RugFreeCoins Audit

One Whale Audit

Smart Contract Security Audit

November 12, 2021

Contents

Audit details

Audited project

One Whale Token

Contract Address

0x0196D4D37990aB2e1f7BC8607C7b4198A360c48f

Client contact

One Whale Team

Blockchain

Binance smart chain

Project website

https://www.onewhale.net/

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 One Whale Team to perform an audit of the smart contract.

https://bscscan.com/token/0x0196D4D37990aB2e1f7BC8607C7b4198A360c48f

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

One Whale is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, which in turn, powers a buyback engine built into the contract, as well as a manual buyback that translates to unlimited $BUSD rewards to POW Token holders. Each transaction, purchase, and sale incur a 15% fee.

Features

- The **BUSD rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of 8% **when buying and selling**.

- The **sustainability fee of 3% for marketing and whale investment fund when buying and selling** is what allows One Whale to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, One Whale will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

- One Whale has a controlled buyback and investment fund wallet that will be used to save from massive dips in order to keep the token market price stable, **2% fee when buying and selling** will be allocated for this.

- The additional component included under the sustainability section is a **liquidity fee of 2% from buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

- Anti-whale measures to keep the market stable.

Goal of the project

It is our goal that no holder will be left behind. The whole community will be provided with tools to make their money grow even further. We intend to grow rich together as a community, and we can accomplish this only as a community, as ONE WHALE.

Tokenomics

15% fee when buying and selling

- 8% of trade goes to holders' pockets in BUSD tokens.
- 3% of trade goes to marketing and one whale investment fund wallet.
- 2% of trade goes to the liquidity pool.
- 2% of trade goes to the manual buyback & one whale investment fund wallet.

Roadmap

Phase 1

- ❖ Project Conceptualization and Initial Whitepaper Release.
- ❖ Launch of Website and Social Media Account Creation.
- ❖ Initial AMA.
- ❖ Strategic Partnership and Alliances.
- ❖ Private Presale, Whitelisting, and Public Presale via Pinksale.
- ❖ Launch of POW Token.
- ❖ Initial Marketing Push.
- ❖ Apply for CG and CMC Listings.
- ❖ Commence Investment Process and Prepare Audit Reports for Each Transaction.

Phase 2

- ❖ Conduct Regular AMAs for Old and New Investors.
- ❖ Extensive Marketing Push in all Social Media Platforms and Financial News Content Providers.
- ❖ Expansion of Team (Admin/Staff, Developers, Professional Traders)
- ❖ Extension of Website and Branding.
- ❖ Contract Audit.

Phase 3

- ❖ Launch of Windows DAPPs (Portfolio Tracker and POW Wallet)
- ❖ Launch of VIP Signals and Copy Trading Services, and Trading Bots Service for POW holders.
- ❖ Launch of Android App.
- ❖ Launch of iOS App.

Phase 4

- ❖ Introduction and Launch of The Whale Tank Business Incubator.
- ❖ Creation of POW Launchpad for POW-backed Tokens.

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 BUSD by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part with business incubation system.
- Anyone who's interested in trading bot and signal platform that's going to be built.
- Anyone who's interested in taking part with the future plans of the One Whale token.
- Anyone who's interested in making financial transactions with any other party using BUSD or One Whale as the currency.

Core concept

The One Whale reward system

8% of each transaction when buying and selling get converted to BUSD and is split amongst all holders. Holders will be eligible to receive tokens everyone hour and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The sustainability fee of 3% for marketing and one whale investment fund is what allows One Whale 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, One Whale will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

When you buy and sell POW tokens, 3% of said transactions will go to the One Whale Investment Fund/Marketing Wallet. This fund will then be invested in current or up-and-coming crypto projects that have huge potential through One Whale Wallets. Further, they will also invest in altcoins trading in various Cryptocurrency Exchanges and also in Forex Trading. We will determine all the best avenues for generating revenue streams, and the income from these investments will then be used to buy back One Whale Tokens.

