RugFreeCoins Audit

Baby Avengers Token Audit

Smart Contract Security Audit

August 11, 2021

Contents

Audit details

Audited project

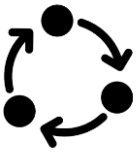
Baby Avengers Token

Contract Address

0xdad6d006ab190e536e2d256526d038d67303f577

Client contact

Baby Avengers Team



Blockchain

Binance smart chain

Project website

https://www.babyavengers.com/

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 BabyAvengers to perform an audit of the smart contract.

https://bscscan.com/token/0xdad6d006ab190e536e2d256526d038d67303f577

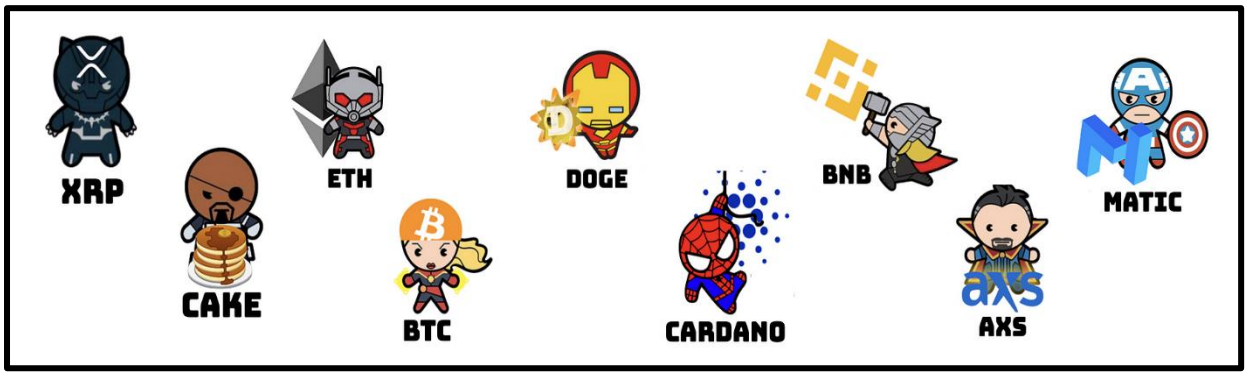
The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

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

Baby Avengers is a unique project built on Binance smart chain, where every 3 days, the community decides which Baby Avenger will be given as a reward. This revolutionary token automatically rewards holders with dividends from a variety of currencies. Investors can build a complete portfolio simply by holding Baby Avengers tokens.

- BTC (Captain Marvel)
- ETH (Antman)
- BNB (Thor)
- MATIC (Captain America)
- XRP (Black Panther)
- DOGE (Iron Man)
- CAKE
- CARDANO
- AXS



Tokenomics

16% tax fee when buying and selling

- 14% of every trade goes to holders' pockets in 9 different coins as per the community decision every 3 days. (BTC, ETH, XRP, BNB, DOGE, MATIC, CARDANO, CAKE, AXS)
- 2% of every trade goes to the marketing & development wallet.
- 2% of every trade goes to the liquidity pool.

Roadmap

Phase 1
Age of Baby Ultron

- Creation of Socials and Website
- Creation of Community and Organic Growth
- 15% of Supply Burned
- Dev Wallet Locked
- Presale and Liquidity Lock (secured by DxSale)
- PCS Launch

Phase 2
Babynity War

- Listing on CoinGecko
- BSCScan icons
- 5,000 holders
- TrustWallet Logo
- Twitter and Telegram Marketing Campaign
- Shilling Contest

Phase 3
End game

- Referral Contest
- Listing on CoinMarketCap
- DexTools trending
- 10,000 holders
- First Baby Avengers Episode released
- Moon

Target market and the concept

Target market

- Anyone who's interested in Crypto space with long term investment plans.
- Anyone who's ready to build a completely different portfolio with different currencies by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part in Baby Avengers future plans.
- Anyone who's interested in making financial transactions with any other party using Baby Avengers as the currency.

Core concept

The reward system

12% of each transaction when buying and selling will be deducted and sent amongst all holders in 9 different currencies. Every 3 days, the community decides which Baby Avenger will be given as a reward. (BTC, ETH, XRP, BNB, DOGE, MATIC, CARDANO, CAKE, AXS)
The rewards are sent to holders that have at least 1 Baby Avengers tokens, holders will be eligible to receive rewards proportional to how many tokens each individual holds.

Sustainable mechanism

The **fee of 2% marketing and team expenses** is what allows Baby Avengers to use them 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, Baby Avenger's 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 2%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Anti-bot measures

No single wallet will be allowed to purchase more than 0.5% of the total supply in one transaction. This restriction, coupled with a 50 second cooldown period, will limit bot users from fluctuating the price at launch or at any period in the life of the token.

