RugFreeCoins Audit

Daddy Token

Smart Contract Security Audit

June 29, 2021

Contents

Audit details

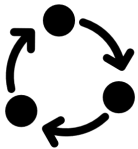**Audited project**

Daddy Token

Contract Address

0xcd0EA4c1BC5d1639ad86e6c074CA0c624014d6Cb

Client contact

Daddy Token Team

Blockchain

Binance smart chain

Project website

https://www.daddytoken.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 Daddy Token to perform an audit of the smart contract.

https://bscscan.com/token/0xcd0EA4c1BC5d1639ad86e6c074CA0c624014d6Cb

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

Daddy is a token built on the Binance Smart Chain. Each transaction, purchase incur a 15% fee, and sales incur an 18% fee.

Token Comparison	TIKI	GHOSTFACE	HODL	DADDY
Redistribution in BNB	✔	✔	✔	✔
Liquidity Tax	✔	✔	✔	✔
Token Redistribution Tax	✖	✔	✔	✔
Audited?	✖	✔	✔	✔
Selling Penalty to Prevent Dumps & Day Traders	✔	✖	✖	✔
Marketing Tax for Lifetime Marketing	✖	✖	✖	✔
Charity Tax	✖	✖	✖	✔
Automatic BNB distribution?	✔	✖	✖	✔
Anti Dump Features	✔	✖	✖	✔
Long Term Self-Sustainability	✖	✖	✖	✔

The **static reward** system is what DADDY's entire marketing strategy is based around, especially in the early stages of the life of the coin. The notion of 'passive income' can be leveraged in DADDY's marketing approach and facilitate the next steps in the development and growth of the community.

The **sustainability fee of 3% marketing** is what allows DADDY to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Daddy will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

The additional component included under the sustainability section is a liquidity fee, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.

1% Charity fee per transaction will be swapped to BNB and sent to the Binance charity wallet. This will empower the DADDY community in the long run, and motivate more people to join in!

Tokenomics

15% fee when buying

- ❖ 6% of every trade goes to holders pockets in BNB.
- ❖ 1% of every trade goes to holders pockets in tokens.
- ❖ 3% of every trade goes to the marketing wallet.
- ❖ 1% of every trade goes to a charity wallet.
- ❖ 4% of every trade goes to the liquidity pool.

18% fee when selling (Additional 3%)

- ❖ 1% of every trade goes to the marketing wallet.
- ❖ 1% of every trade goes to the liquidity pool.
- ❖ 1% of every trade will get burnt.

Roadmap

DADDYMAP

PHASE 1	PHASE 2	PHASE 3	PHASE 4
✅ Website, Community, Socials	⬤ $5M Market Cap	⬤ $25M Market Cap	⬤ Full Doxx of Dev Team
✅ Parternship with Auditing team	⬤ Cluster Marketing	⬤ Launch of Reward Calculator	⬤ YT live video AMAs
⬤ DxSale Presale	⬤ YT/IG/Twitter Influencers	⬤ Parternship with ChildHelp	⬤ Retail Investor Targeting
⬤ PancakeSwap & Liquidity Lock	⬤ CoinGecko Listing	⬤ Migration to Discord community	⬤ $100M Market Cap
⬤ $1M Market Cap	⬤ CoinMarketCap Listing		

Target market and the concept

Target market

- Anyone who's interested in Crypto space with long term investment plans.
- Anyone who's ready to earn a passive income by holding tokens. (Through BNBs & tokens)
- Anyone who's interested in supporting a good cause. (Donation feature)
- Anyone who's interested in trading tokens.
- Anyone who's interested in making financial transactions with any other party using DADDY as the currency.

Core concept

➢ **The BNB reward system**

6% of each transaction gets converted to BNBs, and is split amongst all holders **automatically in a loop**. The rewards are sent in a loop to holders that have at least 4200 DADDY tokens, holders will be eligible to receive tokens every one hour and rewards are proportional to how many tokens each individual holds.

