Cyberscope

Audit Report

Spooky Halloween Floki

September 2022

Type	BEP20
Network	BSC
Address	0x9f59E79127fAD40bBC26c1eC6578d345AF40bD1f
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	SPOOKYHALLOWEENFLOKI
Compiler Version	v0.8.12+commit.f00d7308
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x9f59E79127fAD40bBC26c1eC6578d345AF40bD1f
Symbol	SHF
Decimals	9
Total Supply	10,000,000,000
Domain	http://spookyhalloweenfloki.com

Source Files

Filename	**SHA256**
contract.sol	22e45ba46990ee3154202fbd0bef2fac90dc45828fb46ab5b9ac14604cd171e0

Audit Updates

Initial Audit	29th September 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

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

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L351
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 `multiTransfer` and `clearStuckToken` function.

```
function clearStuckToken(address tokenAddress, uint256 tokens) external onlyOwner
returns (bool success) {
        if(tokens == 0){
            tokens = BEP20(tokenAddress).balanceOf(address(this));
        }
        emit clearToken(tokenAddress, tokens);
        return BEP20(tokenAddress).transfer(msg.sender, tokens);
}
```

OTUT - Transfers User's Tokens

Criticality	critical
Location	contract.sol#L521
Status	Unresolved

Description

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

```
function multiTransfer(address from, address[] calldata addresses, uint256[] calldata tokens) external authorized {
    if(msg.sender != from){
        require(launchMode,"Cannot execute this after launch is done");
    }
    require(addresses.length < 501,"GAS Error: max limit is 500 addresses");
    require(addresses.length == tokens.length,"Mismatch between address and token count");
    uint256 SCCC = 0;
    for(uint i=0; i < addresses.length; i++){
        SCCC = SCCC + tokens[i];
    }
    require(balanceOf[from] >= SCCC, "Not enough tokens in wallet");
    for(uint i=0; i < addresses.length; i++){
        _basicTransfer(from,addresses[i],tokens[i]);
    }
}
```

Recommendation

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

The issue can be resolved if the contract owner executes the `tradingStatus_launchmode(false)` function after the trading opens.

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L343
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 clearStuckBalance methods.

```
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
        require(amountPercentage < 101, "Max 100%");
        uint256 amountBNB = address(this).balance;
        uint256 amountToClear = ( amountBNB * amountPercentage ) / 100;
        payable(msg.sender).transfer(amountToClear);
        emit BalanceClear(amountToClear);
    }
```

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.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	ZD	Zero Division	Unresolved
	BM	Buyback Misleading	Unresolved
	STC	Succeeded Transfer Check	Unresolved
	FSA	Fixed Swap Address	Unresolved
	L01	Public Function could be Declared External	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L07	Missing Events Arithmetic	Unresolved
	L13	Divide before Multiply Operation	Unresolved

ZD - Zero Division

Criticality	critical
Location	contract.sol#L376
Status	Unresolved

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert.

The variables totalETHFee can be set to zero if the totalFee is equal to burnFee.

```
function swapBack() internal swapping {

    uint256 totalETHFee = totalFee - burnFee;

    uint256 amountToLiquify = (swapThreshold * liquidityFee)/(totalETHFee * 2);
```

Recommendation

The contract could embody a check to avoid the totalFee and burnFee equality.

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

BM - Buyback Misleading

Criticality	minor / informative
Location	contract.sol#L410
Status	Unresolved

Description

According to the buyback definition, a buyback is when a token purchases its own shares in the market. The address 0x14940169E2Db1595CDD3CACd30DECC5bbB4d9f19 does not beloge to the contract address 0x9f59e79127fad40bbc26c1ec6578d345af40bd1f. As a result the buyback fees are liquidated to buyback the 0x14940169E2Db1595CDD3CACd30DECC5bbB4d9f19 instead of the 0x14940169E2Db1595CDD3CACd30DECC5bbB4d9f19.

```
buybackPath[0] = router.WETH();
buybackPath[1] = address(0x14940169E2Db1595CDD3CACd30DECC5bbB4d9f19);
//
router.swapExactETHForTokensSupportingFeeOnTransferTokens{value:
amountBNBBuyback}(
    0,
    buybackPath,
    DEAD,
    block.timestamp
);
```

Recommendation

The contract should buyback its own token shares.

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L358
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.

```
return BEP20(tokenAddress).transfer(msg.sender, tokens);
```

Recommendation

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

FSA - Fixed Swap Address

Criticality	minor / informative
Location	contract.sol#L206
Status	Unresolved

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
constructor () Auth(msg.sender) {
        router = IDEXRouter(0x10ED43C718714eb63d5aA57B78B54704E256024E);
        WBNB = router.WETH();

        pair = IDEXFactory(router.factory()).createPair(WBNB, address(this));
        _allowances[address(this)][address(router)] = type(uint256).max;
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L512
Status	Unresolved

Description

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

```
getCirculatingSupply
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L430,541,473,160,258,543,120,465,552,554,150,548,485,365,111,550,508,158,164,499,542,264,161,438,177,357,546,446,553,454,551
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.

```
manage_FeeExempt
Wallet_feeExempt
setFees_base1000
_maxTxAmount
setMaxWalletPercent_base1000
Wallet_holdingExempt
WETH
_trans
config_TradingStatus
...
```

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 / informative
Location	contract.sol#L473,465
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.

