Cyberscope

Audit Report

Sportsverse

August 2022

Type	BEP20
Network	BSC
Address	0x17B4057CA00E667F232D6ed367fb7AbA6400e18a
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
Contract Diagnostics 5
CR - Code Repetition 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
L07 - Missing Events Arithmetic 10
Description 10
Recommendation 10
L09 - Dead Code Elimination 11
Description 11
Recommendation 11
Contract Functions 12
Contract Flow 15

Domain Info 16

Summary 17

Disclaimer 18

About Cyberscope 19

Contract Review

Contract Name	Sportsverse
Compiler Version	v0.8.16+commit.07a7930e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x17B4057CA00E667F232D6ed367fb7AbA6400e18a
Symbol	SV
Decimals	18
Total Supply	1,000,000,000
Domain	

Source Files

Filename	**SHA256**
contract.sol	3c403c01ff90a31f81545ce2ed2efcdd0f7c3f8a388bb1eebf6aef63c2f6c5fd

Audit Updates

Initial Audit	3rd August 2022
Corrected phase 1	9th August 2022
Corrected phase 2	11th August 2022

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CR	Code Repetition
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination

CR - Code Repetition

Criticality	minor
Location	contract.sol#L630,645,665

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The `_transfer` method can be optimized. The following code segment is repetitive.

```
_balances[sender] -= amount;
_balances[recipient] += transferAmount;
emit Transfer(sender, recipient, transferAmount);
return;
```

This code segment is repetitive on `collectBuyFee`, `collectSellFee` methods.

```
uint256 _referralProgram = amount.mul(buyReferralProgramFee).div(10**(_feeDecimal + 2));
transferAmount=transferAmount.sub(_referralProgram);
_balances[ReferralAndFundsWallet] += _referralProgram;

emit Transfer(account, ReferralAndFundsWallet , _referralProgram);
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L291,296,302,306,310,317,460,468,494,501,513,521,532,550,572,591,706

Description

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

```
Burn
decreaseAllowance
increaseAllowance
transferFrom
approve
allowance
transfer
balanceOf
totalSupply
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L416,423

Description

Constant state variables should be declared constant to save gas.

```
totalTaxFee
_feeDecimal
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L336,706,724,731,736,398,399,412,416

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.

```
_feeDecimal
ReferralAndFundsWallet
_maxWallet
_maxAmount
_AutoBurnAndBuyBackFee
_marketingAndMaintananceFee
_ReferralProgramFee
_autoBurnAndBuyBackWallet
_referralAndFundsWallet
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L731,736

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.

```
sellmarketingAndMaintananceFee = _marketingAndMaintananceFee
buyReferralProgramFee = _ReferralProgramFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L160

Description

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

