



Cyberscope

Audit Report

Blockbusters

August 2022

Audited by © cyberscope

Table of Contents

Table of Contents	1
Introduction	2
Corresponding Audits	3
Blockbusters	3
BlockbustersInfrastructureService	3
BlockbustersLiquidityService	3
BlockbustersRewardsService	3
BlockbustersStakingService	3
BlockbustersSwap	3
Domain Info	4
Disclaimer	5
About Cyberscope	6

Introduction

Cyberscope audited six sections of the Blockbusters ecosystem.

Corresponding Audits

Blockbusters

<https://github.com/cyberscope-io/audits/tree/main/bbtf/blockbusters.pdf>

BlockbustersInfrastructureService

<https://github.com/cyberscope-io/audits/tree/main/bbtf/blockbustersInfrastructureService.pdf>

BlockbustersLiquidityService

<https://github.com/cyberscope-io/audits/tree/main/bbtf/blockbustersLiquidityService.pdf>

BlockbustersRewardsService

<https://github.com/cyberscope-io/audits/tree/main/bbtf/blockbustersRewardsService.pdf>

BlockbustersStakingService

<https://github.com/cyberscope-io/audits/tree/main/bbtf/blockbustersStakingService.pdf>

BlockbustersSwap

<https://github.com/cyberscope-io/audits/tree/main/bbtf/blockbustersSwap.pdf>

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.

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.

Cyberscope 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>



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
ULTW - Transfers Liquidity to Team Wallet	7
Description	7
Recommendation	7
BC - Blacklists Addresses	8
Description	8
Recommendation	8
Contract Diagnostics	9
US - Untrusted Source	10
Description	10
Recommendation	10
L01 - Public Function could be Declared External	11
Description	11
Recommendation	11
L04 - Conformance to Solidity Naming Conventions	12
Description	12

Recommendation	12
L05 - Unused State Variable	13
Description	13
Recommendation	13
L09 - Dead Code Elimination	14
Description	14
Recommendation	14
L12 - Using Variables before Declaration	15
Description	15
Recommendation	15
L14 - Uninitialized Variables in Local Scope	16
Description	16
Recommendation	16
Contract Functions	17
Contract Flow	24
Domain Info	25
Summary	26
Disclaimer	27
About Cyberscope	28

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#L469
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.

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`.

```
function 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

There 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	UsingFlags WithStorage , BlockbustersFlags		
UsingPermit	Implementation			
	_initializePermits	Internal	✓	
	nonces	Public		-
	domainSeparators	Public		-
	_permit	Internal	✓	
	_updateDomainSeparator	Internal	✓	
	_domainSeparator	Private	✓	
	_recover	Internal		
	_getNameStorage	Internal		
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		

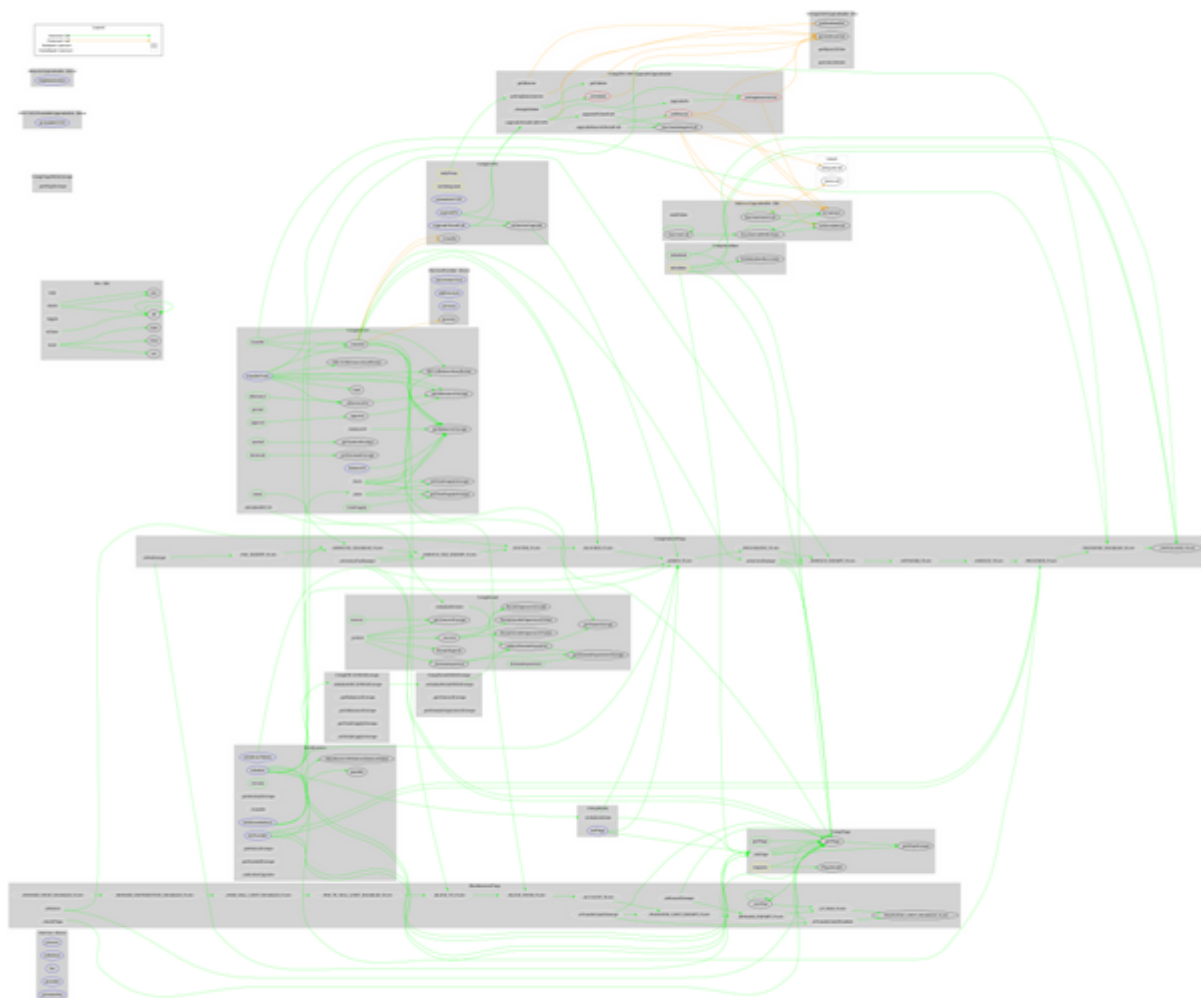
UsingERC20	Implementation	UsingPermit , UsingFlags, UsingDefaultFlags		
	transfer	Public	✓	requires
	transferFrom	External	✓	requires
	allowance	Public		-
	permit	Public	✓	-
	totalSupply	Public		-
	balanceOf	External		-
	symbol	Public		-
	decimals	Public		-
	name	Public		-
	approve	Public	✓	-
	_initializeERC20	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, UsingPermit WithStorage		
	_initializeERC20WithStorage	Internal	✓	
	_getBalancesStorage	Internal		
	_getAllowanceStorage	Internal		
	_getTotalSupplyStorage	Internal		
	_setTotalSupplyStorage	Internal	✓	
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		
	initialized	Public		-
IERC1822ProxiableUpgradable	Interface			
	proxiableUUID	External		-
IBeaconUpgradable	Interface			
	implementation	External		-
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradable	Library			

	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967 UpgradeUpgr adeable	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	IERC1822Pr oxiableUpgr adeable, UsingERC1 967Upgrade Upgradeabl e		
	proxiableUUID	External		notDelegated
	upgradeTo	External	✓	onlyProxy
	upgradeToAndCall	External	Payable	onlyProxy
	_authorizeUpgrade	Internal	✓	
Blockbusters	Implementation	UsingERC2 0WithStorag e, Blockbuster sFlagsWithS torage, 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>



Cyberscope

Audit Report

Blockbusters Infrastructure Service

August 2022

SHA256 5f5298c005a4ff7765732e36bfbe7b035f0335b0844dfcca5aff976ac6b45b7a

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Introduction	4
Contract Diagnostics	5
ST - Stops Transactions	5
Description	6
Recommendation	6
BLC - Business Logic Concern	7
Description	7
Recommendation	7
MC - Missing Check	8
Description	8
Recommendation	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L05 - Unused State Variable	12
Description	12

Recommendation	12
L09 - Dead Code Elimination	13
Description	13
Recommendation	13
L12 - Using Variables before Declaration	14
Description	14
Recommendation	14
L14 - Uninitialized Variables in Local Scope	15
Description	15
Recommendation	15
L15 - Local Scope Variable Shadowing	16
Description	16
Recommendation	16
Contract Functions	17
Contract Flow	23
Domain Info	24
Summary	25
Disclaimer	26
About Cyberscope	27

Contract Review

Contract Name	BlockbustersInfrastructureService
Compiler Version	v0.8.15+commit.e14f2714
Testing Deploy	https://testnet.bscscan.com/token/0xf198dE652d6Bc3d9956D2124dB523ecf13D83639
Domain	https://bbtftoken.com

Source Files

Filename	SHA256
contract.sol	5f5298c005a4ff7765732e36bfbe7b035f0335b0844dfcca5aff976ac6b45b7a

Audit Updates

Initial Audit	23rd August 2022
Corrected	

Introduction

The Blockbusters Infrastructure Service contract for processing the fees of a transaction. It is intended to be used by a token as an external service in order to manipulate the fees of the trade.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	BLC	Business Logic Concern	Unresolved
●	MC	Missing Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	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
●	L15	Local Scope Variable Shadowing	Unresolved

ST - Stops Transactions

Criticality	medium
Location	contract.sol#L295
Status	Unresolved

Description

The contract is going to revert and stop the `_process` transaction on every `_calculateFee`, if the amount is lower than the fixed fee.

