Cyberscope

Audit Report

BoundlessWorld

December 2022

Type	BEP20
Network	BSC
Address	0x13D67Fd10BDBe8301E978e4AdCBD2c0AD26F7549
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BLBToken
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Explorer	https://bscscan.com/token/0x13D67Fd10BDBe8301E978e4AdCBD2c0AD26F7549
Symbol	BLB
Decimals	18
Total Supply	100,000,000
Domain	boundlessworld.org

Audit Updates

Initial Audit	5th December 2022 https://github.com/cyberscope-io/audits/tree/main/2-blb/v1/audit.pdf
Corrected	12th December 2022

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	5af1771388b4fe634e0a566716e32c6d00a5372875099127b274d4cf8a94e9d2
@openzeppelin/contracts/access/AccessControlEnumerable.sol	47861db7fa8d98b58cef570e7c8fca6af6d9d82e3ec0f525c3ad035cbfbed195
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/access/IAccessControlEnumerable.sol	655ab8dc2a9617376734d04ca293e099cc24f8ce893997e68c29cfebc4a61d39
@openzeppelin/contracts/token/ERC20/ERC20.sol	a4ee82ad4893981800b6f57b26e8ee540fbff6d5133fb4baf0f719ede10e8c80
@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol	d070a08919d4a38aa08043c687d1fe1522098b212d2e185aedf2f37275b64087
@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol	3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef

@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	0344809a1044e11ece2401b4f7288f414ea41fa9d1dad24143c84b737c9fc02e
@openzeppelin/contracts/token/ERC20/extensions/ERC20Capped.sol	00d9364a71bfb7590fdeb7e097fe84159f4fc002c4f603b036c61f91e6368861
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/Counters.sol	2fdcb1343e5621385b62e57b5c7775607c272122b6f2dc77da8f84828aa40cd0
@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol	fc0e6c5d7184bd03b8deae6ca9a48a1eaaecf9f5e4703611aabfb63401e6d43f
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	4e45d53327d561848fbcf381262ec5c0ac91b2f1f06432210bf76db55279d945
@openzeppelin/contracts/utils/introspection/ERC16	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154

5.sol	
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
@openzeppelin/contracts/utils/Strings.sol	34127ad0054df5963b0fd694c1b313d17e9114a2f426b85526d6d976210298ab
@openzeppelin/contracts/utils/structs/EnumerableSet.sol	778d5305652c4eb562b12880cb6cf023d67df24844c15783a0b80fac2e715585
contracts/BEP20/Administration.sol	e6c2845214b05b750496da5bf96eedfbb26c0013ecaa90c6d9136b0729291737
contracts/BEP20/BLBToken.sol	f6f5b356e4619f7a6f563f751cf89ea09e47bfdb84f06eb163befc74c7be4bd5
contracts/BEP20/TransactionFee.sol	ba1f8455e21327106c051c983f36343007f4ef4b46e7794aa7b6b3331183dbb5

Introduction

The project consists of three roles, Minter, Fee Setter and Fee Free.

Roles

Minter

The Minter Role has the authority to:

- Mint tokens to an account or multiple accounts at once.

Fee Setter

The Fee Setter Role has the authority to:

- Alter fees to a fixed amount or to a percentage of the amount being transferred up to 10%.

Fee Free

The Fee Free Role has the authority to:

- Make a transaction without paying any fees.

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

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

ELFM - Exceeds Fees Limit

Criticality	critical
Location	contracts/BEP20/TransactionFee.sol#L37
Status	Unresolved

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setTransactionFee` function and setting the `_feeAmount` to a high value. As a result, if the `_feeAmount` is greater than the transfer amount (Reference to TSD - Total Supply Diversion) then the transaction will revert.

```
function setTransactionFee(
    uint256 _feeAmount,
    uint256 _feeFraction,
    address _feeReceiver
) public onlyRole(FEE_SETTER_ROLE) {
    require(
        _feeFraction == 0 || _feeAmount == 0,
        "TransactionFee: Cannot set feeAmount and feeFraction at the same time"
    );
    require(
        _feeFraction <= 10 ** 5,
        "TransactionFee: Up to 10% transactionFee can be set"
    );
    feeAmount = _feeAmount;
    feeFraction = _feeFraction;
    feeReceiver = _feeReceiver;

    emit SetTransactionFee(_feeAmount, _feeFraction, _feeReceiver);
}
```