Concurrently, the Marketing Wallet will be used to develop the Project: One Whale brand, inside and outside the crypto space. We will implement a sustained and aggressive marketing strategy, which will concentrate on the following core areas:

- Spreading the POW Brand and Image
- Unrelentless Hype and Traction Generation
- Running Cross-Platform Ads and News Articles
- Social Media Push through Sponsorships and Influencers
- R&D For Future POW Integrated Projects and New Use Cases

However, marketing will not be limited to attracting new investors. Part of the Marketing Wallet is earmarked for community building and investor retention programs. We believe that it is vital to create a loyal investor base as they will serve as the firm foundation of the project. To incentivize the investors to hold, we will conduct engaging community contests, competitions, and raffle draws with major prizes to celebrate milestone achievements. Simply put, we will not rest on our laurels and be content in making POW Token the best coin in the BSC, we want to be known as having the best community as well, bar none.

The liquidity fee of 2%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The controlled buyback collects 2% tax on each transaction, which is stored inside a private wallet. It is to save from massive dips in order to keep the token market price stable.

One Whale wallet usage

The One Whale Wallet is somewhat akin to a mutual fund wherein investment for other tokens, altcoins, and forex will be pooled together to generate income for POW. Several One Whale Wallets will be created in the blockchain, and this will serve as the common investment for all POW holders. It is prudent that we will invest in various tokens and altcoins since we believe in the adage of not putting all our eggs in one basket. We cannot assure that all investments will be successful but making multiple investments would increase our chances of getting a handsome profit in every investment cycle that we make.

Profits from successful investments will then be collected and pooled and shall be divided as follows:

- 40% will be used to buyback POW tokens
- 40% will be added to the ONE Whale Investment Fund
- 10% will be for operational expenses (office rental and upkeep, utilities, payroll of organic employees, and compensation of professional traders)
- 10% will be earmarked for marketing and development

Anti-whale measures

No single private wallet will be allowed to sell more than 0.25% of the total supply to avoid large holders having too much pricing over the market.

Future plan

II. POW DAPPS, DASHBOARD, AND HOLDER WALLET

We will add DApps to our website and create a mobile app dashboard with the following services - portfolio token tracker, BUSD rewards tracker, and personal wallet so POW holders will always be updated with their portfolio, whether they are at home or on the go.

III. VIP TRADING SIGNALS, COPY TRADING, AND TRADING BOTS SUBSCRIPTION SERVICE

Next, we will create a VIP Trading Signals, Copy Trading App, and Trading Bot Service intended for POW holders who want to reinvest the funds they earned from POW and make their money work for them.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	10/10
3	Information quality	10/10
4	Service quality	9/10
5	System quality	8/10
6	Impact on the community	9/10
7	Impact on the business	10/10
8	Preparing for the future	9/10
Total Points		**9.25/10**

Contract details

Token contract details for 12th November 2021

Contract name	One Whale
Contract address	0x0196D4D37990aB2e1f7BC8607C7b4198A360c48f
Token supply	1,000,000,000,000,000
Token ticker	POW
Decimals	9
Token holders	2
Transaction count	2
MASK	340282366920938463463374607431768211455
WBNB	0xbb4cdb9cbd36b01bd1cbaebf2de08d9173bc095c
Auto liquidity receiver	0x1237d0929953af18e616714e8c446f4e5468f052
Distributor	0x78b135f9d7e2029a643459cfd0ad3cb99c1315bc
Marketing fee receiver	0x9a2d20d4029b6fa75488449e895617ca3f565c4f
Contract deployer address	0xF5bF3226A191b778b40568B3a6DF119ADe5b492a
Contract's current owner address	0xf5bf3226a191b778b40568b3a6df119ade5b492a

Token distribution

Tokens are distributed as follows:

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	medium issue
		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	low issue
		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
SafeMath	**Library**			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		

└	mod	Internal 🔒		
IBEP20	**Interface**			
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🔘	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🔘	NO❗
└	transferFrom	External ❗	🔘	NO❗
Auth	**Implementation**			
└		Public ❗	🔘	NO❗
└	authorize	Public ❗	🔘	onlyOwner
└	unauthorize	Public ❗	🔘	onlyOwner
└	isOwner	Public ❗		NO❗