```
function _process(address from_, address to_, uint256 amount_) internal virtual returns (uint256){  
    return _isServiceExempt(from_, to_) ? 0 : _calculateFee(from_, to_, amount_);  
}
```

Recommendation

A suggested implementation could apply the fees on each amount that is processed.

BLC - Business Logic Concern

Criticality	minor / informative
Location	contract.sol#L1103
Status	Unresolved

Description

The variables `_txSellLimitPerHolder` and `_24hrSellLimitPerHolder` are not initialized, they will always be zero. As a result the if statement `if (_isLPPair(to_))` is not meaningful.

```
function _process(address from_, address to_, uint amount_) internal override returns (uint){
    if (_isLPPair(to_)) {
        if (!_perTxSellLimitDisabled() && amount_ > _txSellLimitPerHolder) {
            revert TXLimitExceeded();
        }
        if (!_24hrSellLimitDisabled()) {
            SellTxData storage txData = _sellTxData[from_];
            uint128 now = _timestamp();
            if (txData.timestamp == 0 || (now - txData.timestamp) > MINUTES_PER_24HRS) {
                _sellTxData[from_] = SellTxData(uint128(amount_), now);
            } else if (txData.total + amount_ > _24hrSellLimitPerHolder) {
                revert TXLimitExceeded();
            } else {
                txData.total += uint128(amount_);
            }
        }
    }
    return super._process(from_, to_, amount_);
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected logic.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L256
Status	Unresolved

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 method `_withdraw` should check if there is sufficient balance before every deposit.

```
function withdraw(address to_) external virtual requires(msg.sender, _PROVIDER_FLAG() |  
_NETWORK_FLAG() | _ADMIN_FLAG(), 0) {  
    _withdraw(to_);  
}  
  
function _withdraw(address to_) internal virtual {  
    uint balance = address(this).balance;  
    if (balance > 0) {  
        address provider = _getProviderStorage();  
        uint providerFee = _getProviderFeeStorage();  
        if ( provider != address(0) && providerFee > 0) {  
            _deposit(provider, balance * _getProviderFeeStorage() / _PRECISION());  
        }  
        _deposit(to_, address(this).balance);  
    }  
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L381,110
Status	Unresolved

Description

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

```
initialized  
getFlags
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L1078,1077
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_24hrSellLimitPerHolder  
_txSellLimitPerHolder
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L165,430,15,312,410,153,181,398,418,145,1077,189,394,406,157,414,173,338,402,426,977,1078,136,169,970,1087,390,149,1055,185,161,422,321,193,177,1095
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
_REWARD_SWAP_DISABLED_FLAG
bits
_flags
_BLOCK_FROM_FLAG
_PROVIDER_FLAG
_ROUTER_FLAG
_REWARD_EXEMPT_FLAG
_PER_TX_SELL_LIMIT_DISABLED_FLAG
...
```

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#L1055
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#L181,611,446,299,450,426,201,454,582,769,592,80,924,557,917,64,355,323,193,939,644,888,881,177,185,760,189,21,430,26,898,654,60,197,232,442,434,625
Status	Unresolved

Description

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

```
_ROUTER_FLAG  
functionCallWithValue  
_isRewardExempt  
_receive  
_isTransferLimitExempt  
_REWARD_DISTRIBUTION_DISABLED_FLAG  
_isServiceFeeExempt  
_isRouter  
functionCall  
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L857
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#L857
Status	Unresolved

Description

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

```
slot
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L287,286
Status	Unresolved

Description

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

```
providerFee
provider
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IService	Interface			
	process	External	✓	-
	withdraw	External	✓	-
	fee	External		-
	provider	External		-
	providerFee	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		
UsingPrecision	Implementation			
	_PRECISION	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		
UsingAdmin	Implementation	UsingFlags, UsingDefaultFlags		
	_initializeAdmin	Internal	✓	
	setFlags	External	✓	requires
UsingFees	Implementation	UsingDefaultFlags, UsingPrecision		
	_setFee	Internal	✓	
	_getFee	Internal		
	_applyFee	Internal		
	_getFeesStorage	Internal		
UsingService	Implementation	IService, UsingAdmin, UsingFees		
	<Receive Ether>	External	Payable	-

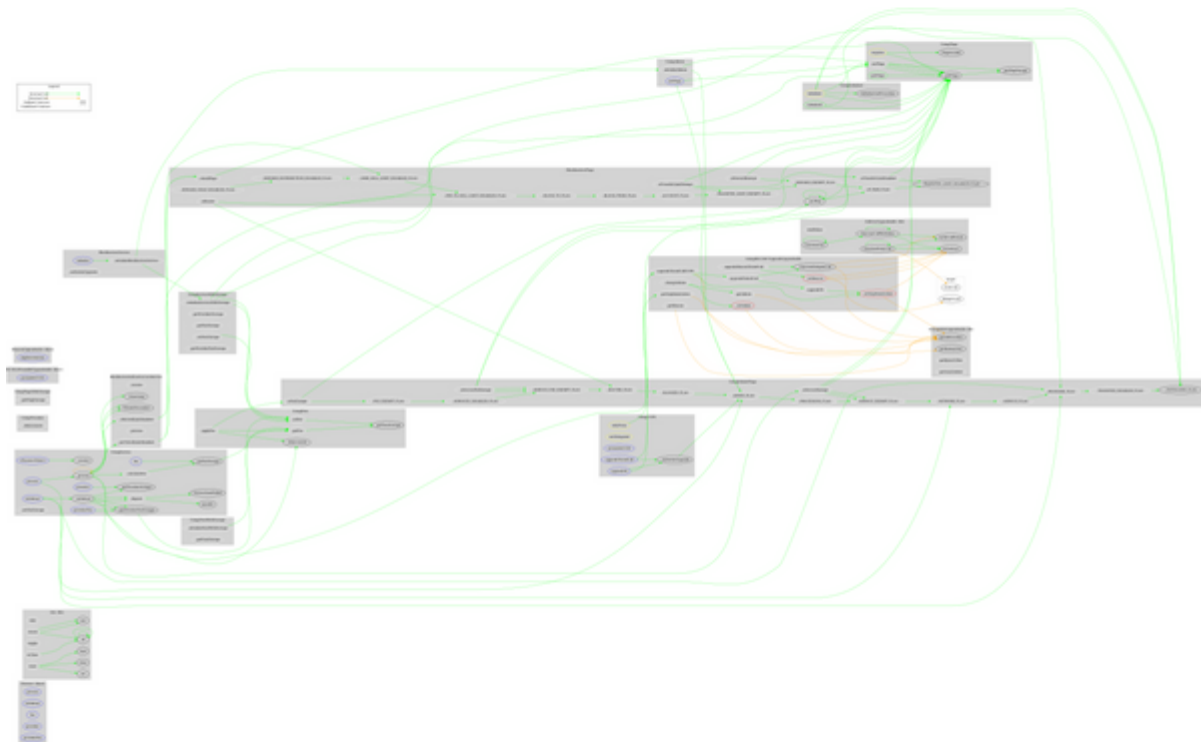
	process	External	✓	requires
	withdraw	External	✓	requires
	provider	External		-
	providerFee	External		-
	fee	External		-
	_calculateFee	Internal		
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	_process	Internal	✓	
	_receive	Internal	✓	
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getProviderFeeStorage	Internal		
UsingFlagsWithStorage	Implementation	UsingFlags		
	_getFlagStorage	Internal		
UsingFeesWithStorage	Implementation	UsingFees		
	_initializeFeesWithStorage	Internal	✓	
	_getFeesStorage	Internal		
UsingServiceWithStorage	Implementation	UsingService, UsingFees WithStorage		
	_initializeServiceWithStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderFeeStorage	Internal		
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		

	initialized	Public		-
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	UsingFlags WithStorage , BlockbustersFlags		
IERC1822ProxiableUpgradable	Interface			
	proxiableUUID	External		-

IBeaconUpgradable	Interface			
	implementation	External		-
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967 UpgradeUpgradable	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	✓	
BlockbustersService	Implementation	UsingServiceWithStorage, UsingInitializer, BlockbustersFlagsWithStorage, UsingUUPS		
	_initializeBlockbustersService	Internal	✓	
	initialize	External	✓	initializer
	_authorizeUpgrade	Internal	✓	requires
BlockbustersInfrastructureService	Implementation	BlockbustersService		
	_receive	Internal	✓	
	_perTxSellLimitDisabled	Internal	✓	
	_24hrSellLimitDisabled	Internal	✓	
	_timestamp	Internal		
	_process	Internal	✓	

Contract Flow



Domain Info

Domain Name	bttftoken.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

This audit focuses on the business logic issues, the security concerns and the potential improvements. The owner has the authority to upgrade the contract via proxy.