Recommendation

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

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

MT - Mints Tokens

Criticality	critical
Location	contracts/BEP20/BLBToken.sol#L36,50
Status	Unresolved

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` or `mintBatch` function. As a result the contract tokens will be highly inflated.

```
function mint(address account, uint256 amount) public onlyRole(MINTER_ROLE) {
    _mint(account, amount);
}
...
function mintBatch(
    address[] calldata accounts,
    uint256 amount
) public onlyRole(MINTER_ROLE) {
    for(uint16 i; i < accounts.length; i++){
        _mint(accounts[i], amount);
    }
}
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	TSD	Total Supply Diversion	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L05	Unused State Variable	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

TSD - Total Supply Diversion

Criticality	critical
Location	contracts/BEP20/TransactionFee.sol#L93
Status	Unresolved

Description

The amount that is added to the total supply does not equal the amount that is added to the balances. As a result, the sum of balances is diverse from the total supply. The `_payTransactionFee` method calculates the fee and transfers it to the `feeAddress` but the fee amount is not being subtracted from the transfer amount.

```
function _beforeTokenTransfer(address from, address to, uint256 amount)
    internal
    virtual
    override
{
    if(from != address(0) && to != address(0)){
        _payTransactionFee(from, amount);
    }

    super._beforeTokenTransfer(from, to, amount);
}
```

Recommendation

The team is advised to subtract the fee from the transfer amount. The sum of balances should always be equal to the total supply.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/BEP20/TransactionFee.sol#L39,40,38
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.

```
_feeFraction
_feeReceiver
_feeAmount
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-conventions.

L05 - Unused State Variable

Criticality	minor / informative
Location	@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#L37
Status	Unresolved

Description

There are segments that contain unused state variables.

```
_PERMIT_TYPEHASH_DEPRECATED_SLOT
```

Recommendation

Remove unused state variables.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contracts/BEP20/BLBToken.sol#L54
Status	Unresolved

