Cyberscope

Audit Report

Stargate

June 2022

Type	BEP20
Network	BSC
Address	0x46af007b7794DCa28076D026c92B1712212Fa559
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Audit Updates 3
Source Files 4
Contract Analysis 5
ELFM - Exceed Limit Fees Manipulation 6
Description 6
Recommendation 6
Contract Diagnostics 7
BLC - Business Logic Concern 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
L04 - Conformance to Solidity Naming Conventions 11
Description 11
Recommendation 11
L07 - Missing Events Arithmetic 12
Description 12
Recommendation 12
L13 - Divide before Multiply Operation 13
Description 13
Recommendation 13

Contract Functions 14
Contract Flow 18
Domain Info 19
Summary 20
Disclaimer 21
About Cyberscope 22

Contract Review

Contract Name	StarGateToken
Compiler Version	v0.8.5+commit.a4f2e591
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x46af007b7794DCa28076D026c92B1712212Fa559
Symbol	STAR
Decimals	18
Total Supply	99,999
Domain	stargateprotocol.io

Source Files

Filename	SHA256
Context.sol	6f7bd013f2a4000b92811ab05a3bc5ec25aefde638cb15ea68faac1546d21246
IERC20.sol	af5d31a257314cef3ec105377fd1c6f5db4a9b9038669de892cb7591432b825d
IPancakeRouterV2.sol	cb47b3c53a6ff3ca8edda2ff1ae47006272418f394dedeff71be1c4b3b5ca505
Ownable.sol	3007da1e0e54e6c24fdde11696fbf40ef26304b1f53d53a63697cfd844b972fa
StarGateToken.sol	2a87dc3aa34eb3c78e8615a9ad92dac8d315b2a8bd0c635f026107176c6c20ac

Audit Updates

Initial Audit	13th June 2022
Corrected	24th June 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

ELFM - Exceed Limit Fees Manipulation

Criticality	medium
Location	contract.sol#L210

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by setting addresses to the blacklist. The blacklisted addresses may be taxed up to 10 + 36 = **46%**

```
function calculateFeeRate(address sender, address recipient) private view
returns(uint256) {
	bool applyFees = _isFeeEnabled && !_addressesExcludedFromFees[sender] &&
!_addressesExcludedFromFees[recipient];
	if (applyFees) {
		if (isPancakeswapPair(recipient)) {
			if (_blacklistedAddresses[sender]) {
				return _totalFee + 36;
			} else {
				return _totalFee + _additionalSellFee;
			}
		}
		return _totalFee;
	}
	return 0;
}
```

Recommendation

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

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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	BLC	Business Logic Concern
	MC	Missing Check
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L13	Divide before Multiply Operation

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L55

Description

One of the contract fees is called burnFee. The burn fee gives the perspective that the taxed amount will be burned. On the contrary, **the contract sends the burned amount to the owner's wallet**.

```
// during the transfer
burnFee = amount * _blackHoleFee / 100;

// constructor
_blackHole = msg.sender;
```

Recommendation

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

MC - Missing Check

Criticality	medium
Location	contract.sol#L159

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 poolFee that is used as a divisor is the sum of _totalFee and _additionalSellFee. Both variables can be set to zero in the setFees() method. As a result, the sale transactions will revert and the contract will behave similar to a **honeypot**.

```
uint256 poolFee = _totalFee + _additionalSellFee;

uint256 tokensReservedForLiquidity = amount * _celestialLpFee / poolFee;
uint256 tokensReservedForReward = amount * _stargateFee / poolFee;
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contracts/Ownable.sol#L53,61 contracts/StarGateToken.sol#L72,77,85,90,96,259,299,309,313,318,323,328,338,343,347,355,359,375,379,383,388

Description

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

```
sendToBigBang
isBlacklistedWallet
removeBlacklistedWallet
setBlacklistedWallet
isFeeEnabled
setStarGateReserve
pancakeswapPairAddress
allowance
decimals
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/IPancakeRouterV2.sol#L7 contracts/StarGateToken.sol#L299,355,15,16,17,18,19,20,22,36,38,39,40,41

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.

```
_blackHole
_StarGateReserve
_bigBangWallet
_celestialLiquidityWallet
_pancakeswapV2Router
_totalTokens
_totalFee
_additionalSellFee
_blackHoleFee
...
```

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	contracts/StarGateToken.sol#L282,299,304,313

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.

```
_tokenSwapThreshold = threshold
_transactionLimit = limit
_tokenSwapThresholdAmount = _amount
_celestialLpFee = liquidityFee
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contracts/StarGateToken.sol#L72,155

Description

Performing divisions before multiplications may cause lose of prediction.