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.

Cyberscope 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>



Cyberscope

Audit Report

Blockbusters Liquidity Service

August 2022

SHA256 ae669bd095262f08a04990bf860eaf95a06ce4ca5df8583ede4041ea46d277a1

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Introduction	4
Contract Diagnostics	5
ST - Stops Transactions	6
Description	6
Recommendation	6
STC - Succeeded Transfer Check	7
Description	7
Recommendation	7
CO - Code Optimization	8
Description	8
Recommendation	8
MC - Missing Check	9
Description	9
Recommendation	9
L01 - Public Function could be Declared External	10
Description	10
Recommendation	10
L02 - State Variables could be Declared Constant	11
Description	11
Recommendation	11
L04 - Conformance to Solidity Naming Conventions	12
Description	12

Recommendation	12
L05 - Unused State Variable	13
Description	13
Recommendation	13
L09 - Dead Code Elimination	14
Description	14
Recommendation	14
L12 - Using Variables before Declaration	15
Description	15
Recommendation	15
L13 - Divide before Multiply Operation	16
Description	16
Recommendation	16
L14 - Uninitialized Variables in Local Scope	17
Description	17
Recommendation	17
L15 - Local Scope Variable Shadowing	18
Description	18
Recommendation	18
Contract Functions	19
Contract Flow	31
Domain Info	32
Summary	33
Disclaimer	34
About Cyberscope	35

Contract Review

Contract Name	BlockbustersLiquidityService
Compiler Version	v0.8.15+commit.e14f2714
Testing Deploy	https://testnet.bscscan.com/token/0x12A97Cb066B8F00fA8284f1F693Fba61dff827b3
Domain	https://bbtftoken.com

Source Files

Filename	SHA256
contract.sol	ae669bd095262f08a04990bf860eaf95a06ce4ca5df8583e de4041ea46d277a1

Audit Updates

Initial Audit	23rd August 2022
Corrected	

Introduction

The Blockbusters Liquidity Service handles the functionality that is related to the liquidity pool. Additionally, it provides assisting methods between the token and the decentralized exchanges.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	STC	Succeeded Transfer Check	Unresolved
●	CO	Code Optimization	Unresolved
●	MC	Missing Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L05	Unused State Variable	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

ST - Stops Transactions

Criticality	medium
Location	contract.sol#L938
Status	Unresolved

Description

The contract is going to revert and stop the `_process` transaction on every `_calculateFee`, if the amount is lower than the fixed fee.

```
function _process(address from_, address to_, uint256 amount_) internal virtual returns (uint256){  
    return _isServiceExempt(from_, to_) ? 0 : _calculateFee(from_, to_, amount_);  
}
```

Recommendation

A suggested implementation could apply the fees on each amount that is processed.

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L1881
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

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

Recommendation

The contract should check if the result of the transfer methods is successful.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L926,1866
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

Since the `_withdraw` may transfer all the balance, the loop is redundant. It should break if the contract balance is zero.

```
function _withdraw(address to_) internal virtual override {  
    for (uint i = 0; i < _getServicesStorage().length(); i++) {  
        IService(_getServicesStorage().getAsAddress(i)).withdraw(to_);  
    }  
    super._withdraw(to_);  
}
```

The variable fee is not used in the function implementation. As a result its is redundant.

```
function _process(address from_, address to_, uint256 amount_) internal override returns  
(uint256) {  
    uint totalFee = super._process(from_, to_, amount_);  
    uint fee = totalFee * amount_ / _PRECISION();
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L2049
Status	Unresolved

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 method `_withdraw` should check if there is sufficient balance before every deposit.

```
function _withdraw(address to_) internal override {  
    _deposit(to_, address(this).balance);  
}  
  
function _deposit(address account_, uint value_) internal virtual {  
    (bool success,) = payable(account_).call{value: value_}("");  
    if (!success) {  
        revert ServiceSendFailed();  
    }  
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L589,567,596,528,560,579,396,584,402,564,210,101
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.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L1145
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_maxPrivateSalePercentage
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L148,1220,6,731,1200,136,164,1188,1208,1115,1138,128,1145,1111,172,1184,1196,140,1204,156,1058,1192,1216,1767,848,152,1760,1180,2030,132,1845,168,1088,1144,738,218,144,1137,1212,1041,176,160
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
_REWARD_SWAP_DISABLED_FLAG
bits
_flags
_BLOCK_FROM_FLAG
_PROVIDER_FLAG
_ROUTER_FLAG
_REWARD_EXEMPT_FLAG
_PER_TX_SELL_LIMIT_DISABLED_FLAG
...
```

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#L1845,1145
Status	Unresolved

Description

There are segments that contain unused state variables.

```
__gap  
_maxPrivateSalePercentage
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L1401,1248,2040,1236,787,1240,457,1216,1196,407,799,184,1244,1372,642,719,1559,653,1382,71,1714,1347,926,1707,55,693,1075,1043,1192,176,1729,1434,1204,1678,1117,1671,773,2019,981,389,1200,2015,715,438,621,760,603,839,168,1550,468,803,172,1208,769,1188,12,1103,1220,17,795,1688,1444,51,180,875,1232,2032,1224,709,1212,1415
Status	Unresolved

Description

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

```
functionCallWithValue
_checkFlags
_getTokensStorage
_isRewardExempt
next
_isTransferLimitExempt
_domainSeparator
_REWARD_DISTRIBUTION_DISABLED_FLAG
_ACCOUNT_FLAG
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L1647
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.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L1008
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
value = _getFeesStorage()[service] * 10 ** 18 / totalFee * value_ / 10 ** 18
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L1647
Status	Unresolved

Description

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

slot

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L997,930,1866,929
Status	Unresolved

Description

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

```
fee
providerFee
provider
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
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		
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		
	initialized	Public		-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-

IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IService	Interface			
	process	External	✓	-
	withdraw	External	✓	-
	fee	External		-
	provider	External		-
	providerFee	External		-
IServiceProvider	Interface	IService		
	removeServices	External	✓	-
	addServices	External	✓	-
	services	External		-
ISwap	Interface	IServiceProvider		
	quote	External		-
	swapNativeForTokens	External	Payable	-
	swapTokensForNative	External	✓	-
	swapTokensForNativeWithPermit	External	✓	-
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		

Sets	Library			
	add	Internal	✓	
	remove	Internal	✓	
	get	Internal		
	pop	Internal	✓	
	contains	Internal		
	length	Internal		
UsingFlagsWithStorage	Implementation	UsingFlags		
	_getFlagStorage	Internal		
collections	Library			
	add	Internal	✓	
	add	Internal	✓	
	replace	Internal	✓	
	replace	Internal	✓	
	pop	Internal	✓	
	get	Internal		
	getAsAddress	Internal		
	next	Internal	✓	
	current	Internal		
	currentAsAddress	Internal		
	nextAsAddress	Internal	✓	
	length	Internal		
	remove	Internal	✓	
	remove	Internal	✓	
	itemsAsAddresses	Internal		
	indexOf	Internal		
UsingPrecision	Implementation			
	_PRECISION	Internal		

UsingAdmin	Implementation	UsingFlags, UsingDefaultFlags		
	_initializeAdmin	Internal	✓	
	setFlags	External	✓	requires
UsingFees	Implementation	UsingDefaultFlags, UsingPrecision		
	_setFee	Internal	✓	
	_getFee	Internal		
	_applyFee	Internal		
	_getFeesStorage	Internal		
UsingService	Implementation	IService, UsingAdmin, UsingFees		
	<Receive Ether>	External	Payable	-
	process	External	✓	requires
	withdraw	External	✓	requires
	provider	External		-
	providerFee	External		-
	fee	External		-
	_calculateFee	Internal		
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	_process	Internal	✓	
	_receive	Internal	✓	
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getProviderFeeStorage	Internal		
UsingServiceProvider	Implementation	IServiceProvider, UsingService		

	services	External		-
	addServices	External	✓	requires
	removeServices	External	✓	requires
	setServiceFee	External	✓	requires
	_removeServices	Internal	✓	
	_addServices	Internal	✓	
	_withdraw	Internal	✓	
	_processService	Internal	✓	
	_processServices	Internal	✓	
	_depositServiceFees	Internal	✓	
	_addService	Internal	✓	
	_removeService	Internal	✓	
	_getServicesStorage	Internal		
UsingFeesWithStorage	Implementation	UsingFees		
	_initializeFeesWithStorage	Internal	✓	
	_getFeesStorage	Internal		
UsingServiceWithStorage	Implementation	UsingService, UsingFeesWithStorage		
	_initializeServiceWithStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderFeeStorage	Internal		
UsingServiceProviderWithStorage	Implementation	UsingServiceProvider, UsingServiceWithStorage		
	_initializeServiceProviderWithStorage	Internal	✓	
	_getServicesStorage	Internal		
	_withdraw	Internal	✓	