Description

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

```
i
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
AccessControlEnumerable	Implementation	IAccessControlEnumerable, AccessControl		
	supportsInterface	Public		-
	getRoleMember	Public		-
	getRoleMemberCount	Public		-
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-

	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
IAccessControlEnumerable	Interface	IAccessControl		
	getRoleMember	External		-
	getRoleMemberCount	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_pureTransfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20Permit	Implementation	ERC20, IERC20Per		

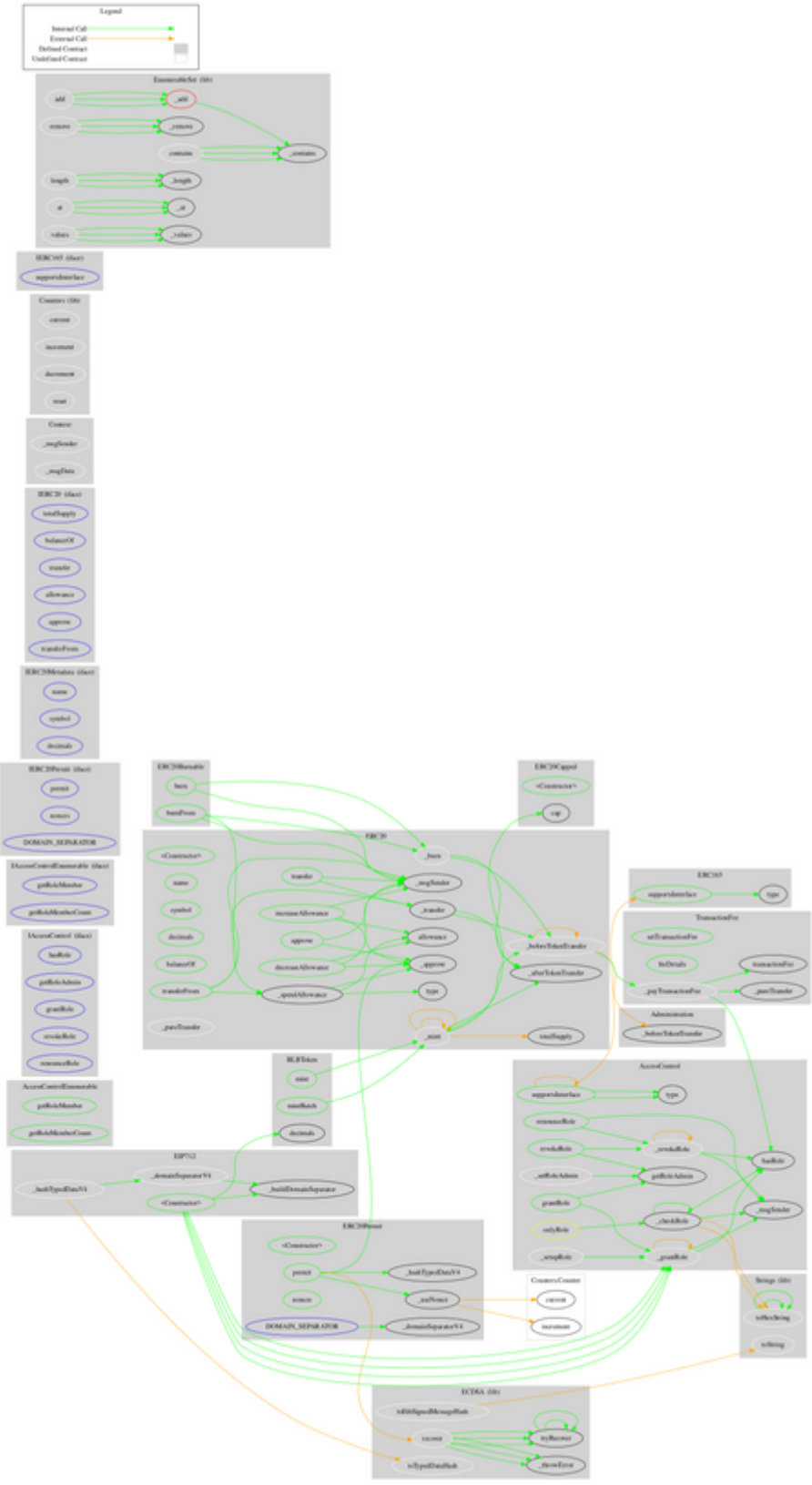
		mit, EIP712		
	<Constructor>	Public	✓	EIP712
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-
	_useNonce	Internal	✓	
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
ERC20Capped	Implementation	ERC20		
	<Constructor>	Public	✓	-
	cap	Public		-
	_mint	Internal	✓	
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
EIP712	Implementation			
	<Constructor>	Public	✓	-
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		
ECDSA	Library			
	_throwError	Private		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toTypedDataHash	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC165	Interface			
	supportsInterface	External		-
Strings	Library			

	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	_values	Private		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
Administration	Implementation	AccessControlEnumerable		

BLBToken	Implementation	ERC20, ERC20Capped, ERC20Burnable, ERC20Permit, TransactionFee		
	<Constructor>	Public	✓	ERC20 ERC20Capped ERC20Permit
	mint	Public	✓	onlyRole
	mintBatch	Public	✓	onlyRole
	_mint	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
TransactionFee	Implementation	ERC20, Administration		
	setTransactionFee	Public	✓	onlyRole
	feeDetails	Public		-
	transactionFee	Public		-
	_payTransactionFee	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	boundlessworld.org
Registry Domain ID	07e5646fcbfe4a58bf2077f2109998de-LROR
Creation Date	2022-04-25T09:54:42Z
Updated Date	2022-11-13T13:54:15Z
Registry Expiry Date	2032-04-25T09:54:42Z
Registrar WHOIS Server	http://whois.joker.com
Registrar URL	http://www.joker.com
Registrar	CSL Computer Service Langenbach GmbH d/b/a joker.com a German GmbH
Registrar IANA ID	113

The domain was created 8 months before the creation of the audit. It will expire in over 9 years.

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 manipulating fees and minting tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. 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

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io