```
ceil
```

Recommendation

Remove unused functions.

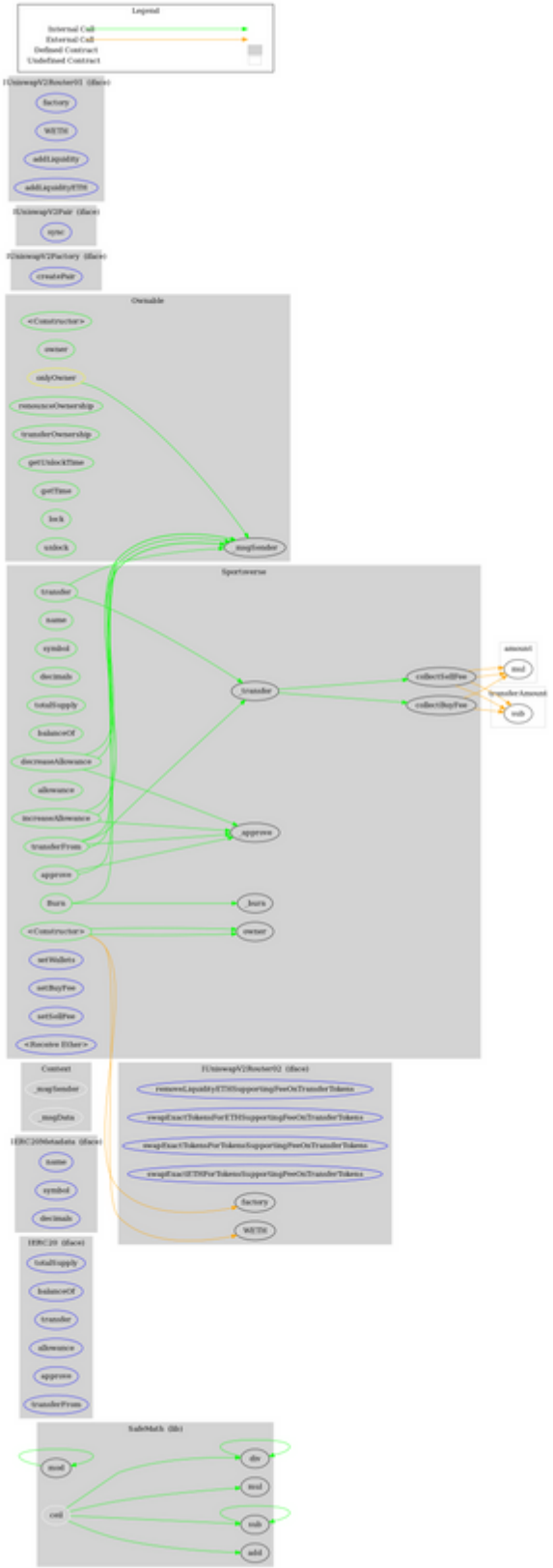
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
	ceil	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		

	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Pair	Interface			
	sync	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
IUniswapV2Router02	Interface	IUniswapV2 Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-

Sportsverse	Implementation	Ownable, 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	✓	
	collectBuyFee	Private	✓	
	collectSellFee	Private	✓	
	_approve	Internal	✓	
	Burn	Public	✓	-
	_burn	Internal	✓	
	setWallets	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	sportsverse.me
Registry Domain ID	D425500000339768159-AGRS
Creation Date	2022-07-28T20:47:56Z
Updated Date	2022-07-28T20:47:59Z
Registry Expiry Date	2023-07-28T20:47:56Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	www.namecheap.com
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 12 months before the creation of the audit.

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

Summary

Sportsverse is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 24% fees.

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

Sportsverse

August 2022

SHA256 9c8ca6ae92b5281bcf5239893fcde599f22635178210d51c5d6d084f7b139589

Audited by © cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 5
ST - Stop Transactions 6
Description 6
Recommendation 6
ELFM - Exceed Limit Fees Manipulation 7
Description 7
Recommendation 7
BT - Burn Tokens 8
Description 8
Recommendation 8
Contract Diagnostics 9
CR - Code Repetition 10
Description 10
Recommendation 10
L01 - Public Function could be Declared External 11
Description 11
Recommendation 11
L02 - State Variables could be Declared Constant 12
Description 12
Recommendation 12
L04 - Conformance to Solidity Naming Conventions 13
Description 13

Recommendation 13
L07 - Missing Events Arithmetic 14
Description 14
Recommendation 14
L09 - Dead Code Elimination 15
Description 15
Recommendation 15
Contract Functions 16
Contract Flow 19
Domain Info 20
Summary 21
Disclaimer 22
About Cyberscope 23

Contract Review

Contract Name	BallInu
Testing Deploy	https://testnet.bscscan.com/address/0xEc558b38F3B7d5aad0D6E51a545728C4D7678CFe#code
Symbol	$$SV
Decimals	18
Total Supply	1,000,000,000
Domain	https://sportsverse.me

Source Files

Filename	**SHA256**
contract.sol	9c8ca6ae92b5281bcf5239893fcde599f22635178210d51c5d6d084f7b139589

Audit Updates

Initial Audit	3rd August 2022
Corrected	



The burn method is publicly available without allowance. As a result, any user has the authority to arbitrarily burn the tokens from any address.

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L621

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `sellmarketingAndMaintananceFee` ,`sellAutoBurnAndBuyBackFee` to the maximum amount and turn the contract into a honeypot.

```
if(recipient == uniswapV2Pair){

    transferAmount = collectSellFee(sender,amount);

    _balances[sender] -= amount;
    _balances[recipient] += transferAmount;

    emit Transfer(sender, recipient, transferAmount);

    return;

}
```

Recommendation

The contract could embody a check for not allowing setting the `sellmarketingAndMaintananceFee` ,`sellAutoBurnAndBuyBackFee` to a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L727,731

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 setTaxFeePercent function with a high percentage value.

```
  function setBuyFee(uint256 _ReferralProgramFee) external onlyOwner {
      buyReferralProgramFee=_ReferralProgramFee;
  }

   function setSellFee(uint256 _marketingAndMaintananceFee, uint256
_AutoBurnAndBuyBackFee) external onlyOwner {
      sellmarketingAndMaintananceFee=_marketingAndMaintananceFee;
      sellAutoBurnAndBuyBackFee=_AutoBurnAndBuyBackFee;

  }
```

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.

BT - Burn Tokens

Criticality	critical
Location	contract.sol#L704

Description

Any user can burn any user's tokens. The user may take advantage of it by calling the `burn` function. As a result the targeted contract address will lose the corresponding tokens.

```
function Burn(address account, uint256 amount) public {
    _burn(_from, _amount);
}
```

Recommendation

An allowance mechanism should be introduced in order to prevent users from arbitrary burn tokens.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CO	Code Optimization
	CR	Code Repetition
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination

CR - Code Repetition

Criticality	minor
Location	contract.sol#L611,655,670

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The `_transfer` method can be optimized. The following code segment is repetitive.

```
_balances[sender] -= amount;
_balances[recipient] += transferAmount;
emit Transfer(sender, recipient, transferAmount);
return;
```

This code segment is repetitive on `collectBuyFee`, `collectSellFee` methods.