```
tokensToSwapForLiquidity = tokensReservedForLiquidity / 2
setTransactionLimit(_totalTokens / 100 * 1)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

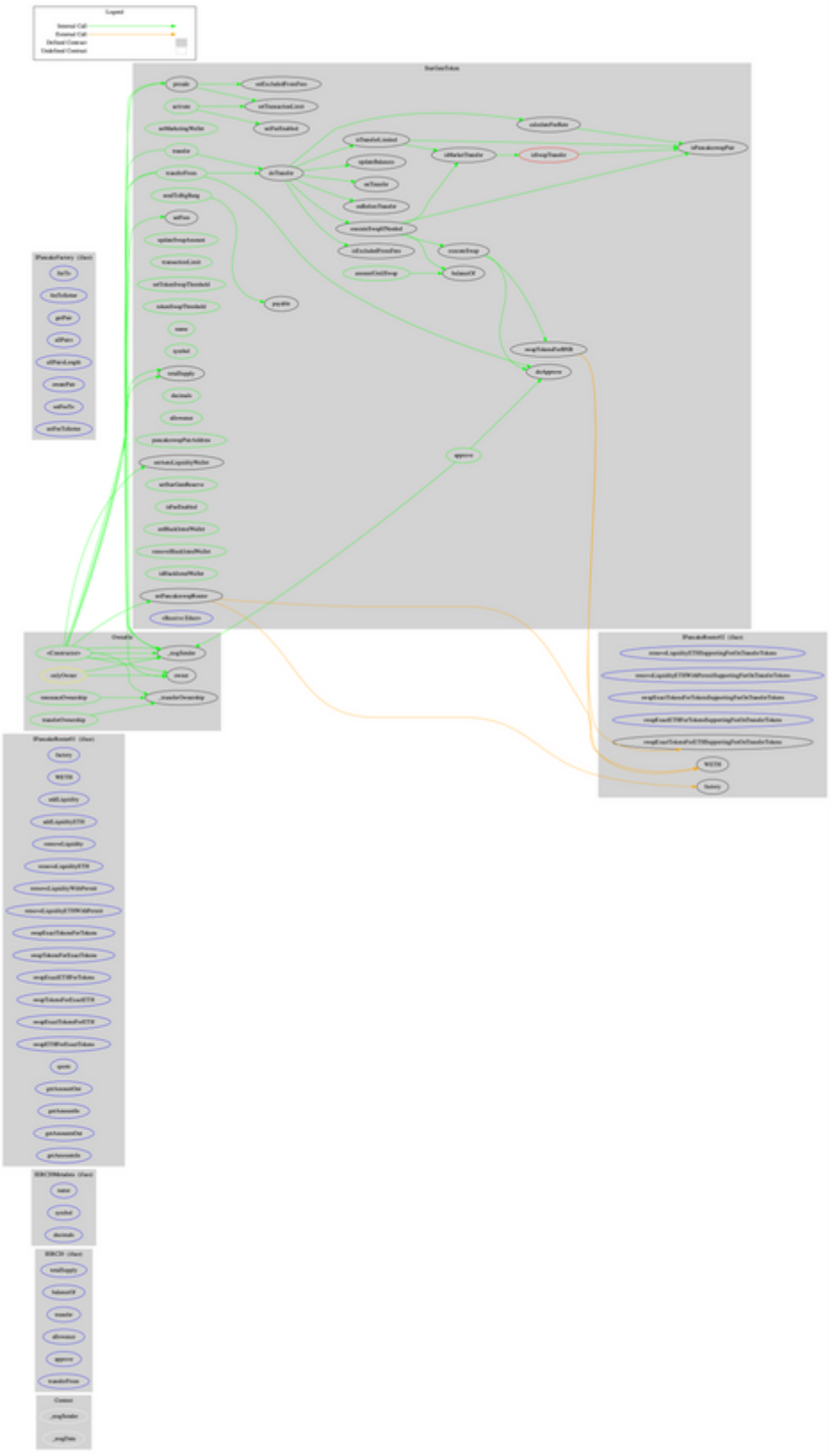
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IPancakeRouter01	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		-
IPancakeRouter02	Interface	IPancakeRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-

