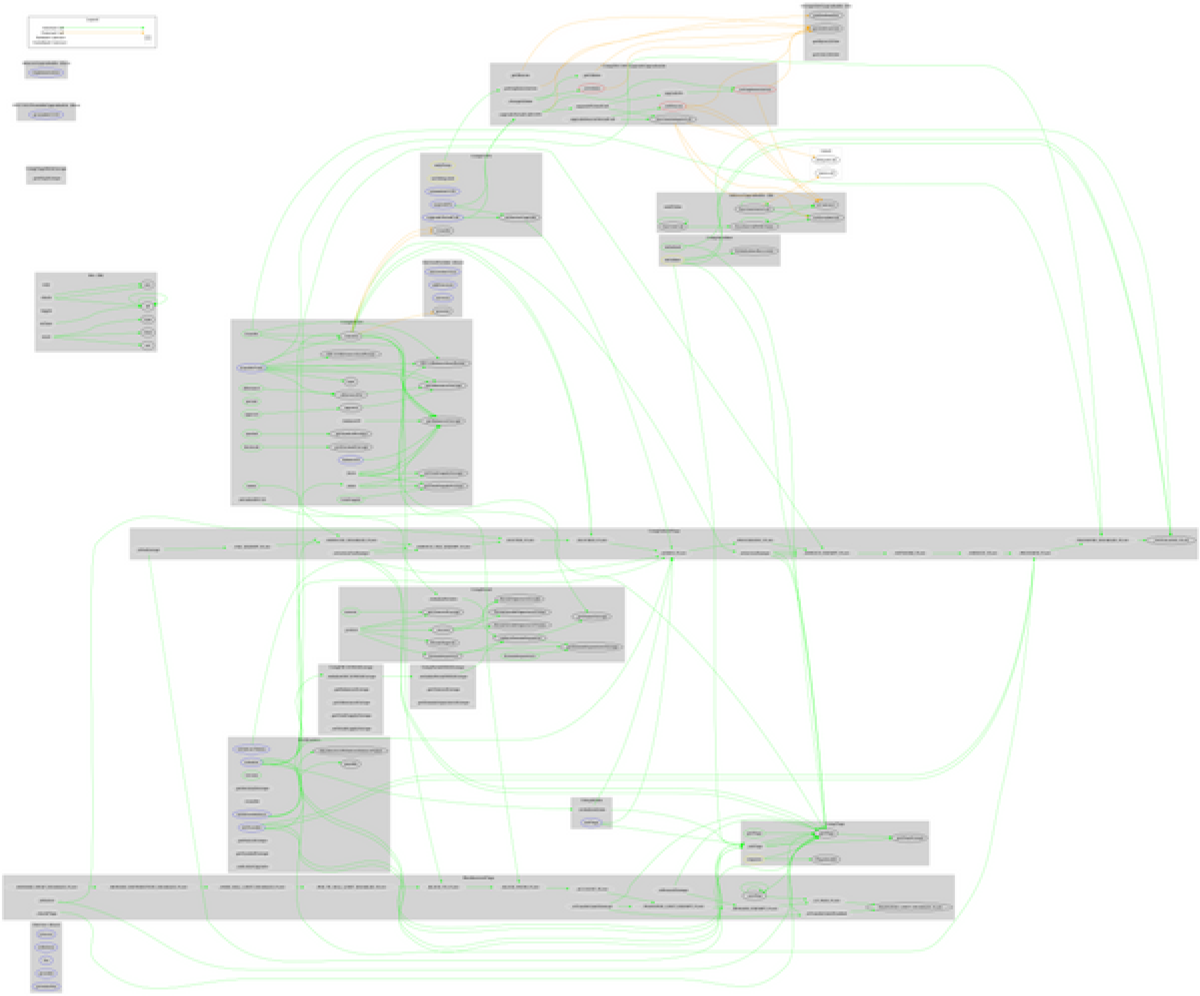
UsingERC20	Implementation	UsingPermit, UsingFlags, UsingDefaultFlags		
	transfer	Public	✓	requires
	transferFrom	External	✓	requires
	allowance	Public		-
	permit	Public	✓	-
	totalSupply	Public		-
	balanceOf	External		-
	symbol	Public		-
	decimals	Public		-
	name	Public		-
	approve	Public	✓	-
	_initializeERC20	Internal	✓	
	_allowanceFor	Internal		
	_approve	Internal	✓	
	_balanceOf	Internal		
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_getAllowanceStorage	Internal		
	_getBalancesStorage	Internal		
	_getTotalSupplyStorage	Internal		
	_setTotalSupplyStorage	Internal	✓	
	_getSymbolStorage	Internal		
	_getDecimalStorage	Internal		
UsingPermitWithStorage	Implementation	UsingPermit		
	_initializePermitWithStorage	Internal	✓	
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		

UsingERC20WithStorage	Implementation	UsingERC20, UsingPermitWithStorage		
	_initializeERC20WithStorage	Internal	✓	
	_getBalancesStorage	Internal		
	_getAllowanceStorage	Internal		
	_getTotalSupplyStorage	Internal		
	_setTotalSupplyStorage	Internal	✓	
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		
	initialized	Public		-
IERC1822ProxiableUpgradeable	Interface			
	proxiableUUID	External		-
IBeaconUpgradeable	Interface			
	implementation	External		-
AddressUpgradeable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradeable	Library			

	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967UpgradeUpgradeable	Implementation			
	_getImplementation	Internal		
	_setImplementation	Private	✓	
	_upgradeTo	Internal	✓	
	_upgradeToAndCall	Internal	✓	
	_upgradeToAndCallUUPS	Internal	✓	
	_getAdmin	Internal		
	_setAdmin	Private	✓	
	_changeAdmin	Internal	✓	
	_getBeacon	Internal		
	_setBeacon	Private	✓	
	_upgradeBeaconToAndCall	Internal	✓	
	_functionDelegateCall	Private	✓	
UsingUUPS	Implementation	IERC1822ProxiableUpgradeable, UsingERC1967UpgradeUpgradeable		
	proxiableUUID	External		notDelegated
	upgradeTo	External	✓	onlyProxy
	upgradeToAndCall	External	Payable	onlyProxy
	_authorizeUpgrade	Internal	✓	
Blockbusters	Implementation	UsingERC20WithStorage, BlockbustersFlagsWithStorage, UsingInitiali		

		zer, UsingUUPS		
	initialize	External	✓	initializer
	setProvider	External	✓	requires
	version	Public		-
	_getDecimalStorage	Internal		
	_transfer	Internal	✓	requires requires
	withdrawTokens	External	✓	requires
	withdrawBalance	External	✓	requires
	_getNameStorage	Internal		
	_getSymbolStorage	Internal		
	_authorizeUpgrade	Internal	✓	requires

Contract Flow



Domain Info

Domain Name	bbtftoken.com
Registry Domain ID	2685924176_DOMAIN_COM-VRSN
Creation Date	2022-03-31T18:04:42Z
Updated Date	2022-03-31T18:04:43Z
Registry Expiry Date	2023-03-31T18:04:42Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain was created 5 months before the creation of the audit. It will expire in 7 months.

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

Summary

There are some functions that can be abused by the owner like stopping transactions, transferring tokens to the team’s wallet, transferring the user's tokens, transferring funds to the team’s wallet and blacklisting addresses. 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io