RugFreeCoins Audit

SunShield Token

Smart Contract Security Audit

October 04, 2021

Contents

Audit details

Audited project

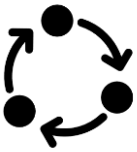
SunShield Token

Contract Address

https://bscscan.com/token/0xc8412dced6ba127225edc4c17a096ee35f99d296

Client contact

SunShield Team



Blockchain

Binance smart chain

Project website

https://www.sunshield.finance/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Rugfreecoins hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

Rugfreecoins was commissioned by SunShield Token to perform an audit of the smart contract.

https://bscscan.com/token/0xc8412dced6ba127225edc4c17a096ee35f99d296

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

About the project

SunShield Token is a token built on the Binance Smart Chain. Each transaction, purchase incurs a 17% fee, and sales incur a 19% fee.

Features

- The **automatic BNB reward of 10%** is what SunShield's entire marketing strategy is based around: that BNB rewards will be distributed among every holder proportional to how many tokens each individual holds.

- The liquidity fee of 1%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.

- SunShield will be exchanging 3% of the tax for Bnb buys back from the supply every minute and burn all tokens bought automatically. An additional 2% will be charged when selling and allocated for buyback and burn.

- The sustainability fee of 2% marketing is what allows SunShield to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, SunShield will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

- The fee of 1% will be deducted and swapped into BNBs and sent to the Moonshield BNB pool.

Tokenomics

17% fee when buying

- 10% of trade goes to holders' pockets in BNB.
- 1% of trade goes to the liquidity pool.
- 3% of trade goes to marketing.
- 3% of trade goes to buyback & burn.
- 1% of trade goes to the Moonshield BNB pool.

19% fee when selling

- 10% of trade goes to holders pockets in BNB.
- 1% of trade goes to the liquidity pool.
- 3% of trade goes to marketing.
- 5% of trade goes to buyback & burn.
- 1% of trade goes to the Moonshield BNB pool.

Roadmap

2021/Q4

- ✓ Rug free coins audit
- ✓ Establish tg group
- ✓ Luanch social media platforms

Dx pre sale

Liquidity locked in dx locker

Pcs launch

Sunshield dapp

1 mil market cap

Large marketing campaign

Apply for coin market cap

Apply for coin gecko

2022/Q1

Earth shield development

Exhange listings

Target market and the concept

Target market

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income in BNB by holding tokens.
- Anyone who's interested in trading tokens.
- All BNB investors and fans out there.
- Anyone who's interested in taking part in the future plans of SunShield project.
- Anyone who's interested in making financial transactions with any other party using SunShield tokens or BNB as the currency.

Core concept

The BNB reward system

10% of each transaction gets converted to BNB, and is split amongst all holders. The rewards are sent to holders that have at least 100 billion SunShield tokens, holders will be eligible to receive tokens every 12 hours and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The liquidity fee of 1%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The **fee of 2% marketing** is what allows SunShield to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, SunShield will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The buyback and burn mechanism collect 3% tax on when buying and 5% when selling, which is stored inside the contract. Whenever a buy or sell occurs, a fraction of the buyback amount is used to automatically purchase tokens from the liquidity pool. Those tokens are immediately burned after purchase, which keeps the token price stable.

BNB pool for Moonshield

The reserved pool is another BNB pool that starts to fill up when the main BNB pool is full (5000BNB). It improves consistency in the amount of rewards received by all users in times where the volume is low.

When the volume is low, the main pool would normally start to dry out. With the reserved pool, this is not the case since it will be used to refill it.

Anti-dump

The Anti-Dump feature prevents users from selling more than a pre-set amount of tokens within a single transaction.

This prevents users with a very high amount of tokens from affecting the price significantly and further stabilizes the price.

The transaction selling limit is set to 2 Trillion. Any transaction selling an amount higher than the limit will be rejected by the network. This may be changed accordingly as the price of the token fluctuates in the market. Any potential change will be communicated to the community beforehand.

Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	8/10
5	System quality	8/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	7/10
Total Points		**7.88/10**

Contract details

Token contract details for 04th October 2021

Contract name	SunShield
Contract address	0xc8412dced6ba127225edc4c17a096ee35f99d296
Token supply	2,000,000,000,000,000
Token ticker	SSHLD
Decimals	9
Token holders	2
Transaction count	2
Marketing wallet address	0xb0f8dd8469dcb6e1eb50a818fe206f0d1299936e
Contract deployer address	0x7Bf7b57346de12a54eBc799444820D94796b9508
Contract's current owner address	0x7bf7b57346de12a54ebc799444820d94796b9508

Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	pass
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass

Contract description table

Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.