	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
StarGateToken	Implementation	Context, IERC20Metadata, Ownable		
	<Constructor>	Public	✓	-
	presale	Public	✓	onlyOwner
	activate	Public	✓	onlyOwner
	setMarketingWallet	Public	✓	onlyOwner
	balanceOf	Public		-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	approve	Public	✓	-
	doTransfer	Internal	✓	
	executeSwapIfNeeded	Private	✓	
	executeSwap	Private	✓	
	onBeforeTransfer	Internal	✓	
	onTransfer	Internal	✓	
	updateBalances	Private	✓	
	doApprove	Private	✓	
	calculateFeeRate	Private		
	swapTokensForBNB	Internal	✓	
	isTransferLimited	Private		
	isSwapTransfer	Private		
	isMarketTransfer	Internal		
	amountUntilSwap	Public		-
	setPancakeswapRouter	Public	✓	onlyOwner
	isPancakeswapPair	Internal		
	setFees	Public	✓	onlyOwner
	updateSwapAmount	Public	✓	onlyOwner
	setTransactionLimit	Public	✓	onlyOwner
	transactionLimit	Public		-
	setTokenSwapThreshold	Public	✓	onlyOwner

	tokenSwapThreshold	Public		-
	name	Public		-
	symbol	Public		-
	totalSupply	Public		-
	decimals	Public		-
	allowance	Public		-
	pancakeswapPairAddress	Public		-
	setAutoLiquidityWallet	Public	✓	onlyOwner
	setStarGateReserve	Public	✓	onlyOwner
	isFeeEnabled	Public		-
	setFeeEnabled	Public	✓	onlyOwner
	isExcludedFromFees	Public		-
	setExcludedFromFees	Public	✓	onlyOwner
	setBlacklistedWallet	Public	✓	onlyOwner
	removeBlacklistedWallet	Public	✓	onlyOwner
	isBlacklistedWallet	Public		onlyOwner
	sendToBigBang	Public	Payable	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	stargateprotocol.io
Registry Domain ID	d40b51074cdb46fbb3f22bc65c2d0994-DONUTS
Creation Date	2022-06-09T15:06:41Z
Updated Date	2022-06-09T16:04:21Z
Registry Expiry Date	2023-06-09T15:06:41Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created in 12 months before the creation of the audit.

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

Summary

The Smart Contract analysis reported one medium severity issue. The contract owner has the authority to manipulate the fees. Additionally, the contract contains some misleading implementations regarding the naming and the variables sanitization. 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

Stargate

June 2022

Github https://github.com/StargateBSC/STARGATE

Commit e69ed0280a760603a7aceb1bf8bee70177924672

Audited by © cyberscope

Table of Contents

Table of Contents 1
Audit Updates 3
Source Files 4
Contract Analysis 5
ELFM - Exceed Limit Fees Manipulation 6
Description 6
Recommendation 6
Contract Diagnostics 7
BLC - Business Logic Concern 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
L04 - Conformance to Solidity Naming Conventions 11
Description 11
Recommendation 11
L07 - Missing Events Arithmetic 12
Description 12
Recommendation 12
L13 - Divide before Multiply Operation 13
Description 13
Recommendation 13

Contract Functions 14
Contract Flow 18
Domain Info 19
Summary 20
Disclaimer 21
About Cyberscope 22

Contract Name	StarGateToken
Compiler Version	v0.8.5+commit.a4f2e591
Optimization	200 runs
Testing Deploy	https://testnet.bscscan.com/token/0x226D8bD767Ae0336F6c1bfec6Bdb37C5954F0CB0
Symbol	STAR
Decimals	18
Total Supply	100,000
Domain	stargateprotocol.io

Audit Updates

Initial Audit	13th June 2022
Corrected	

Source Files

Filename	SHA256
Context.sol	6f7bd013f2a4000b92811ab05a3bc5ec25aefde638cb15ea68faac1546d21246
IERC20.sol	af5d31a257314cef3ec105377fd1c6f5db4a9b9038669de892cb7591432b825d
IPancakeRouterV2.sol	cb47b3c53a6ff3ca8edda2ff1ae47006272418f394dedeff71be1c4b3b5ca505
Ownable.sol	3007da1e0e54e6c24fdde11696fbf40ef26304b1f53d53a63697cfd844b972fa
StarGateToken.sol	075e302530c9f7e9209179ee3a079e85fbaedc7d05cfd3382302d32cb0ef088d

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

ELFM - Exceed Limit Fees Manipulation

Criticality	medium
Location	contract.sol#L210

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by setting addresses to the blacklist. The blacklisted addresses may be taxed up to 10 + 36 = **46%**

```
function calculateFeeRate(address sender, address recipient) private view
returns(uint256) {
	bool applyFees = _isFeeEnabled && !_addressesExcludedFromFees[sender] &&
!_addressesExcludedFromFees[recipient];
	if (applyFees) {
		if (isPancakeswapPair(recipient)) {
			if (_blacklistedAddresses[sender]) {
				return _totalFee + 36;
			} else {
				return _totalFee + _additionalSellFee;
			}
		}
		return _totalFee;
	}
	return 0;
}
```