UsingPermitWithStorage	Implementation	UsingPermit		
	_initializePermitWithStorage	Internal	✓	
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		
UsingSwapWithStorage	Implementation	UsingServiceProviderWithStorage		
	_initializeSwapWithStorage	Internal	✓	
	_addRouter	Internal	✓	
	_process	Internal	✓	
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		
IERC1822ProxiableUpgradable	Interface			
	proxiableUUID	External		-
IBeaconUpgradable	Interface			
	implementation	External		-
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967UpgradeUpgradable	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	✓	
BlockbustersSwap	Implementation	UsingSwapWithStorage, BlockbustersFlagsWithStorage, UsingInitializer, UsingUUPS		
	initialize	External	✓	initializer
	setup	External	✓	requires
	_process	Internal	✓	
	withdrawTokens	External	✓	requires
	_withdraw	Internal	✓	
	_authorizeUpgrade	Internal	✓	requires

IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
BlockbustersService	Implementation	UsingServiceWithStorage, UsingInitializer, BlockbustersFlagsWithStorage, UsingUUPS		
	_initializeBlockbustersService	Internal	✓	
	initialize	External	✓	initializer
	_authorizeUpgrade	Internal	✓	requires
UsingMultiToken	Implementation			
	_addToken	Internal	✓	
	_removeToken	Internal	✓	
	_getTokensStorage	Internal		
UsingMultiTokenWithStorage	Implementation	UsingMultiToken		
	_initializeMultiTokenWithStorage	Internal	✓	
	_getTokensStorage	Internal		

BlockbustersLiquidityService	Implementation	BlockbustersService		
	_withdraw	Internal	✓	
	_process	Internal	✓	

Contract Flow



Domain Info

Domain Name	bttftoken.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

This audit focuses on the business logic issues, the security concerns and the potential improvements. The owner has the authority to upgrade the contract via proxy.

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.

Cyberscope 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>



Cyberscope

Audit Report

Blockbusters Rewards Service

August 2022

SHA256 0ad528a23ecf3cf688050a6ca7745db772bff44d0f1c2ecded4caa8171ba60c8

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Introduction	4
Contract Diagnostics	5
ST - Stops Transactions	6
Description	6
Recommendation	6
STC - Succeeded Transfer Check	7
Description	7
Recommendation	7
MC - Missing Check	8
Description	8
Recommendation	9
L01 - Public Function could be Declared External	10
Description	10
Recommendation	10
L02 - State Variables could be Declared Constant	11
Description	11
Recommendation	11
L04 - Conformance to Solidity Naming Conventions	12
Description	12
Recommendation	12
L05 - Unused State Variable	13
Description	13

Recommendation	13
L07 - Missing Events Arithmetic	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
L15 - Local Scope Variable Shadowing	18
Description	18
Recommendation	18
Contract Functions	19
Contract Flow	31
Domain Info	32
Summary	33
Disclaimer	34
About Cyberscope	35

Contract Review

Contract Name	BlockbustersRewardsService
Compiler Version	v0.8.15+commit.e14f2714
Testing Deploy	https://testnet.bscscan.com/token/0x59d5e91f09625ACE9891190F7aB6124A01B832F7
Domain	https://bbtftoken.com

Source Files

Filename	SHA256
contract.sol	0ad528a23ecf3cf688050a6ca7745db772bff44d0f1c2ecd ed4caa8171ba60c8

Audit Updates

Initial Audit	23rd August 2022
Corrected	

Introduction

The Blockbusters Rewards Service contract is responsible for handling the reward functionality. The main tasks are to keep the reward balance updated and distribute the rewards proportionally to the users.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	STC	Succeeded Transfer Check	Unresolved
●	MC	Missing Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L05	Unused State Variable	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

ST - Stops Transactions

Criticality	medium
Location	contract.sol#L296
Status	Unresolved

Description

The contract is going to revert and stop the `_process` transaction on every `_calculateFee`, if the amount is lower than the fixed fee.

```
function _process(address from_, address to_, uint256 amount_) internal virtual returns (uint256){  
    return _isServiceExempt(from_, to_) ? 0 : _calculateFee(from_, to_, amount_);  
}
```

Recommendation

A suggested implementation could apply the fees on each amount that is processed.

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L2187
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(reward_).transfer(account_, amount * _rewardDivisor);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L2156
Status	Unresolved

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 method `_withdraw` should check if there is sufficient balance before every deposit.

```
function _withdraw(address to_) internal virtual {
    uint balance = address(this).balance;
    if (balance > 0) {
        address provider = _getProviderStorage();
        uint providerFee = _getProviderFeeStorage();
        if ( provider != address(0) && providerFee > 0) {
            _deposit(provider, balance * _getProviderFeeStorage() / _PRECISION());
        }
        _deposit(to_, address(this).balance);
    }
}
```

Prior to the `swapExactETHForTokensSupportingFeeOnTransferTokens` method the contract should check if there is sufficient balance before every swap.

```
function _swapForReward(address reward_, uint value_) internal {
    Snapshot storage rewardSnapshot = _snapshots[reward_];
    address[] storage path = _rewardPaths[reward_];
    uint value = value_ * _rewardRatio[reward_] / DEFAULT_PRECISION;
    uint oldRewardBalance = IERC20(reward_).balanceOf(address(this));

    IUniswapV2Router02(_rewardRouters[reward_]).swapExactETHForTokensSupportingFeeOnTransferTokens{value: value}(0, path, address(this), block.timestamp + 1);
    _updateAccountSnapshot(rewardSnapshot, reward_,
        _scaleAmount(IEC20(reward_).balanceOf(address(this)) - oldRewardBalance), 0);
}
```

The method `_process` should check for reentrance.

```
function _process(address from_, address to_, uint amount_) internal override returns (uint256) {
    uint toFlags = _getFlags(to_);
    uint fromFlags = _getFlags(from_);
    if (!_isRewardSwapDisabled()) {
        uint balance = address(this).balance;
        if (balance > 0) {
            _swapForRewards(balance);
        }
    }
    uint128 amount = _scaleAmount(amount_);
    if (!_isRewardExempt(to_)) {
        if (!_isAccount(to_)) _addAccount(to_);
        _updateSnapshot(to_, amount, 0);
        _updateSnapshot(address(this), amount, 0);
    }
    if (!_isRewardExempt(from_)) {
        _updateSnapshot(from_, 0, amount);
        _updateSnapshot(address(this), 0, amount);
    }
    if (!_isRewardDistributionDisabled()) {
        _distribute(_isLPPair(from_), _isLPPair(to_));
    }
    return super._process(from_, to_, amount_);
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L1471,2025,1449,1478,1410,1442,1461,1278,1466,1284,1446,381,110
Status	Unresolved

Description

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

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

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L1934
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_reserved
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L165,1588,430,1926,15,312,410,153,1931,181,398,418,1564,1924,145,1560,1933,189,1930,394,406,1928,157,414,1590,173,338,402,1585,426,977,136,1925,169,970,1932,390,1884,149,1055,185,1725,1927,1124,161,422,321,1849,1929,193,1934,177
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
_rewardRouters
bits
_flags
_BLOCK_FROM_FLAG
_PROVIDER_FLAG
_distributionsPerBuy
...
```

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#L1934,1055
Status	Unresolved

Description

There are segments that contain unused state variables.

```
_reserved  
__gap
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L1983
Status	Unresolved

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.