└	isAuthorized	Public ❗		NO❗
└	transferOwnership	Public ❗	🔘	onlyOwner
IDEXFactory	**Interface**			
└	createPair	External ❗	🔘	NO❗
IDEXRouter	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🔘	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
IDividendDistributor	**Interface**			
└	setDistributionCriteria	External ❗	🔘	NO❗

└	setShare	External ❗	🛑	NO❗
└	deposit	External ❗	💵	NO❗
└	process	External ❗	🛑	NO❗
IPinkAntiBot	**Interface**			
└	setTokenOwner	External ❗	🛑	NO❗
└	onPreTransferCheck	External ❗	🛑	NO❗
DividendDistributor	**Implementation**	**IDividendDistributor**		
└		Public ❗	🛑	NO❗
└	setDistributionCriteria	External ❗	🛑	onlyToken
└	setShare	External ❗	🛑	onlyToken
└	deposit	External ❗	💵	onlyToken
└	process	External ❗	🛑	onlyToken
└	shouldDistribute	Internal 🔒		
└	distributeDividend	Internal 🔒	🛑	
└	claimDividend	External ❗	🛑	NO❗
└	getUnpaidEarnings	Public ❗		NO❗
└	getCumulativeDividends	Internal 🔒		

└	addShareholder	Internal 🔒	🔘	
└	removeShareholder	Internal 🔒	🔘	
POW	**Implementation**	**IBEP20, Auth**		
└		Public ❗	🔘	Auth
└		External ❗	💸	NO❗
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	Public ❗	🔘	NO❗
└	approveMax	External ❗	🔘	NO❗
└	transfer	External ❗	🔘	NO❗
└	transferFrom	External ❗	🔘	NO❗
└	_transferFrom	Internal 🔒	🔘	

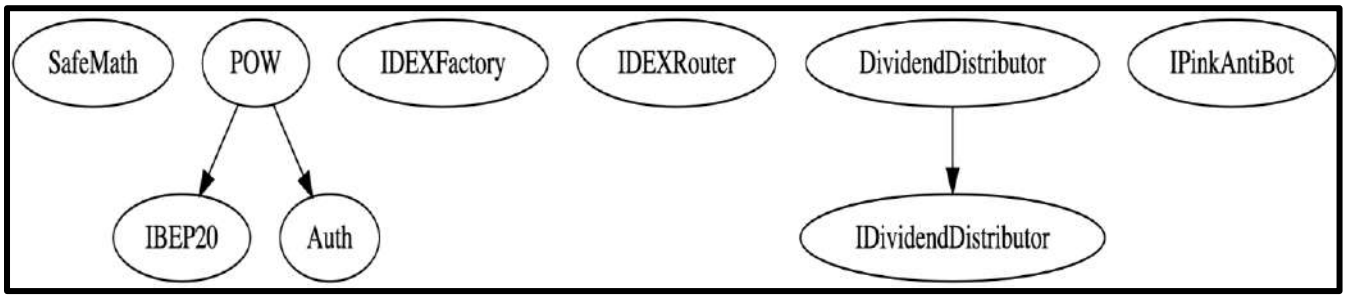
└	_basicTransfer	Internal 🔒	🛑	
└	setEnableAntiBot	External ❗	🛑	onlyOwner
└	checkTxLimit	Internal 🔒		
└	shouldTakeFee	Internal 🔒		
└	getTotalFee	Public ❗		NO❗
└	getMultipliedFee	Public ❗		NO❗
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	swapBack	Internal 🔒	🛑	swapping
└	shouldAutoBuyback	Internal 🔒		
└	triggerZeusBuyback	External ❗	🛑	authorized
└	sweep	External ❗	🛑	onlyOwner
└	clearBuybackMultiplier	External ❗	🛑	authorized
└	triggerAutoBuyback	Internal 🔒	🛑	
└	buyTokens	Internal 🔒	🛑	swapping
└	setAutoBuybackSettings	External ❗	🛑	authorized
└	setBuybackMultiplierSettings	External ❗	🛑	authorized
└	launched	Internal 🔒		