Recommendation

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

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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	BLC	Business Logic Concern
	MC	Missing Check
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L13	Divide before Multiply Operation

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L55

Description

One of the contract fees is called burnFee. The burn fee gives the perspective that the taxed amount will be burned. On the contrary, **the contract sends the burned amount to the owner's wallet**.

```
// during the transfer
burnFee = amount * _blackHoleFee / 100;

// constructor
_blackHole = msg.sender;
```

Recommendation

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

MC - Missing Check

Criticality	medium
Location	contract.sol#L159

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 poolFee that is used as a divisor is the sum of _totalFee and _additionalSellFee. Both variables can be set to zero in the setFees() method. As a result, the sale transactions will revert and the contract will behave similar to a **honeypot**.

```
uint256 poolFee = _totalFee + _additionalSellFee;

uint256 tokensReservedForLiquidity = amount * _celestialLpFee / poolFee;
uint256 tokensReservedForReward = amount * _stargateFee / poolFee;
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contracts/Ownable.sol#L53,61 contracts/StarGateToken.sol#L72,77,85,90,96,259,299,309,313,318,323,328,338,343,347,355,359,375,379,383,388

Description

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

```
sendToBigBang
isBlacklistedWallet
removeBlacklistedWallet
setBlacklistedWallet
isFeeEnabled
setStarGateReserve
pancakeswapPairAddress
allowance
decimals
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/IPancakeRouterV2.sol#L7 contracts/StarGateToken.sol#L299,355,15,16,17,18,19,20,22,36,38,39,40,41

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.

```
_blackHole
_StarGateReserve
_bigBangWallet
_celestialLiquidityWallet
_pancakeswapV2Router
_totalTokens
_totalFee
_additionalSellFee
_blackHoleFee
...
```

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	contracts/StarGateToken.sol#L282,299,304,313

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.

```
_tokenSwapThreshold = threshold
_transactionLimit = limit
_tokenSwapThresholdAmount = _amount
_celestialLpFee = liquidityFee
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contracts/StarGateToken.sol#L72,155

Description

Performing divisions before multiplications may cause lose of prediction.

```
tokensToSwapForLiquidity = tokensReservedForLiquidity / 2
setTransactionLimit(_totalTokens / 100 * 1)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