```
(_distributionsPerBuy,_distributionsPerSell,_distributionsPerTransfer) =  
(distributionsPerBuy_,distributionsPerSell_,distributionsPerTransfer_)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L181,611,458,1592,450,1339,1289,1786,201,454,582,1524,1116,769,1535,592,80,924,557,917,64,1090,355,323,193,939,644,888,1566,1615,881,1760,1873,177,1271,1112,272,1320,1503,1485,1818,1855,185,760,1350,189,21,26,1782,898,654,60,197,232,2082,442,1886,434,1106,1102,625
Status	Unresolved

Description

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

```
_ROUTER_FLAG  
functionCallWithValue  
_checkFlags  
_initializeERC20WithStorage  
_isTransferLimitExempt  
_domainSeparator  
_permit  
currentAsAddress  
_isServiceFeeExempt  
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L857
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#L857,2183
Status	Unresolved

Description

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

```
slot  
amount
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L2026,287,286
Status	Unresolved

Description

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

```
accounts
providerFee
provider
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IService	Interface			
	process	External	✓	-
	withdraw	External	✓	-
	fee	External		-
	provider	External		-
	providerFee	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		
UsingPrecision	Implementation			
	_PRECISION	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		
UsingAdmin	Implementation	UsingFlags, UsingDefaultFlags		
	_initializeAdmin	Internal	✓	
	setFlags	External	✓	requires
UsingFees	Implementation	UsingDefaultFlags, UsingPrecision		
	_setFee	Internal	✓	
	_getFee	Internal		
	_applyFee	Internal		
	_getFeesStorage	Internal		
UsingService	Implementation	IService, UsingAdmin, UsingFees		
	<Receive Ether>	External	Payable	-

	process	External	✓	requires
	withdraw	External	✓	requires
	provider	External		-
	providerFee	External		-
	fee	External		-
	_calculateFee	Internal		
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	_process	Internal	✓	
	_receive	Internal	✓	
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getProviderFeeStorage	Internal		
UsingFlagsWithStorage	Implementation	UsingFlags		
	_getFlagStorage	Internal		
UsingFeesWithStorage	Implementation	UsingFees		
	_initializeFeesWithStorage	Internal	✓	
	_getFeesStorage	Internal		
UsingServiceWithStorage	Implementation	UsingService, UsingFees WithStorage		
	_initializeServiceWithStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderFeeStorage	Internal		
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		

	initialized	Public		-
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	UsingFlags WithStorage , BlockbustersFlags		
IERC1822ProxiableUpgradable	Interface			
	proxiableUUID	External		-

IBeaconUpgradable	Interface			
	implementation	External		-
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967 UpgradeUpgradable	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	✓	
BlockbustersService	Implementation	UsingServiceWithStorage, UsingInitializer, BlockbustersFlagsWithStorage, UsingUUPS		
	_initializeBlockbustersService	Internal	✓	
	initialize	External	✓	initializer
	_authorizeUpgrade	Internal	✓	requires
Sets	Library			
	add	Internal	✓	
	remove	Internal	✓	
	get	Internal		
	pop	Internal	✓	
	contains	Internal		
	length	Internal		
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-

	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
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	✓	

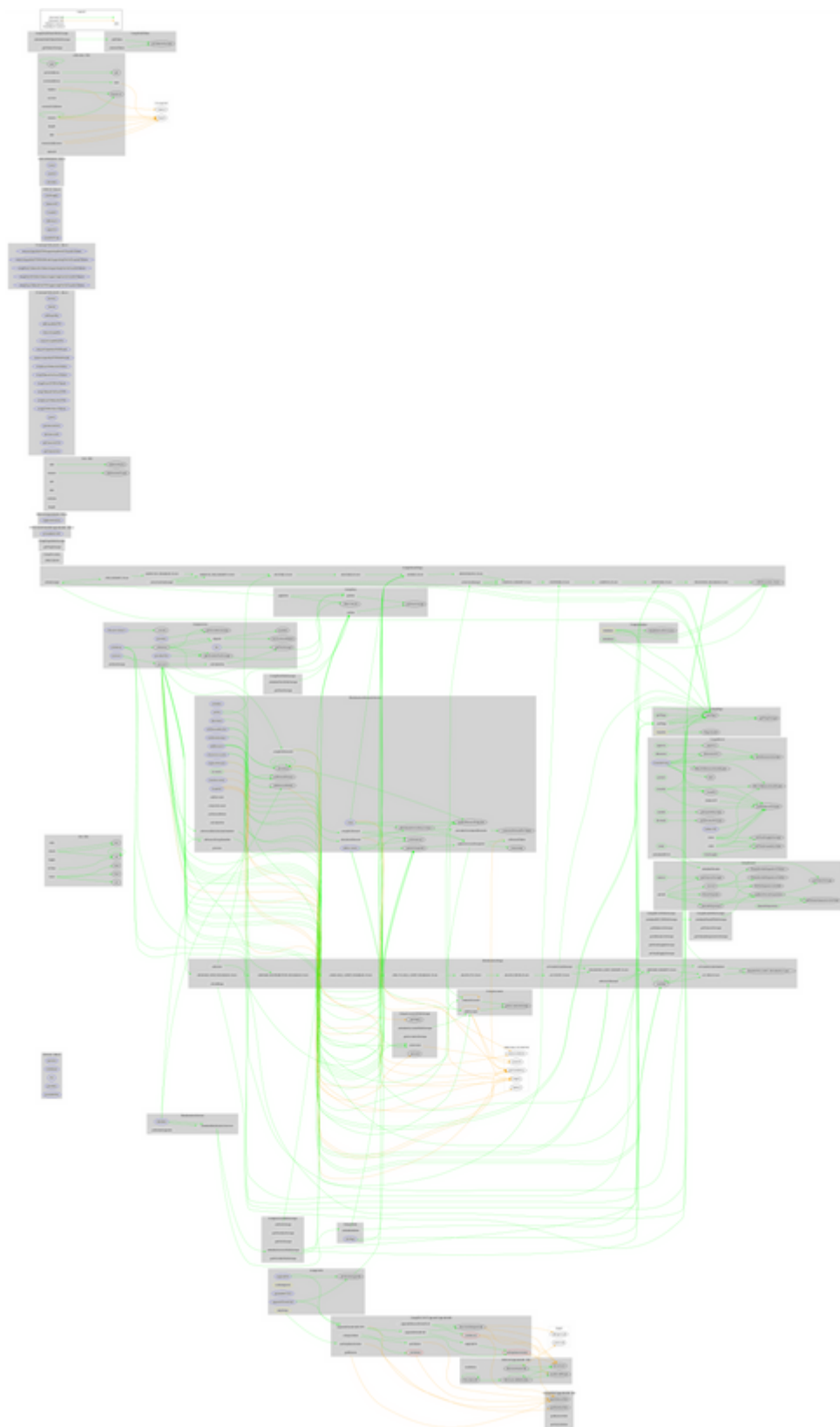
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		
UsingERC20WithStorage	Implementation	UsingERC20, UsingPermitWithStorage		
	_initializeERC20WithStorage	Internal	✓	
	_getBalancesStorage	Internal		
	_getAllowanceStorage	Internal		
	_getTotalSupplyStorage	Internal		
	_setTotalSupplyStorage	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
collections	Library			
	add	Internal	✓	
	add	Internal	✓	
	replace	Internal	✓	
	replace	Internal	✓	
	pop	Internal	✓	
	get	Internal		
	getAsAddress	Internal		
	next	Internal	✓	

	current	Internal		
	currentAsAddress	Internal		
	nextAsAddress	Internal	✓	
	length	Internal		
	remove	Internal	✓	
	remove	Internal	✓	
	itemsAsAddresses	Internal		
	indexOf	Internal		
UsingAccounts	Implementation			
	_addAccount	Internal	✓	
	_removeAccount	Internal	✓	
	_getAccountsStorage	Internal		
UsingAccountsWithStorage	Implementation	UsingAccounts		
	_isAccount	Internal	✓	
	_initializeAccountsWithStorage	Internal	✓	
	_getAccountsStorage	Internal		
UsingMultiToken	Implementation			
	_addToken	Internal	✓	
	_removeToken	Internal	✓	
	_getTokensStorage	Internal		
UsingMultiTokenWithStorage	Implementation	UsingMultiToken		
	_initializeMultiTokenWithStorage	Internal	✓	
	_getTokensStorage	Internal		

BlockbustersRewardsService	Implementation	Blockbuster sService, UsingMultiT okenWithSt orage, UsingAccou ntsWithStor age		
	initialize	External	✓	initializer
	setFee	External	✓	requires
	addAccounts	External	✓	requires
	removeAccounts	External	✓	requires
	setRewardRouter	External	✓	requires
	setDistributions	External	✓	requires
	replaceReward	External	✓	requires
	addReward	External	✓	requires
	claim	External	✓	requires
	distribute	External	✓	-
	accounts	Public		-
	totalAccounts	External		-
	snapshot	External		-
	_addAccount	Internal	✓	
	_removeAccount	Internal	✓	
	_addRewardToken	Internal	✓	
	_setRewardRouter	Internal	✓	
	_setRewardRatio	Internal	✓	
	_timestamp	Internal		
	_calculateUnclaimedRewards	Internal		
	_calculateRewardPerToken	Internal		
	_updateRewardSnapshot	Internal	✓	
	_isRewardToken	Internal	✓	
	_updateAccountSnapshot	Internal	✓	
	_updateSnapshot	Internal	✓	
	_scaleAmount	Internal		
	_calculateFee	Internal		
	_swapForReward	Internal	✓	
	_swapForRewards	Internal	✓	

	_isRewardDistributionDisabled	Internal		
	_isRewardSwapDisabled	Internal		
	_process	Internal	✓	
	_distributeRewardToAccount	Internal	✓	
	_distributeReward	Internal	✓	
	_distribute	Internal	✓	
	_distribute	Internal	✓	

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

This audit focuses on the business logic issues, the security concerns and the potential improvements. The owner has the authority to upgrade the contract via proxy.

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.

Cyberscope 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>



Cyberscope

Audit Report

Blockbusters Staking Service

August 2022

SHA256 d9632d47b3d8708266f5d4a469a2310561ae661d0307cc670002af76ba473f4b

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Introduction	4
Contract Diagnostics	5
BLC - Business Logic Concern	6
Description	6
Recommendation	6
L01 - Public Function could be Declared External	7
Description	7
Recommendation	7
L02 - State Variables could be Declared Constant	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L05 - Unused State Variable	10
Description	10
Recommendation	10
L09 - Dead Code Elimination	10
Description	10
Recommendation	11
L12 - Using Variables before Declaration	12
Description	12

Recommendation	12
L13 - Divide before Multiply Operation	13
Description	13
Recommendation	13
L14 - Uninitialized Variables in Local Scope	14
Description	14
Recommendation	14
L15 - Local Scope Variable Shadowing	15
Description	15
Recommendation	15
Contract Functions	16
Contract Flow	28
Domain Info	29
Summary	30
Disclaimer	31
About Cyberscope	32

Contract Review

Contract Name	BlockbustersStakingService
Compiler Version	v0.8.15+commit.e14f2714
Testing Deploy	https://testnet.bscscan.com/token/0x4601187513D6257CBB60aeC13654793Fe21B2312
Domain	https://bbtftoken.com

Source Files

Filename	SHA256
contract.sol	d9632d47b3d8708266f5d4a469a2310561ae661d0307cc670002af76ba473f4b

Audit Updates

Initial Audit	23rd August 2022
Corrected	

Introduction

The Blockbusters Staking Service contract for processing the fees of a transaction. It is intended to be used by a token as an external service in order to manipulate the fees of the trade. The contract's name is misleading.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	BLC	Business Logic Concern	Unresolved
●	MC	Missing Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L05	Unused State Variable	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

BLC - Business Logic Concern

Criticality	minor / informative
Location	contract.sol#L251
Status	Unresolved

Description

The business logic seems peculiar. The implementation may not follow the expected behavior.

The calculation is not taking in account the transferred amount, it's always fixed. So trades with less than this amount will revert.