➢ **Sustainable mechanism**

The **fee of 3% marketing** is what allows DADDY to promote the token and use funds to further development of the platform. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Daddy 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 liquidity fee of 4%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

➢ **Anti-dumping and anti-whale strategy**

DADDY contract includes a function that stops ALL sales above 0.1% of the total supply. This will discourage (mini)-whales from dumping all their bags at once.

Anti pump and dump: Certain groups of individuals practice pump and dump schemes, in order to lure in outside investors by the looks of a bullish chart, and sell at a high point. DADDY charges an additional 3% fee for all sales.

> **Good cause**

1% Charity fee per transaction will be swapped to BNB and sent to the Binance charity wallet.

Apart from that, DADDY will spread awareness, focusing specifically on **children protection** from child abuse spedning a perecetage from the marketing wallet. This problem, which is Universal, is common all around the world but rarely discussed, especially in underdeveloped countries, and does not require donations as much as it necessitates the spread of awareness.

> **The plan**

DADDY will create engaging community contests, competitions, creativity challenges, as well as empower the community through inviting each member to participate and collaborate in their Decision-making process, from development all the way to marketing.

Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	9/10
3	Information quality	8/10
4	Service quality	9/10
5	System quality	8/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	8/10
Total Points		**8.25/10**

Contract details

Token contract details for 29th June 2021

Contract name	DADDY
Contract address	0xcd0EA4c1BC5d1639ad86e6c074CA0c624014d6Cb
Token supply	420,000,000
Token ticker	$DADDY
Decimals	9
Token holders	3
Transaction count	6
Top 100% holders dominance	Not launched yet
Contract deployer address	0x1Df925D52a7dade6411Cf669fa796F1e62bBc509
Contract's current owner address	0x24BFe3F0621193E84CccD75Ee56BCeC7a3d7bcCF

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		
L	Function Name	Visibility	Mutability	Modifiers
Address	**Library**			
L	isContract	Internal 🔒		
L	sendValue	Internal 🔒	🛑	
L	functionCall	Internal 🔒	🛑	
L	functionCall	Internal 🔒	🛑	
L	functionCallWithValue	Internal 🔒	🛑	
L	functionCallWithValue	Internal 🔒	🛑	
L	functionStaticCall	Internal 🔒		
L	functionStaticCall	Internal 🔒		
L	functionDelegateCall	Internal 🔒	🛑	
L	functionDelegateCall	Internal 🔒	🛑	
L	_verifyCallResult	Private 🔐		
IBEP20	**Interface**			
L	totalSupply	External ❗		NO❗
L	decimals	External ❗		NO❗
L	symbol	External ❗		NO❗

L	name	External ❗		NO ❗
L	getOwner	External ❗		NO ❗
L	balanceOf	External ❗		NO ❗
L	transfer	External ❗	🛑	NO ❗
L	allowance	External ❗		NO ❗
L	approve	External ❗	🛑	NO ❗
L	transferFrom	External ❗	🛑	NO ❗
SafeBEP20	**Library**			
L	safeTransfer	Internal 🔒	🛑	
L	safeTransferFrom	Internal 🔒	🛑	
L	safeApprove	Internal 🔒	🛑	
L	safeIncreaseAllowance	Internal 🔒	🛑	
L	safeDecreaseAllowance	Internal 🔒	🛑	
L	_callOptionalReturn	Private 🔐	🛑	
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
IERC20	**Interface**			
L	totalSupply	External ❗		NO ❗
L	balanceOf	External ❗		NO ❗
L	transfer	External ❗	🛑	NO ❗
L	allowance	External ❗		NO ❗

11

L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
L	mod	Internal 🔒		
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO❗
L	owner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
IUniswapV2Factory	**Interface**			
L	feeTo	External ❗		NO❗
L	feeToSetter	External ❗		NO❗
L	getPair	External ❗		NO❗
L	allPairs	External ❗		NO❗
L	allPairsLength	External ❗		NO❗
L	createPair	External ❗	🛑	NO❗