White-listed presale

To prevent bots from snatching all the tokens, wallets will be whitelisted to take part with the presale.

Anti-whale measures

No single private wallet will be allowed to own more than 1% of the total supply to avoid large holders having too much pricing power over the rest of the community.

Potential to grow with score points

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

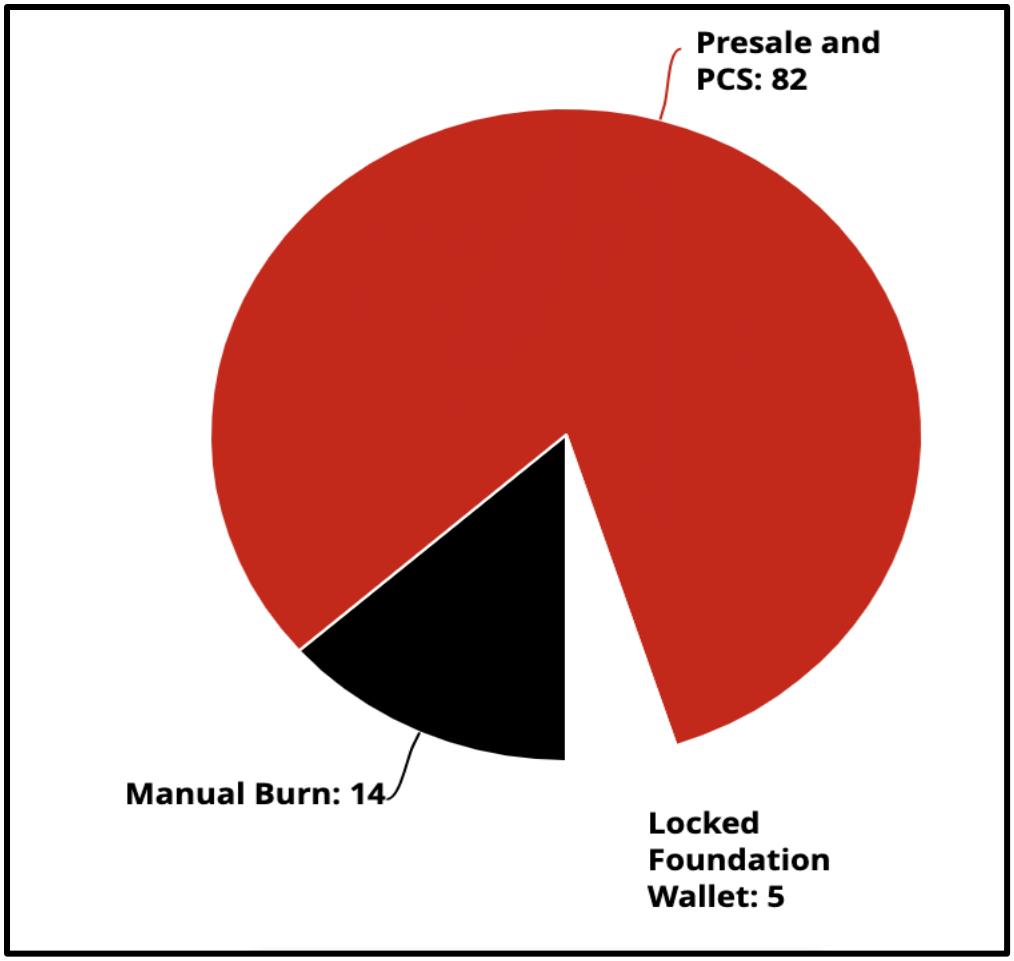
Contract details

Token contract details for 11th August 2021

Contract name	BabyAvengers Token
Contract address	0xdad6d006ab190e536e2d256526d038d67303f577
Token supply	1,000,000,000,000
Token ticker	AVNGRS
Decimals	4
Token holders	2
Transaction count	2
Distributor address	0xd15e9d357ef1e3ee4c799e84d243247b23a0bf29
Marketing address	Not publicly visible
Contract deployer address	0x67f8fE8f4b664451365e6bd5D514d43A7693965a
Contract's current owner address	0x67f8fe8f4b664451365e6bd5d514d43a7693965a

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	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	Low issue
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**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	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❗

L	addLiquidityETH	External ❗		NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗		NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗		NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗		NO❗
IDividendDistributor	**Interface**			
L	setDistributionCriteria	External ❗		NO❗
L	setShare	External ❗		NO❗
L	deposit	External ❗		NO❗
L	process	External ❗		NO❗
L	purge	External ❗		NO❗
DividendDistributor	**Implementation**	**IDividendDistributor**		
L		Public ❗		NO❗
L	setDistributionCriteria	External ❗		onlyToken
L	purge	External ❗		onlyToken
L	setShare	External ❗		onlyToken
L	deposit	External ❗		onlyToken
L	process	External ❗		onlyToken