```
uint256 _referralProgram = amount.mul(buyReferralProgramFee).div(10**(_feeDecimal + 2));
transferAmount=transferAmount.sub(_referralProgram);
_balances[ReferralAndFundsWallet] += _referralProgram;

emit Transfer(account, ReferralAndFundsWallet , _referralProgram);
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L499,458,492,291,511,704,530,466,589,548,306,296,570,317,519,310,302

Description

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

```
getUnlockTime
lock
allowance
unlock
increaseAllowance
transferOwnership
getTime
transferFrom
decreaseAllowance
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L416

Description

Constant state variables should be declared constant to save gas.

```
_feeDecimal
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L704,720,412,727,399,398,416,336,731

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.

```
_referralAndFundsWallet
_marketingAndMaintananceFee
_AutoBurnAndBuyBackFee
WETH
_feeDecimal
_maxAmount
_autoBurnAndBuyBackWallet
_maxWallet
_ReferralProgramFee
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L731,727

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.

```
buyReferralProgramFee = _ReferralProgramFee
sellmarketingAndMaintananceFee = _marketingAndMaintananceFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L160

Description

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

```
ceil
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
	ceil	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		

Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Pair	Interface			
	sync	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-

BallInu	Implementation	Ownable, 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	✓	
	collectBuyFee	Private	✓	
	collectSellFee	Private	✓	
	_approve	Internal	✓	
	Burn	Public	✓	-
	_burn	Internal	✓	
	setWallets	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	sportsverse.me
Registry Domain ID	D425500000339768159-AGRS
Creation Date	2022-07-28T20:47:56Z
Updated Date	2022-07-28T20:47:59Z
Registry Expiry Date	2023-07-28T20:47:56Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	www.namecheap.com
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 12 months before the creation of the audit.

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

Summary

The contract is not functional since the burn method is open to any user without allowance.

Other than that, there are some functions that can be abused by the owner like stopping transactions, manipulating fees. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions.

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

Sportsverse

August 2022

SHA256 2ec56f93ab3021d0c076bf2943620c2888a65ce86681ea85d323d25cf6458644

Audited by © cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
Contract Diagnostics 5
CR - Code Repetition 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
L07 - Missing Events Arithmetic 10
Description 10
Recommendation 10
L09 - Dead Code Elimination 11
Description 11
Recommendation 11
Contract Functions 12
Contract Flow 15

Domain Info 16

Summary 17

Disclaimer 18

About Cyberscope 19

Contract Review

Contract Name	Sportsverse
Testing Deploy	https://testnet.bscscan.com/address/0x92BbFD5Ed5CE0e712CEEDB1F56a0D6ee18D3918C#code
Symbol	$$SV
Decimals	18
Total Supply	1,000,000,000
Domain	https://sportsverse.me

Source Files

Filename	**SHA256**
contract.sol	2ec56f93ab3021d0c076bf2943620c2888a65ce86681ea85d323d25cf6458644

Audit Updates

Initial Audit	3rd August 2022
Corrected	8th August 2022

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CR	Code Repetition
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination

CR - Code Repetition

Criticality	minor
Location	contract.sol#L626,641,661

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The `_transfer` method can be optimized. The following code segment is repetitive.

```
_balances[sender] -= amount;
_balances[recipient] += transferAmount;
emit Transfer(sender, recipient, transferAmount);
return;
```

This code segment is repetitive on `collectBuyFee`, `collectSellFee` methods.

```
uint256 _referralProgram = amount.mul(buyReferralProgramFee).div(10**(_feeDecimal + 2));
transferAmount=transferAmount.sub(_referralProgram);
_balances[ReferralAndFundsWallet] += _referralProgram;

emit Transfer(account, ReferralAndFundsWallet , _referralProgram);
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L499,458,492,291,511,704,530,466,589,548,306,296,570,317,519,310,302

Description

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

```
getUnlockTime
lock
allowance
unlock
increaseAllowance
transferOwnership
getTime
transferFrom
decreaseAllowance
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L416

Description

Constant state variables should be declared constant to save gas.

```
_feeDecimal
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L704,720,412,727,399,398,416,336,731

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.

```
_referralAndFundsWallet
_marketingAndMaintananceFee
_AutoBurnAndBuyBackFee
WETH
_feeDecimal
_maxAmount
_autoBurnAndBuyBackWallet
_maxWallet
_ReferralProgramFee
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L731,727

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.

```
buyReferralProgramFee = _ReferralProgramFee
sellmarketingAndMaintananceFee = _marketingAndMaintananceFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L160

Description

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

```
ceil
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
	ceil	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		

Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Pair	Interface			
	sync	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-

Sportsverse	Implementation	Ownable, 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	✓	
	collectBuyFee	Private	✓	
	collectSellFee	Private	✓	
	_approve	Internal	✓	
	Burn	Public	✓	-
	_burn	Internal	✓	
	setWallets	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	sportsverse.me
Registry Domain ID	D425500000339768159-AGRS
Creation Date	2022-07-28T20:47:56Z
Updated Date	2022-07-28T20:47:59Z
Registry Expiry Date	2023-07-28T20:47:56Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	www.namecheap.com
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 12 months before the creation of the audit.

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

Summary

Sportsverse is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 24% fees.

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