L	setFeeTo	External ❗️	🛑	NO❗️
L	setFeeToSetter	External ❗️	🛑	NO❗️
IUniswapV2Router01	**Interface**			
L	factory	External ❗️		NO❗️
L	WETH	External ❗️		NO❗️
L	addLiquidity	External ❗️	🛑	NO❗️
L	addLiquidityETH	External ❗️	💵	NO❗️
L	removeLiquidity	External ❗️	🛑	NO❗️
L	removeLiquidityETH	External ❗️	🛑	NO❗️
L	removeLiquidityWithPermit	External ❗️	🛑	NO❗️
L	removeLiquidityETHWithPermit	External ❗️	🛑	NO❗️
L	swapExactTokensForTokens	External ❗️	🛑	NO❗️
L	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❗️
IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗️	🛑	NO❗️
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗️	🛑	NO❗️
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗️	🛑	NO❗️
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗️	💵	NO❗️
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗️	🛑	NO❗️
DADDY	**Implementation**	**Context, IERC20, Ownable**		
L		Public ❗️	🛑	NO❗️
L	name	Public ❗️		NO❗️
L	symbol	Public ❗️		NO❗️
L	decimals	Public ❗️		NO❗️
L	totalSupply	Public ❗️		NO❗️
L	balanceOf	Public ❗️		NO❗️
L	transfer	Public ❗️	🛑	NO❗️
L	allowance	Public ❗️		NO❗️
L	approve	Public ❗️	🛑	NO❗️

L	transferFrom	Public ❗	🛑	NO❗
L	increaseAllowance	Public ❗	🛑	NO❗
L	decreaseAllowance	Public ❗	🛑	NO❗
L	isExcludedFromReward	Public ❗		NO❗
L	totalFees	Public ❗		NO❗
L	reflectionFromToken	Public ❗		NO❗
L	tokenFromReflection	Public ❗		NO❗
L	excludeFromReward	Public ❗	🛑	onlyOwner
L	includeInReward	External ❗	🛑	onlyOwner
L	addAdminAccount	External ❗	🛑	onlyOwner
L	removeAdminAccount	External ❗	🛑	onlyOwner
L	excludeFromFee	External ❗	🛑	onlyOwner
L	includeInFee	External ❗	🛑	onlyOwner
L	setTaxFeePercent	External ❗	🛑	onlyOwner
L	setConvertBNBFeePercent	External ❗	🛑	onlyOwner
L	setLiquidityFeePercent	External ❗	🛑	onlyOwner
L	setSwapAndLiquifyEnabled	External ❗	🛑	onlyOwner
L	setMaxTxAmount	External ❗	🛑	onlyOwner
L	setMaxTxAmountNumber	External ❗	🛑	onlyOwner
L	_reflectFee	Private 🔐	🛑	
L	_getValues	Private 🔐		
L	_getTValues	Private 🔐		

L	_getRValues	Private 🔒		
L	_getRate	Private 🔒		
L	_getCurrentSupply	Private 🔒		
L	_takeLiquidity	Private 🔒	🛑	
L	_takeBNB	Private 🔒	🛑	
L	calculateTaxFee	Private 🔒		
L	calculateConvertBNBFee	Private 🔒		
L	calculateLiquidityFee	Private 🔒		
L	removeAllFee	Private 🔒	🛑	
L	restoreAllFee	Private 🔒	🛑	
L	setFeesAndSwapAndLiquifyAndRewardSystem	External ❗	🛑	onlyOwner
L	isAdminAccount	Public ❗		NO❗
L	isExcludedFromFee	Public ❗		NO❗
L	_approve	Private 🔒	🛑	
L	_transfer	Private 🔒	🛑	
L	swapTokensForEthAndSwapAndLiquify	Private 🔒	🛑	lockTheSwap
L	_tokenTransfer	Private 🔒	🛑	
L	distributeBNBReward	Private 🔒	🛑	
L	distributeBNBRewardManual	External ❗	🛑	NO❗
L	removeIndexFromRewardArray	Private 🔒	🛑	
L	removeIndexFromRewardArrayOwnerOnly	External ❗	🛑	onlyOwner