```
function _calculateFee(address from_, address to_, uint amount_) internal view override returns (uint){  
    if (_isLPPair(to_)) return _getFeeStorage();  
    return 0;  
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L589,567,596,528,560,579,396,584,402,564,210,101
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.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L1145
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_maxPrivateSalePercentage
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L148,1220,6,731,1200,136,164,1188,1208,1115,1138,128,1145,1111,172,1184,1196,140,1204,156,1058,1192,1216,1767,848,152,1760,1180,2030,132,1845,168,1088,1144,738,218,144,1137,1212,1041,176,160
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
_REWARD_SWAP_DISABLED_FLAG
bits
_flags
_BLOCK_FROM_FLAG
_PROVIDER_FLAG
_ROUTER_FLAG
_REWARD_EXEMPT_FLAG
_PER_TX_SELL_LIMIT_DISABLED_FLAG
...
```

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#L1845,1145
Status	Unresolved

Description

There are segments that contain unused state variables.

```
__gap  
_maxPrivateSalePercentage
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L1401,1248,2040,1236,787,1240,457,1216,1196,407,799,184,1244,1372,642,719,1559,653,1382,71,1714,1347,926,1707,55,693,1075,1043,1192,176,1729,1434,1204,1678,1117,1671,773,2019,981,389,1200,2015,715,915,438,621,760,603,839,168,1550,468,803,172,1208,769,1188,12,1103,1220,17,795,1688,1444,51,180,875,1232,2032,1224,709,1212,1415
Status	Unresolved

Description

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

```
functionCallWithValue
_checkFlags
_getTokensStorage
_isRewardExempt
next
_isTransferLimitExempt
_domainSeparator
_REWARD_DISTRIBUTION_DISABLED_FLAG
_ACCOUNT_FLAG
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L1647
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.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L1008
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
value = _getFeesStorage()[service] * 10 ** 18 / totalFee * value_ / 10 ** 18
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L1647
Status	Unresolved

Description

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

slot

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L997,930,1866,929
Status	Unresolved

Description

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

```
fee
providerFee
provider
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
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		
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		
	initialized	Public		-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-

IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IService	Interface			
	process	External	✓	-
	withdraw	External	✓	-
	fee	External		-
	provider	External		-
	providerFee	External		-
IServiceProvider	Interface	IService		
	removeServices	External	✓	-
	addServices	External	✓	-
	services	External		-
ISwap	Interface	IServiceProvider		
	quote	External		-
	swapNativeForTokens	External	Payable	-
	swapTokensForNative	External	✓	-
	swapTokensForNativeWithPermit	External	✓	-
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		

Sets	Library			
	add	Internal	✓	
	remove	Internal	✓	
	get	Internal		
	pop	Internal	✓	
	contains	Internal		
	length	Internal		
UsingFlagsWithStorage	Implementation	UsingFlags		
	_getFlagStorage	Internal		
collections	Library			
	add	Internal	✓	
	add	Internal	✓	
	replace	Internal	✓	
	replace	Internal	✓	
	pop	Internal	✓	
	get	Internal		
	getAsAddress	Internal		
	next	Internal	✓	
	current	Internal		
	currentAsAddress	Internal		
	nextAsAddress	Internal	✓	
	length	Internal		
	remove	Internal	✓	
	remove	Internal	✓	
	itemsAsAddresses	Internal		
	indexOf	Internal		
UsingPrecision	Implementation			
	_PRECISION	Internal		

UsingAdmin	Implementation	UsingFlags, UsingDefaultFlags		
	_initializeAdmin	Internal	✓	
	setFlags	External	✓	requires
UsingFees	Implementation	UsingDefaultFlags, UsingPrecision		
	_setFee	Internal	✓	
	_getFee	Internal		
	_applyFee	Internal		
	_getFeesStorage	Internal		
UsingService	Implementation	IService, UsingAdmin, UsingFees		
	<Receive Ether>	External	Payable	-
	process	External	✓	requires
	withdraw	External	✓	requires
	provider	External		-
	providerFee	External		-
	fee	External		-
	_calculateFee	Internal		
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	_process	Internal	✓	
	_receive	Internal	✓	
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getProviderFeeStorage	Internal		
UsingServiceProvider	Implementation	IServiceProvider, UsingService		

	services	External		-
	addServices	External	✓	requires
	removeServices	External	✓	requires
	setServiceFee	External	✓	requires
	_removeServices	Internal	✓	
	_addServices	Internal	✓	
	_withdraw	Internal	✓	
	_processService	Internal	✓	
	_processServices	Internal	✓	
	_depositServiceFees	Internal	✓	
	_addService	Internal	✓	
	_removeService	Internal	✓	
	_getServicesStorage	Internal		
UsingFeesWithStorage	Implementation	UsingFees		
	_initializeFeesWithStorage	Internal	✓	
	_getFeesStorage	Internal		
UsingServiceWithStorage	Implementation	UsingService, UsingFeesWithStorage		
	_initializeServiceWithStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderFeeStorage	Internal		
UsingServiceProviderWithStorage	Implementation	UsingServiceProvider, UsingServiceWithStorage		
	_initializeServiceProviderWithStorage	Internal	✓	
	_getServicesStorage	Internal		
	_withdraw	Internal	✓	

UsingPermitWithStorage	Implementation	UsingPermit		
	_initializePermitWithStorage	Internal	✓	
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		
UsingSwapWithStorage	Implementation	UsingServiceProviderWithStorage		
	_initializeSwapWithStorage	Internal	✓	
	_addRouter	Internal	✓	
	_process	Internal	✓	
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		