└	launch	Public ❗	🔘	authorized
└	setTxLimit	External ❗	🔘	authorized
└	setIsDividendExempt	External ❗	🔘	authorized
└	setIsFeeExempt	External ❗	🔘	authorized
└	setIsTxLimitExempt	External ❗	🔘	authorized
└	setFees	External ❗	🔘	authorized
└	setFeeReceivers	External ❗	🔘	authorized
└	setSwapBackSettings	External ❗	🔘	authorized
└	setTargetLiquidity	External ❗	🔘	authorized
└	setDistributionCriteria	External ❗	🔘	authorized
└	setDistributorSettings	External ❗	🔘	authorized
└	getCirculatingSupply	Public ❗		NO❗
└	getLiquidityBacking	Public ❗		NO❗
└	isOverLiquified	Public ❗		NO❗

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**

- **Unused Functions**
  buyBack feature is not happening internally but buyback functions and buyBack multiplier functions are still in the contract.

```
ftrace | funcSig
function clearBuybackMultiplier() external authorized {
    buybackMultiplierTriggeredAt = 0;
}

ftrace | funcSig
function triggerAutoBuyback() internal {
    buyTokens(autoBuybackAmount, DEAD);
    autoBuybackBlockLast = block.number;
    autoBuybackAccumulator = autoBuybackAccumulator.add(autoBuybackAmount);
    if (autoBuybackAccumulator > autoBuybackCap) {
        autoBuybackEnabled = false;
    }
}

ftrace | funcSig
function buyTokens(uint256 amount⬆, address to⬆) internal swapping {
    address[] memory path = new address[](2);
    path[0] = WBNB;
    path[1] = address(this);

    router.swapExactETHForTokensSupportingFeeOnTransferTokens{
        value: amount⬆
    }(0, path, to⬆, block.timestamp);
}
```

❖ Low severity issues

- The addLiquidity function calls the uniswapV2Router.addLiquidityETH function with the address specified as owner() for acquiring the generated LP tokens from the OneWhale-BNB pool. As a result,over time the _owner address will accumulate a significant portion of LP tokens. If the _owner is an EOA(Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Recommendation

We advise the address of the uniswapV2Router.addLiquidityETH function call to be replaced by the contract itself.

```
if (amountToLiquify > 0) {
    router.addLiquidityETH{value: amountBNBLiquidity}(
        address(this),
        amountToLiquify,
        0,
        0,
        autoLiquidityReceiver,
        block.timestamp
    );
    emit AutoLiquify(amountBNBLiquidity, amountToLiquify);
}
```

Owner privileges

- The owner can enable and disable the anti bot feature.

```
ftrace | funcSig
function setEnableAntiBot(bool _enable⬆) external onlyOwner {
    antiBotEnabled = _enable⬆;
}
```

- The owner can get the contract BNB balance to the owner's wallet.

```
ftrace | funcSig
function sweep() external onlyOwner {
    address payable _owner = payable(msg.sender);
    _owner.transfer(address(this).balance);
}
```

- The owner can set launch time in contract.

```
ftrace | funcSig
function launch() public authorized {
    require(launchedAt == 0, "Already launched boi");
    launchedAt = block.number;
    launchedAtTimestamp = block.timestamp;
}
```

- The owner can change the max transaction limit.

```
ftrace | funcSig
function setTxLimit(uint256 amount⬆) external authorized {
    require(amount⬆ >= _totalSupply / 1000);
    _maxTxAmount = amount⬆;
}
```

❖ The owner can exempt wallets from dividends.

```
ftrace | funcSig
function setIsDividendExempt(address holder⬆, bool exempt⬆)
    external
    authorized
{
    require(holder⬆ != address(this) && holder⬆ != pair);
    isDividendExempt[holder⬆] = exempt⬆;
    if (exempt⬆) {
        distributor.setShare(holder⬆, 0);
    } else {
        distributor.setShare(holder⬆, _balances[holder⬆]);
    }
}
```