L	setBNBBuyResetThresholdPercent	External ❗	🛑	onlyOwner
L	setSupplyAmountCompareForBNBRewards	External ❗	🛑	onlyOwner
L	addExemptWalletFromBNBRewards	External ❗	🛑	onlyOwner
L	removeExemptWalletFromBNBRewards	External ❗	🛑	onlyOwner
L	setMinimumBNBInContractToReward	External ❗	🛑	onlyOwner
L	setBNBMaxUntilCharityCut	External ❗	🛑	onlyOwner
L	setMinimumAmountToHoldForRewards	External ❗	🛑	onlyOwner
L	setMaxNumOfTransersToDoForReward	External ❗	🛑	onlyOwner
L	setRewardSystemEnabledOrDisabled	External ❗	🛑	onlyOwner
L	setHoursForRewardTimer	External ❗	🛑	onlyOwner
L	setNewPercentOfCharityCut	External ❗	🛑	onlyOwner
L	setNewCharityWallet	External ❗	🛑	onlyOwner
L	viewCurrentBNBinContract	Public ❗		NO❗
L	payableAddress	Private 🔐		
L	setRouterAddress	Public ❗	🛑	onlyOwner
L	setPairAddress	Public ❗	🛑	onlyOwner
L	_transferStandard	Private 🔐	🛑	
L		External ❗	💵	NO❗

Symbol	Meaning
⬤	**Function can modify state**
🔳	**Function is payable**

Inheritance Hierarchy
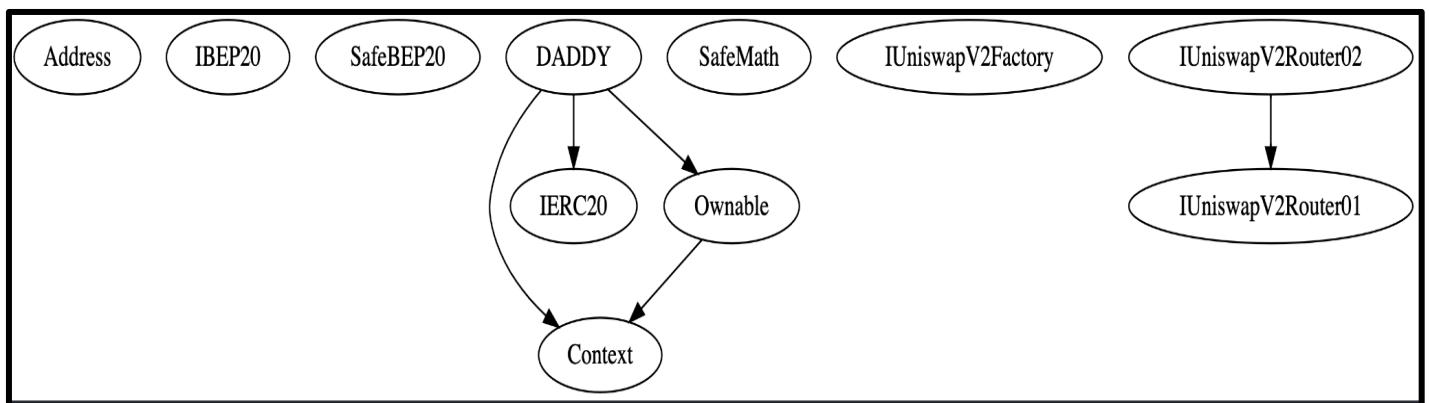
Security issue checking status