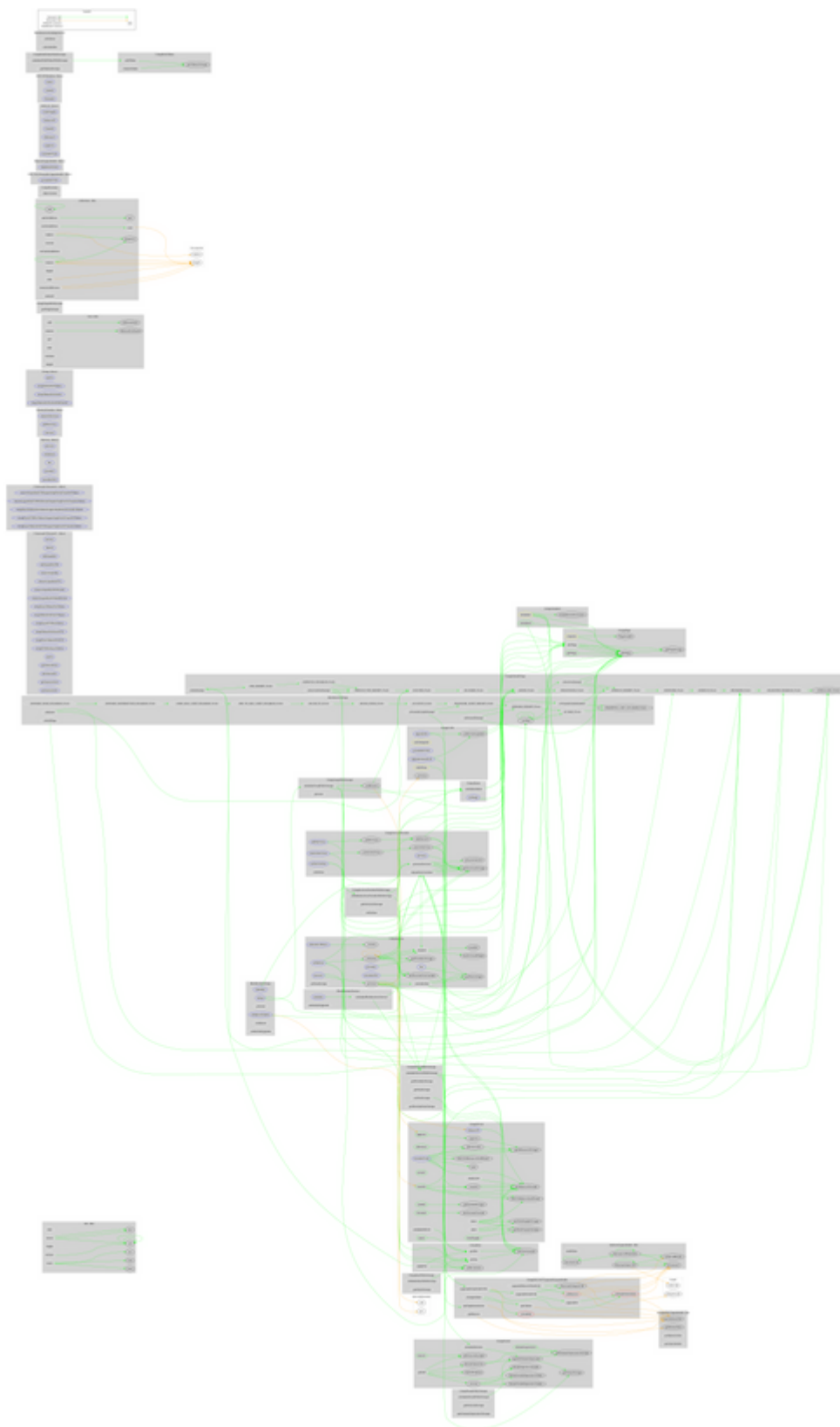
	_checkFlags	Internal		
BlockbustersFlagsWithStorage	Implementation	UsingFlags WithStorage , BlockbustersFlags		
IERC1822ProxiableUpgradable	Interface			
	proxiableUUID	External		-
IBeaconUpgradable	Interface			
	implementation	External		-
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967UpgradeUpgradable	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	IERC1822ProxiableUpgradable, UsingERC1967UpgradeUpgradeable		
	proxiableUUID	External		notDelegated
	upgradeTo	External	✓	onlyProxy
	upgradeToAndCall	External	Payable	onlyProxy
	_authorizeUpgrade	Internal	✓	
BlockbustersSwap	Implementation	UsingSwapWithStorage, BlockbustersFlagsWithStorage, UsingInitializer, UsingUUPS		
	initialize	External	✓	initializer
	setup	External	✓	requires
	_process	Internal	✓	
	withdrawTokens	External	✓	requires
	_withdraw	Internal	✓	
	_authorizeUpgrade	Internal	✓	requires

IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
BlockbustersService	Implementation	UsingServiceWithStorage, UsingInitializer, BlockbustersFlagsWithStorage, UsingUUPS		
	_initializeBlockbustersService	Internal	✓	
	initialize	External	✓	initializer
	_authorizeUpgrade	Internal	✓	requires
UsingMultiToken	Implementation			
	_addToken	Internal	✓	
	_removeToken	Internal	✓	
	_getTokensStorage	Internal		
UsingMultiTokenWithStorage	Implementation	UsingMultiToken		
	_initializeMultiTokenWithStorage	Internal	✓	
	_getTokensStorage	Internal		

BlockbustersStakingService	Implementation	BlockbustersService		
	_withdraw	Internal	✓	
	_calculateFee	Internal		

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

This audit focuses on the business logic issues, the security concerns and the potential improvements. The owner has the authority to upgrade the contract via proxy. The staking name is misleading since this contract is not related to staking.

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.

Cyberscope 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>



Cyberscope

Audit Report

Blockbusters Swap

August 2022

SHA256 04291ebe717a1dcc37239b7daf3ebfbbd5926333dba78462831bf22foe100eb9

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Introduction	4
Contract Diagnostics	5
STC - Succeeded Transfer Check	6
Description	6
Recommendation	6
CO - Code Optimization	7
Description	7
Recommendation	7
MC - Missing Check	8
Description	8
Recommendation	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L05 - Unused State Variable	12
Description	12

Recommendation	12
L09 - Dead Code Elimination	13
Description	13
Recommendation	13
L12 - Using Variables before Declaration	14
Description	14
Recommendation	14
L13 - Divide before Multiply Operation	15
Description	15
Recommendation	15
L14 - Uninitialized Variables in Local Scope	16
Description	16
Recommendation	16
L15 - Local Scope Variable Shadowing	17
Description	17
Recommendation	17
Contract Functions	18
Contract Flow	28
Domain Info	29
Summary	30
Disclaimer	31
About Cyberscope	32

Contract Review

Contract Name	BlockbustersSwap
Compiler Version	v0.8.15+commit.e14f2714
Testing Deploy	https://testnet.bscscan.com/token/0x66C85dA9F9E19F71Cb363189ed5a9E50755F74bE
Domain	https://bbtftoken.com

Source Files

Filename	SHA256
contract.sol	04291ebe717a1dcc37239b7daf3ebfbbd5926333dba78462831bf22fae100eb9

Audit Updates

Initial Audit	23rd August 2022
Corrected	

Introduction

The Blockbusters Swap contract is responsible for liquefying tokens and provides asset withdrawal functionality.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	CO	Code Optimization	Unresolved
●	MC	Missing Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L05	Unused State Variable	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L1881
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

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

Recommendation

The contract should check if the result of the transfer methods is successful.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L1864
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The variable fee is not used in the function implementation. As a result its is redundant.

```
function _process(address from_, address to_, uint256 amount_) internal override returns
(uint256) {
    uint totalFee = super._process(from_, to_, amount_);
    uint fee = totalFee * amount_ / _PRECISION();
    if (_isLPPair(to_)) {
        uint tokenBalance = _token.balanceOf(address(this));
        if (tokenBalance > 0) {
            address router = _routers.get(0);
            _token.approve(router, tokenBalance + 1);
            uint balance = address(this).balance;
            try
            IUniswapV2Router02(router).swapExactTokensForETHSupportingFeeOnTransferTokens(tokenBal
            ance, 0, _paths[router].toNative, address(this), block.timestamp + 1) {
                _depositServiceFees(address(this).balance - balance);
            } catch {}
        }
    }
    return totalFee;
}
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L1881,1884
Status	Unresolved

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 method `_withdraw` should check if there is sufficient balance before every deposit.

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

function _withdraw(address to_) internal override {
    _deposit(to_, address(this).balance);
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L589,567,596,528,560,579,396,584,402,564,210,101
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.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L1145
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_maxPrivateSalePercentage
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L148,1220,6,731,1200,136,164,1188,1208,1115,1138,128,1145,1111,172,1184,1196,140,1204,156,1058,1192,1216,1767,848,152,1760,1180,132,1845,168,1088,1144,738,218,144,1137,1212,1041,176,160
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
_REWARD_SWAP_DISABLED_FLAG
bits
_flags
_BLOCK_FROM_FLAG
_PROVIDER_FLAG
_ROUTER_FLAG
_REWARD_EXEMPT_FLAG
_PER_TX_SELL_LIMIT_DISABLED_FLAG
...
```

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#L1845,1145
Status	Unresolved

Description

There are segments that contain unused state variables.

```
__gap  
_maxPrivateSalePercentage
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L1401,1248,1236,787,1240,457,1216,1196,407,799,184,1244,188,1372,642,719,1559,653,1382,71,1714,1347,926,1707,55,693,1075,1043,1192,176,1729,1434,1204,1678,1117,1671,773,981,389,1200,715,915,438,621,760,603,938,839,168,1550,468,803,172,1208,769,1188,12,1103,1220,17,795,1688,1444,51,180,875,1232,1224,709,1212,1415
Status	Unresolved

Description

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