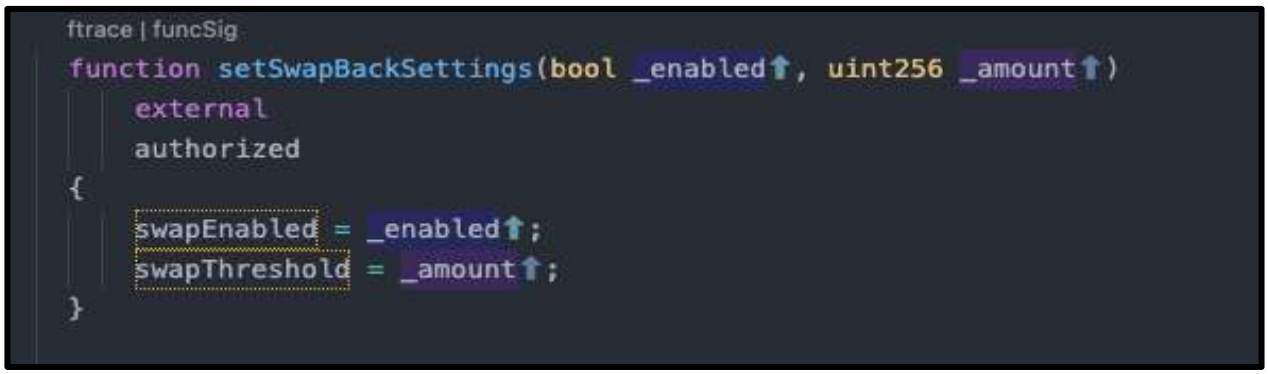
❖ The owner can exempt wallets from fees and max transaction limit.

```
ftrace | funcSig
function setIsFeeExempt(address holder⬆, bool exempt⬆) external authorized {
    isFeeExempt[holder⬆] = exempt⬆;
}

ftrace | funcSig
function setIsTxLimitExempt(address holder⬆, bool exempt⬆)
    external
    authorized
{
    isTxLimitExempt[holder⬆] = exempt⬆;
}
```

❖ The owner can change all fees and fee receivers.

```
ftrace | funcSig
function setFees(
    uint256 _liquidityFee⬆,
    uint256 _buybackFee⬆,
    uint256 _reflectionFee⬆,
    uint256 _marketingFee⬆,
    uint256 _feeDenominator⬆
) external authorized {
    liquidityFee = _liquidityFee⬆;
    buybackFee = _buybackFee⬆;
    reflectionFee = _reflectionFee⬆;
    marketingFee = _marketingFee⬆;
    totalFee = _liquidityFee⬆.add(_buybackFee⬆).add(_reflectionFee⬆).add(
        _marketingFee⬆
    );
    feeDenominator = _feeDenominator⬆;
    require(totalFee < feeDenominator / 4);
}

ftrace | funcSig
function setFeeReceivers(
    address _autoLiquidityReceiver⬆,
    address _marketingFeeReceiver⬆
) external authorized {
    autoLiquidityReceiver = _autoLiquidityReceiver⬆;
    marketingFeeReceiver = _marketingFeeReceiver⬆;
}
```

❖ The owner can change swapBack settings.



```
ftrace | funcSig
function setSwapBackSettings(bool _enabled⬆, uint256 _amount⬆)
    external
    authorized
{
    swapEnabled = _enabled⬆;
    swapThreshold = _amount⬆;
}
```

❖ The owner can change target liquidity settings.

```
ftrace | funcSig
function setTargetLiquidity(uint256 _target⬆, uint256 _denominator⬆)
    external
    authorized
{
    targetLiquidity = _target⬆;
    targetLiquidityDenominator = _denominator⬆;
}
```

❖ The owner can change distribution criteria and settings.

```
ftrace | funcSig
function setDistributionCriteria(
    uint256 _minPeriod⬆,
    uint256 _minDistribution⬆
) external authorized {
    distributor.setDistributionCriteria(_minPeriod⬆, _minDistribution⬆);
}

ftrace | funcSig
function setDistributorSettings(uint256 gas⬆) external authorized {
    require(gas⬆ < 750000);
    distributorGas = gas⬆;
}
```

Audit conclusion

While conducting the audit of the One Whale smart contract, it was observed that there is nothing alarming with the code and it only contains a medium severity issue and a low severity issue.