❖ **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
(In the period when the owner is not renounced)

❖ The owner can change the minimum BNB amount in the BNB pool to perform the BNB reward.

```
ftrace | funcSig
function setMinimumBNBInContractToReward(uint256 newBNBMinimum↑) external onlyOwner() {
    bnbMinimumAmountInContractToReward = newBNBMinimum↑;
}
```

❖ The owner can change the minimum token hold amount to get the BNB reward.

```
ftrace | funcSig
function setMinimumAmountToHoldForRewards(uint256 newMinimumAmountToHoldForRewards↑) external onlyOwner() {
    minimumAmountToHoldForRewards = newMinimumAmountToHoldForRewards↑;
}
```

❖ The owner can change the maximum number of BNB reward distributions per transaction.

```
ftrace | funcSig
function setMaxNumOfTransersToDoForReward(uint256 newMaxNumOfTransfers↑) external onlyOwner() {
    maxNumOfTransfersToDoForReward = newMaxNumOfTransfers↑;
}
```

❖ The owner can enable/disable the reward system

```
ftrace | funcSig
function setRewardSystemEnabledOrDisabled(bool isRewardSystemEnabledNew↑) external onlyOwner() {
    isRewardSystemEnabled = isRewardSystemEnabledNew↑;
}
```

❖ The owner can change the minimum reward receival time per holder.

```
ftrace | funcSig
function setHoursForRewardTimer(uint256 newNoursToAddForReward↑) external onlyOwner() {
    hoursToAddForReward = newNoursToAddForReward↑;
}
```

❖ The owner can change the charity wallet.

```
ftrace | funcSig
function setNewCharityWallet(address newCharityWallet↑) external onlyOwner() {
    charityWallet = newCharityWallet↑;        // new charity wallet
}
```

❖ The owner can remove wallets from the reward.

```
ftrace | funcSig
function removeExemptWalletFromBNBRewards(address removeAddress↑) external onlyOwner() {
    isAddressExemptFromBNBRewards[removeAddress↑] = false;
}
```

❖ The owner can call the reward system manually.

```
ftrace | funcSig
function distributeBNBRewardManual() external {
    require(isRewardSystemEnabled, "Reward system is not enabled");
    distributeBNBReward();
}
```

❖ The owner can enable and disable all fees.

```
ftrace | funcSig
function setFeesAndSwapAndLiquifyAndRewardSystem() external onlyOwner() {
    _taxFee = 1;
    _convertBNBFee = 12;
    _liquidityFee = 3;
    isRewardSystemEnabled = true;
    isSwapAndLiquifyEnabled = true;
    _maxTxAmount = 420000 * 10**9;
}
```

❖ The owner can change the maximum transaction amount.

```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount↑) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxAmount↑).div(
        10**2
    );
}
```

❖ The owner can enable and disable liquidity swap.

```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool _enabled↑) external onlyOwner() {
    isSwapAndLiquifyEnabled = _enabled↑;
    emit SwapAndLiquifyEnabledUpdated(_enabled↑);
}
```

❖ The owner can change the liquidity fee.

```
ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner() {
    require(liquidityFee↑ <= 5, "Percentage too high. Please use a lower fee.");
    _liquidityFee = liquidityFee↑;
}
```

❖ The owner can change the transaction fee.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner() {
    require(taxFee↑ <= 5, "Percentage too high. Please use a lower fee.");
    _taxFee = taxFee↑;
}
```

❖ The owner can transfer the ownership

```
ftrace | funcSig
function transferOwnership(address newOwner↑) public virtual onlyOwner {
    require(newOwner↑ != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner↑);
    _owner = newOwner↑;
}
```

❖ The owner can renounce the ownership

```
ftrace | funcSig
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}
```

Audit conclusion

While conducting the audit of the DADDY smart contract, it was observed that there is nothing alarming with the code and the contract has no issues.