Contract	Type	Bases		
└	Function Name	Visibility	Mutability	Modifiers
IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IERC20Metadata	**Interface**	**IERC20**		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
Context	Implementation			

└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
IPancakeRouter01	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗

L	swapExactETHForTokens	External ❗	💵	NO❗
L	swapTokensForExactETH	External ❗	🛑	NO❗
L	swapExactTokensForETH	External ❗	🛑	NO❗
L	swapETHForExactTokens	External ❗	💵	NO❗
L	quote	External ❗		NO❗
L	getAmountOut	External ❗		NO❗
L	getAmountIn	External ❗		NO❗
L	getAmountsOut	External ❗		NO❗
L	getAmountsIn	External ❗		NO❗
IPancakeRouter02	**Interface**	**IPancakeRouter01**		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingF	External ❗	🛑	NO❗

	eeOnTransferTokens			
IPancakeFactory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
ReentrancyGuard	**Implementation**			
└		Public ❗	🛑	NO❗
SunShieldBase	**Implementation**	**Context, IERC20Metadata, Ownable, ReentrancyGuard**		
└		Public ❗	🛑	NO❗
└	activate	Public ❗	🛑	onlyOwner

└	onActivated	Internal 🔒	🛑	
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	approve	Public ❗	🛑	NO❗
└	doTransfer	Internal 🔒	🛑	
└	onBeforeTransfer	Internal 🔒	🛑	
└	onTransfer	Internal 🔒	🛑	
└	updateBalances	Private 🤞	🛑	
└	doApprove	Private 🤞	🛑	
└	calculateFeeRate	Private 🤞		
└	executeSwapIfNeeded	Private 🤞	🛑	
└	executeSwap	Private 🤞	🛑	
└	swapTokensForBNB	Internal 🔒	🛑	
└	swapBNBForTokens	Internal 🔒	🛑	
└	isTransferLimited	Private 🤞		
└	isSwapTransfer	Private 🤞		
└	isMarketTransfer	Internal 🔒		

└	amountUntilSwap	Public ❗		NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	setPancakeSwapRouter	Public ❗	🛑	onlyOwner
└	onPancakeSwapRouterUpdated	Internal 🔒	🛑	
└	isPancakeSwapPair	Internal 🔒		
└	setFees	Public ❗	🛑	onlyOwner
└	setTransactionLimit	Public ❗	🛑	onlyOwner
└	transactionLimit	Public ❗		NO❗
└	setTokenSwapThreshold	Public ❗	🛑	onlyOwner
└	tokenSwapThreshold	Public ❗		NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	allowance	Public ❗		NO❗
└	pancakeSwapRouterAddress	Public ❗		NO❗
└	pancakeSwapPairAddress	Public ❗		NO❗

└	autoLiquidityWallet	Public ❗		NO❗
└	setAutoLiquidityWallet	Public ❗	🛑	onlyOwner
└	devmarketingWallet	Public ❗		NO❗
└	setMarketingWallet	Public ❗	🛑	onlyOwner
└	totalFeesPooled	Public ❗		NO❗
└	totalBNBLiquidityAddedFromFees	Public ❗		NO❗
└	isSwapEnabled	Public ❗		NO❗
└	setSwapEnabled	Public ❗	🛑	onlyOwner
└	isFeeEnabled	Public ❗		NO❗
└	setFeeEnabled	Public ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	setExcludedFromFees	Public ❗	🛑	onlyOwner
└	activateBuying	Public ❗	🛑	onlyOwner
└		External ❗	💵	NO❗
SunShield	**Implementation**	**SunShieldBase**		
└		Public ❗	🛑	SunShield Base
└	onActivated	Internal 🔒	🛑	

L	onBeforeTransfer	Internal 🔒	🛑	
L	onTransfer	Internal 🔒	🛑	
L	processGradualBurn	Private 🤞	🛑	
L	updateAutoClaimQueue	Private 🤞	🛑	
L	claimReward	External ❗	🛑	isHuman nonReentrant
L	claimReward	Public ❗	🛑	NO❗
L	doClaimReward	Private 🤞	🛑	
L	claimBNB	Private 🤞	🛑	
L	claimSSHLD	Private 🤞	🛑	
L	processRewardClaimQueue	Public ❗	🛑	NO❗
L	processRewardClaimQueueAndRefundGas	External ❗	🛑	NO❗
L	isRewardReady	Public ❗		NO❗
L	isIncludedInRewards	Public ❗		NO❗
L	calculateRewardCycleExtension	Public ❗		NO❗
L	calculateClaimRewards	Public ❗		NO❗
L	calculateBNBReward	Public ❗		NO❗
L	onPancakeSwapRouterUpdated	Internal 🔒	🛑	