```
functionCallWithValue
_checkFlags
_isRewardExempt
next
_isTransferLimitExempt
_domainSeparator
_REWARD_DISTRIBUTION_DISABLED_FLAG
_ACCOUNT_FLAG
_permit
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L1647
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.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L1008
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
value = _getFeesStorage()[service] * 10 ** 18 / totalFee * value_ / 10 ** 18
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L1647
Status	Unresolved

Description

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

```
slot
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L997,930,1866,929
Status	Unresolved

Description

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

```
fee
providerFee
provider
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
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		
UsingInitializer	Implementation	UsingFlags, UsingDefaultFlags		
	initialized	Public		-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-

IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IService	Interface			
	process	External	✓	-
	withdraw	External	✓	-
	fee	External		-
	provider	External		-
	providerFee	External		-
IServiceProvider	Interface	IService		
	removeServices	External	✓	-
	addServices	External	✓	-
	services	External		-
ISwap	Interface	IServiceProvider		
	quote	External		-
	swapNativeForTokens	External	Payable	-
	swapTokensForNative	External	✓	-
	swapTokensForNativeWithPermit	External	✓	-
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		

Sets	Library			
	add	Internal	✓	
	remove	Internal	✓	
	get	Internal		
	pop	Internal	✓	
	contains	Internal		
	length	Internal		
UsingFlagsWithStorage	Implementation	UsingFlags		
	_getFlagStorage	Internal		
collections	Library			
	add	Internal	✓	
	add	Internal	✓	
	replace	Internal	✓	
	replace	Internal	✓	
	pop	Internal	✓	
	get	Internal		
	getAsAddress	Internal		
	next	Internal	✓	
	current	Internal		
	currentAsAddress	Internal		
	nextAsAddress	Internal	✓	
	length	Internal		
	remove	Internal	✓	
	remove	Internal	✓	
	itemsAsAddresses	Internal		
	indexOf	Internal		
UsingPrecision	Implementation			
	_PRECISION	Internal		

UsingAdmin	Implementation	UsingFlags, UsingDefaultFlags		
	_initializeAdmin	Internal	✓	
	setFlags	External	✓	requires
UsingFees	Implementation	UsingDefaultFlags, UsingPrecision		
	_setFee	Internal	✓	
	_getFee	Internal		
	_applyFee	Internal		
	_getFeesStorage	Internal		
UsingService	Implementation	IService, UsingAdmin, UsingFees		
	<Receive Ether>	External	Payable	-
	process	External	✓	requires
	withdraw	External	✓	requires
	provider	External		-
	providerFee	External		-
	fee	External		-
	_calculateFee	Internal		
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	_process	Internal	✓	
	_receive	Internal	✓	
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getProviderFeeStorage	Internal		
UsingServiceProvider	Implementation	IServiceProvider, UsingService		

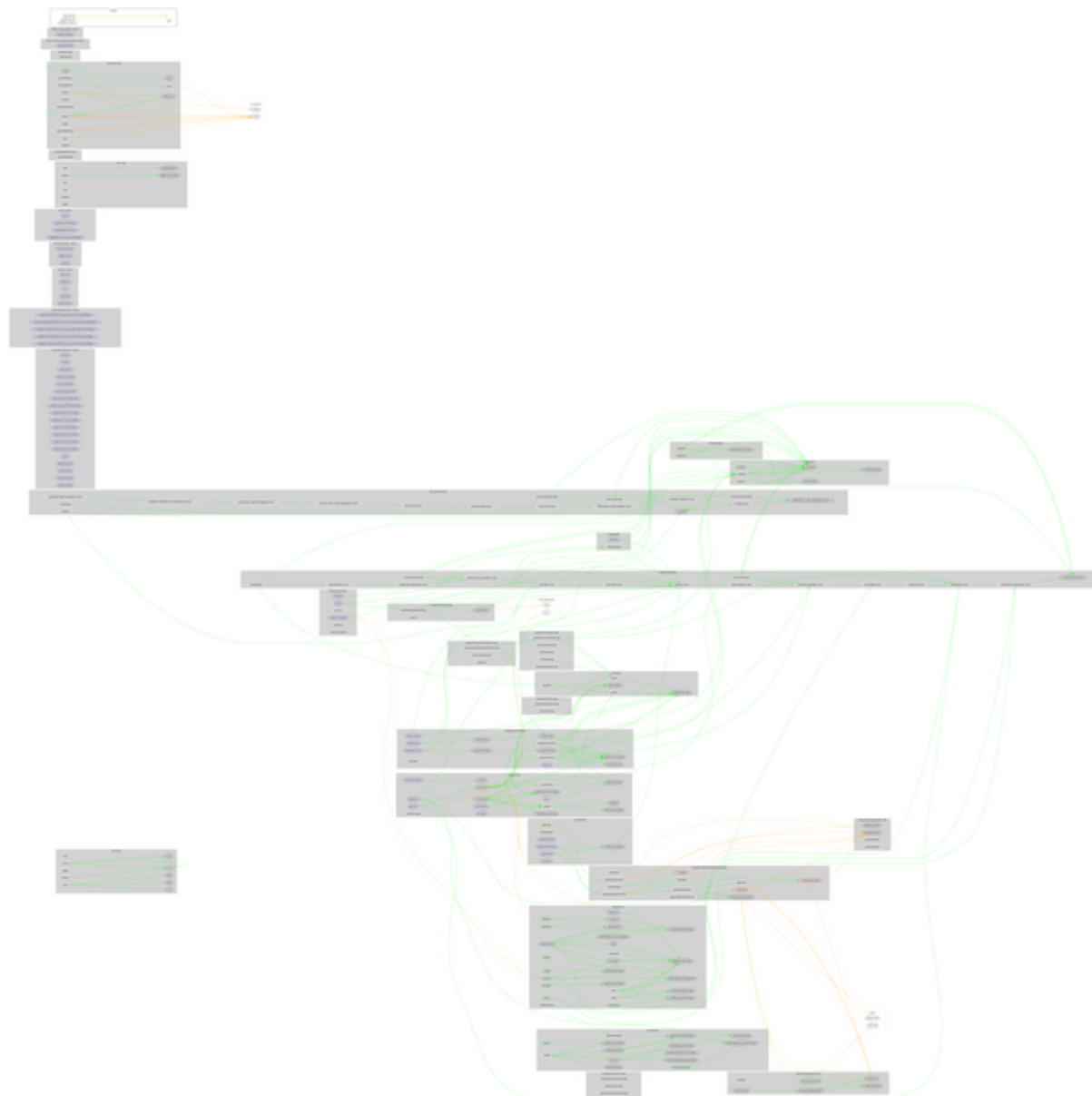
	services	External		-
	addServices	External	✓	requires
	removeServices	External	✓	requires
	setServiceFee	External	✓	requires
	_removeServices	Internal	✓	
	_addServices	Internal	✓	
	_withdraw	Internal	✓	
	_processService	Internal	✓	
	_processServices	Internal	✓	
	_depositServiceFees	Internal	✓	
	_addService	Internal	✓	
	_removeService	Internal	✓	
	_getServicesStorage	Internal		
UsingFeesWithStorage	Implementation	UsingFees		
	_initializeFeesWithStorage	Internal	✓	
	_getFeesStorage	Internal		
UsingServiceWithStorage	Implementation	UsingService, UsingFeesWithStorage		
	_initializeServiceWithStorage	Internal	✓	
	_getProviderStorage	Internal		
	_getFeeStorage	Internal		
	_setFeeStorage	Internal	✓	
	_getProviderFeeStorage	Internal		
UsingServiceProviderWithStorage	Implementation	UsingServiceProvider, UsingServiceWithStorage		
	_initializeServiceProviderWithStorage	Internal	✓	
	_getServicesStorage	Internal		
	_withdraw	Internal	✓	

UsingPermitWithStorage	Implementation	UsingPermit		
	_initializePermitWithStorage	Internal	✓	
	_getNoncesStorage	Internal		
	_getDomainSeparatorsStorage	Internal		
UsingSwapWithStorage	Implementation	UsingServiceProviderWithStorage		
	_initializeSwapWithStorage	Internal	✓	
	_addRouter	Internal	✓	
	_process	Internal	✓	
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		
IERC1822ProxiableUpgradable	Interface			
	proxiableUUID	External		-
IBeaconUpgradable	Interface			
	implementation	External		-
AddressUpgradable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
StorageSlotUpgradable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
UsingERC1967UpgradeUpgradable	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	✓	
BlockbustersSwap	Implementation	UsingSwapWithStorage, BlockbustersFlagsWithStorage, UsingInitializer, UsingUUPS		
	initialize	External	✓	initializer
	setup	External	✓	requires
	_process	Internal	✓	
	withdrawTokens	External	✓	requires
	_withdraw	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

This audit focuses on the business logic issues, the security concerns and the potential improvements. The owner has the authority to upgrade the contract via proxy.

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>