```
liquidityFee = _liquidityFee
sellMultiplier = _sell
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L302
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
feeAmount = amount.mul(totalFee).mul(multiplier).div(feeDenominator * 100)
```

Recommendation

The multiplications should be prior to the divisions.

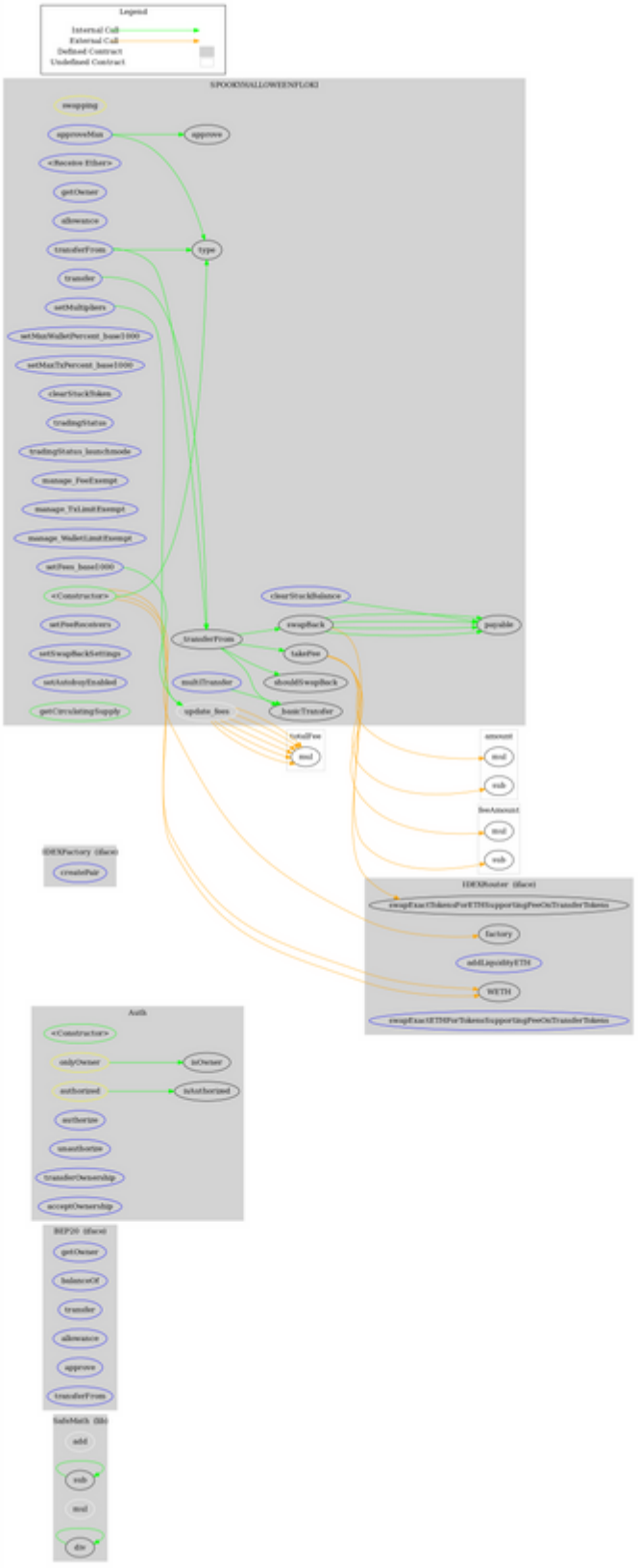
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
BEP20	Interface			
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-
	authorize	External	✓	onlyOwner
	unauthorize	External	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	External	✓	onlyOwner
	acceptOwnership	External	✓	-
IDEXFactory	Interface			
	createPair	External	✓	-

IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
SPOOKYHALLOWEENFLOKI	Implementation	BEP20, Auth		
	<Constructor>	Public	✓	Auth
	<Receive Ether>	External	Payable	-
	getOwner	External		-
	allowance	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	setMaxWalletPercent_base1000	External	✓	onlyOwner
	setMaxTxPercent_base1000	External	✓	onlyOwner
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	clearStuckBalance	External	✓	onlyOwner
	clearStuckToken	External	✓	onlyOwner
	tradingStatus	External	✓	onlyOwner
	tradingStatus_launchmode	External	✓	onlyOwner
	swapBack	Internal	✓	swapping
	manage_FeeExempt	External	✓	authorized
	manage_TxLimitExempt	External	✓	authorized
	manage_WalletLimitExempt	External	✓	authorized
	update_fees	Internal	✓	
	setMultipliers	External	✓	authorized

	setFees_base1000	External	✓	onlyOwner
	setFeeReceivers	External	✓	onlyOwner
	setSwapBackSettings	External	✓	onlyOwner
	setAutobuyEnabled	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	multiTransfer	External	✓	authorized

Contract Flow



Domain Info

Domain Name	spookyhalloweenfloki.com
Registry Domain ID	2723208257_DOMAIN_COM-VRSN
Creation Date	2022-09-04T21:06:35Z
Updated Date	2022-09-04T21:06:36Z
Registry Expiry Date	2023-09-04T21:06:35Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	http://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain was created 24 days before the creation of the audit. It will expire in 11 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 transferring tokens from any address to the team’s wallet. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 20% fees on sell transactions, a limit of max 15% fee on buy and a limit of max 10% on transfer transactions.

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