└	isMarketTransfer	Internal 🔒		
└	isBurnTransfer	Private 🔐		
└	shouldBurn	Public ❗		NO❗
└	buyAndBurn	External ❗	🛑	onlyOwner
└	doBuyAndBurn	Private 🔐	🛑	
└	isContract	Public ❗		NO❗
└	totalAmountOfTokensHeld	Public ❗		NO❗
└	bnbRewardClaimed	Public ❗		NO❗
└	bnbRewardClaimedAsSSHLD	Public ❗		NO❗
└	totalBNBClaimed	Public ❗		NO❗
└	totalBNBClaimedAsSSHLD	Public ❗		NO❗
└	rewardCyclePeriod	Public ❗		NO❗
└	setRewardCyclePeriod	Public ❗	🛑	onlyOwner
└	setRewardCycleExtensionThreshold	Public ❗	🛑	onlyOwner
└	nextAvailableClaimDate	Public ❗		NO❗
└	maxClaimAllowed	Public ❗		NO❗
└	setMaxClaimAllowed	Public ❗	🛑	onlyOwner
└	minRewardBalance	Public ❗		NO❗

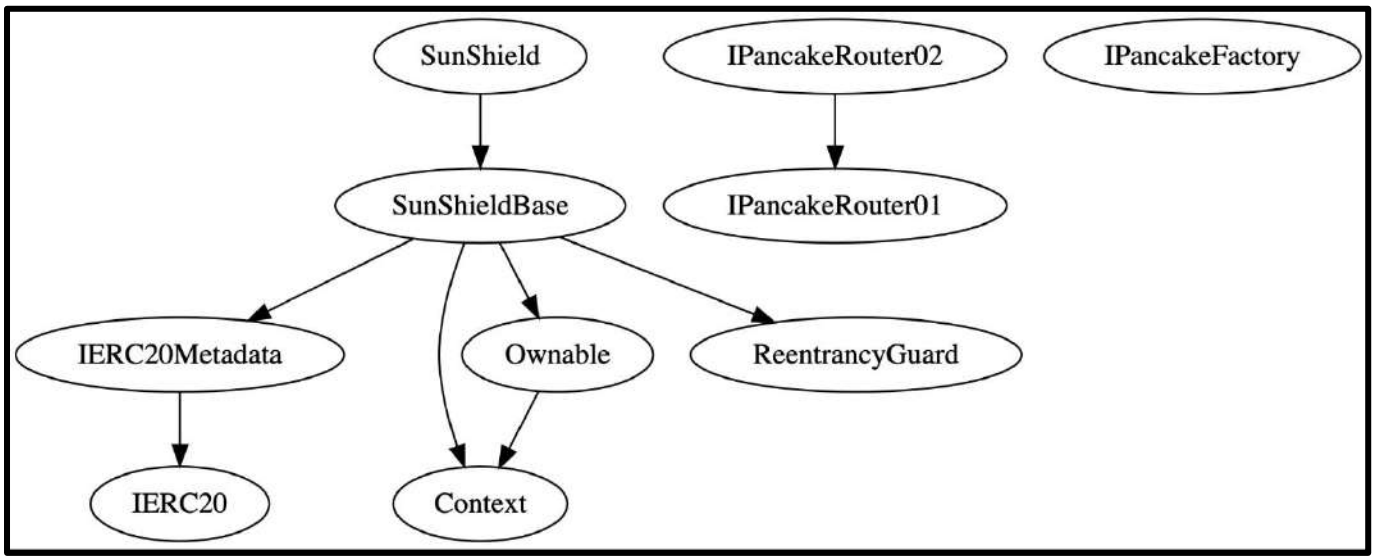
L	setMinRewardBalance	Public ❗	🛑	onlyOwner
L	maxGasForAutoClaim	Public ❗		NO❗
L	setMaxGasForAutoClaim	Public ❗	🛑	onlyOwner
L	isAutoClaimEnabled	Public ❗		NO❗
L	setAutoClaimEnabled	Public ❗	🛑	onlyOwner
L	isExcludedFromRewards	Public ❗		NO❗
L	setExcludedFromRewards	Public ❗	🛑	onlyOwner
L	globalRewardDampeningPercentage	Public ❗		NO❗
L	setGlobalRewardDampeningPercentage	Public ❗	🛑	onlyOwner
L	approveClaim	Public ❗	🛑	NO❗
L	isClaimApproved	Public ❗		NO❗
L	isRewardAsTokensEnabled	Public ❗		NO❗
L	setRewardAsTokensEnabled	Public ❗	🛑	onlyOwner
L	gradualBurnMagnitude	Public ❗		NO❗
L	setGradualBurnMagnitude	Public ❗	🛑	onlyOwner
L	gradualBurnTimespan	Public ❗		NO❗
L	setGradualBurnTimespan	Public ❗	🛑	onlyOwner
L	claimRewardAsTokensPercentage	Public ❗		NO❗