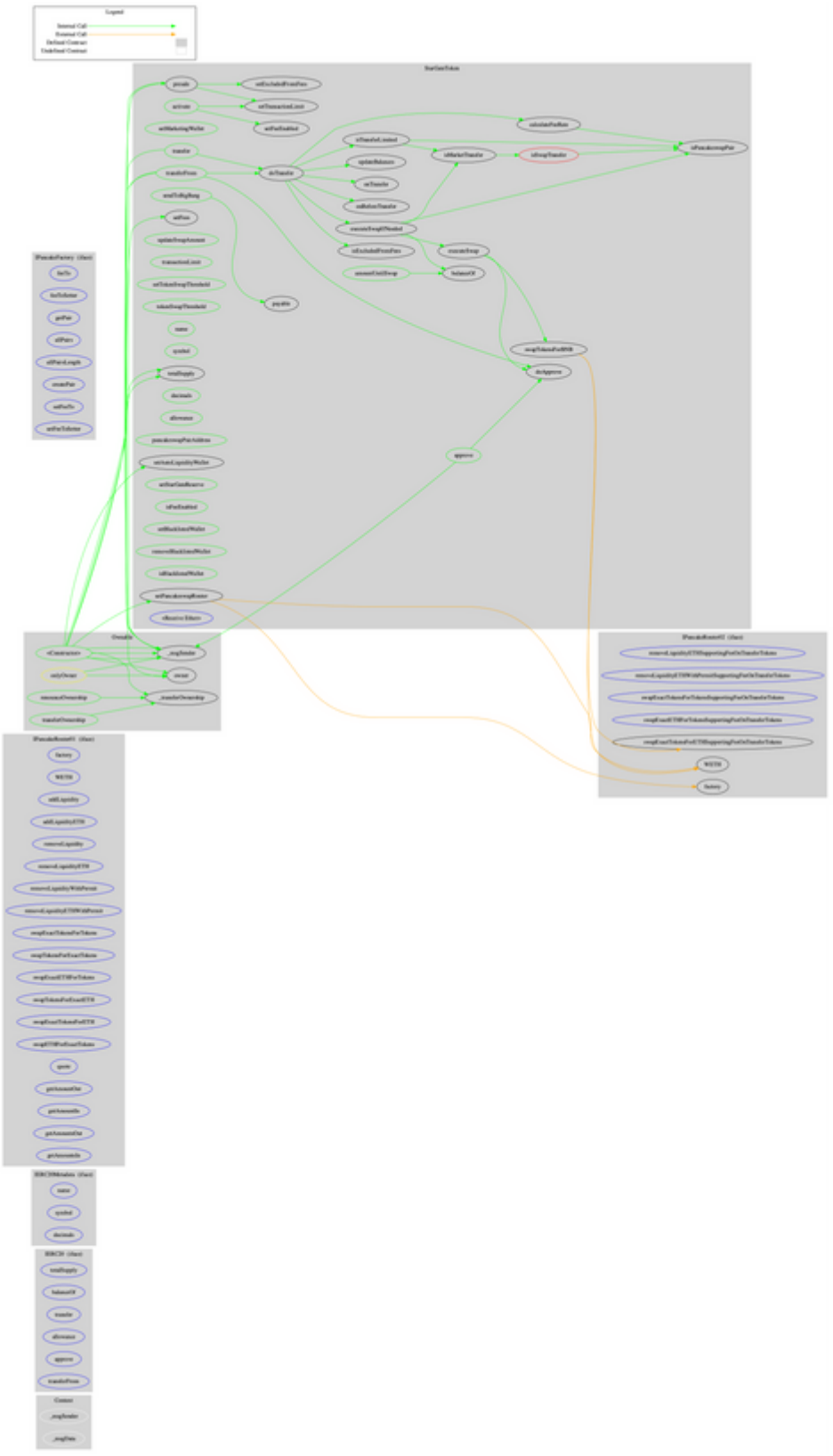
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IPancakeRouter01	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		-
IPancakeRouter02	Interface	IPancakeRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-

	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
StarGateToken	Implementation	Context, IERC20Metadata, Ownable		
	<Constructor>	Public	✓	-
	presale	Public	✓	onlyOwner
	activate	Public	✓	onlyOwner
	setMarketingWallet	Public	✓	onlyOwner
	balanceOf	Public		-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	approve	Public	✓	-
	doTransfer	Internal	✓	
	executeSwapIfNeeded	Private	✓	
	executeSwap	Private	✓	
	onBeforeTransfer	Internal	✓	
	onTransfer	Internal	✓	
	updateBalances	Private	✓	
	doApprove	Private	✓	
	calculateFeeRate	Private		
	swapTokensForBNB	Internal	✓	
	isTransferLimited	Private		
	isSwapTransfer	Private		
	isMarketTransfer	Internal		
	amountUntilSwap	Public		-
	setPancakeswapRouter	Public	✓	onlyOwner
	isPancakeswapPair	Internal		
	setFees	Public	✓	onlyOwner
	updateSwapAmount	Public	✓	onlyOwner
	setTransactionLimit	Public	✓	onlyOwner
	transactionLimit	Public		-
	setTokenSwapThreshold	Public	✓	onlyOwner

	tokenSwapThreshold	Public		-
	name	Public		-
	symbol	Public		-
	totalSupply	Public		-
	decimals	Public		-
	allowance	Public		-
	pancakeswapPairAddress	Public		-
	setAutoLiquidityWallet	Public	✓	onlyOwner
	setStarGateReserve	Public	✓	onlyOwner
	isFeeEnabled	Public		-
	setFeeEnabled	Public	✓	onlyOwner
	isExcludedFromFees	Public		-
	setExcludedFromFees	Public	✓	onlyOwner
	setBlacklistedWallet	Public	✓	onlyOwner
	removeBlacklistedWallet	Public	✓	onlyOwner
	isBlacklistedWallet	Public		onlyOwner
	sendToBigBang	Public	Payable	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	stargateprotocol.io
Registry Domain ID	d40b51074cdb46fbb3f22bc65c2d0994-DONUTS
Creation Date	2022-06-09T15:06:41Z
Updated Date	2022-06-09T16:04:21Z
Registry Expiry Date	2023-06-09T15:06:41Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created in 12 months before the creation of the audit.

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

Summary

The Smart Contract analysis reported one medium severity issue. The contract owner has the authority to manipulate the fees. Additionally, the contract contains some misleading implementations regarding the naming and the variables sanitization. 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