L	setClaimRewardAsTokensPercentage	Public ❗	🛑	NO❗
L	mainBnbPoolSize	Public ❗		NO❗
L	setMainBnbPoolSize	Public ❗	🛑	onlyOwner
L	isInRewardClaimQueue	Public ❗		NO❗
L	reimburseAfterSSHLDClaimFailure	Public ❗		NO❗
L	setReimburseAfterSSHLDClaimFailure	Public ❗	🛑	onlyOwner
L	lastBurnDate	Public ❗		NO❗
L	rewardClaimQueueLength	Public ❗		NO❗
L	rewardClaimQueueIndex	Public ❗		NO❗
L	isWhitelistedExternalProcessor	Public ❗		NO❗
L	setWhitelistedExternalProcessor	Public ❗	🛑	onlyOwner
L	setSendWeiGasLimit	Public ❗	🛑	onlyOwner
L	setExcludeNonHumansFromRewards	Public ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	**Function can modify state**
💶	**Function is payable**

Inheritance Hierarchy



Security issue checking status

- **High severity issues**
  - **No high severity issues found.**
- **Medium severity issues**
  - **No medium severity issues found.**
- **Low severity issues**
  - **No low severity issues found**

Owner privileges

- The owner can buy and burn tokens manually (max up to 1%).

```
// Up to 1% manual buyback & burn
ftrace | funcSig
function buyAndBurn(uint256 bnbAmount⬆) external onlyOwner {
    require(
        bnbAmount⬆ <= address(this).balance / 100,
        "Manual burn amount is too high!"
    );
    require(bnbAmount⬆ > 0, "Amount must be greater than zero");

    doBuyAndBurn(bnbAmount⬆);
}
```

- The owner can change the reward cycle period.

```
ftrace | funcSig
function setRewardCyclePeriod(uint256 period⬆) public onlyOwner {
    require(
        period⬆ >= 3600 && period⬆ <= 86400,
        "RewardCycle must be updated to between 1 and 24 hours"
    );
    _rewardCyclePeriod = period⬆;
}
```

- The owner can change max manual claim allowed limit.

```
ftrace | funcSig
function setMaxClaimAllowed(uint256 value⬆) public onlyOwner {
    require(value⬆ > 0, "Value must be greater than zero");
    _maxClaimAllowed = value⬆;
}
```

- The owner can change the minimum balance to get rewards.

```
ftrace | funcSig
function setMinRewardBalance(uint256 balance⬆) public onlyOwner {
    _minRewardBalance = balance⬆;
}
```

❖ The owner can change the max gas limit to auto claim.

```
ftrace | funcSig
function setMaxGasForAutoClaim(uint256 gas⬆) public onlyOwner {
    _maxGasForAutoClaim = gas⬆;
}
```

❖ The owner can enable and disable auto claim.

```
ftrace | funcSig
function setAutoClaimEnabled(bool isEnabled⬆) public onlyOwner {
    _autoClaimEnabled = isEnabled⬆;
}
```

❖ The owner can change the main BNB pool size (max 10 BNB).

```
ftrace | funcSig
function setMainBnbPoolSize(uint256 size⬆) public onlyOwner {
    require(size⬆ >= 10 ether, "Size is too small");
    _mainBnbPoolSize = size⬆;
}
```

❖ The owner can change wei gas limit.

```
ftrace | funcSig
function setSendWeiGasLimit(uint256 amount⬆) public onlyOwner {
    _sendWeiGasLimit = amount⬆;
}
```

Audit conclusion

While conducting the audit of the SunShield Token smart contract, it was observed that there is nothing alarming with the code..