L	shouldDistribute	Internal 🔒		
L	distributeDividend	Internal 🔒	🛑	
L	claimDividend	External ❗	🛑	NO❗
L	getUnpaidEarnings	Public ❗		NO❗
L	getCumulativeDividends	Internal 🔒		
L	addShareholder	Internal 🔒	🛑	
L	removeShareholder	Internal 🔒	🛑	
BabyAvengers	**Implementation**	**IBEP20, Auth**		
L		Public ❗	🛑	Auth
L		External ❗	💵	NO❗
L	totalSupply	External ❗		NO❗
L	decimals	External ❗		NO❗
L	symbol	External ❗		NO❗
L	name	External ❗		NO❗
L	getOwner	External ❗		NO❗
L	balanceOf	Public ❗		NO❗
L	allowance	External ❗		NO❗
L	approve	Public ❗	🛑	NO❗
L	approveMax	External ❗	🛑	NO❗

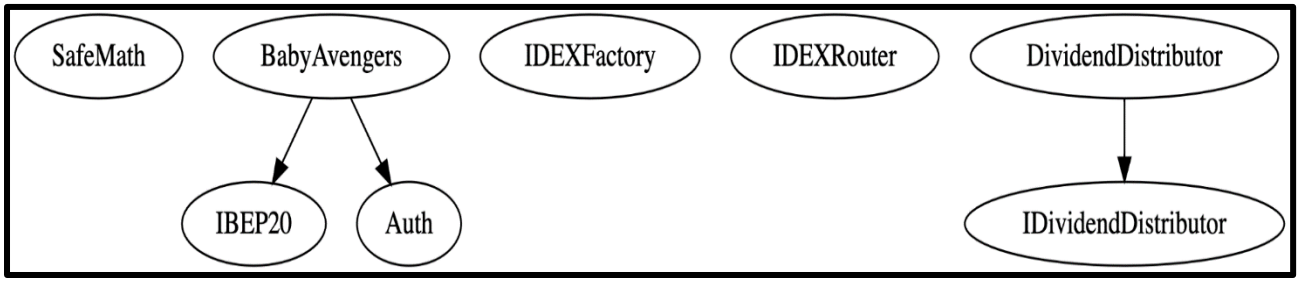
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	setMaxWalletPercent	External ❗	🛑	onlyOwner
└	_transferFrom	Internal 🔒	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	checkTxLimit	Internal 🔒		
└	shouldTakeFee	Internal 🔒		
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	clearStuckBalance	External ❗	🛑	onlyOwner
└	tradingStatus	Public ❗	🛑	onlyOwner
└	cooldownEnabled	Public ❗	🛑	onlyOwner
└	purgeBeforeSwitch	Public ❗	🛑	onlyOwner
└	switchToken	Public ❗	🛑	onlyOwner
└	___claimRewards	Public ❗	🛑	NO❗
└	claimProcess	Public ❗	🛑	NO❗
└	swapBack	Internal 🔒	🛑	swapping
└	setTxLimit	External ❗	🛑	authorized
└	setIsDividendExempt	External ❗	🛑	authorized
└	setIsFeeExempt	External ❗	🛑	authorized

L	setIsTxLimitExempt	External ❗	🛑	authorized
L	setIsTimelockExempt	External ❗	🛑	authorized
L	setFees	External ❗	🛑	authorized
L	setFeeReceivers	External ❗	🛑	authorized
L	setSwapBackSettings	External ❗	🛑	authorized
L	setTargetLiquidity	External ❗	🛑	authorized
L	setDistributionCriteria	External ❗	🛑	authorized
L	setDistributorSettings	External ❗	🛑	authorized
L	getCirculatingSupply	Public ❗		NO❗
L	getLiquidityBacking	Public ❗		NO❗
L	isOverLiquified	Public ❗		NO❗
L	airdrop	External ❗	🛑	onlyOwner
L	airdropFixed	External ❗	🛑	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**

  Wrong calculation

  Max transaction amount and max wallet hold amount calculation are wrong in the contract code, but still, the owner can change these amounts at any time.

```
uint256 _totalSupply = 1 * 10**12 * (10 ** _decimals);
uint256 public _maxTxAmount = _totalSupply * 100 / 100;

//max wallet holding of 2%
uint256 public _maxWalletToken = ( _totalSupply * 100 ) / 100;
```

Owner privileges

- The owner can send the BNB balance to the marketing wallet.

```
ftrace | funcSig
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
    uint256 amountBNB = address(this).balance;
    payable(marketingFeeReceiver).transfer(amountBNB * amountPercentage / 100);
}
```

- The owner can enable and disable trading any time.

```
// switch Trading
ftrace | funcSig
function tradingStatus(bool _status) public onlyOwner {
    tradingOpen = _status;
}
```

- The owner can change reward tokens.

```
// new dividend tracker
ftrace | funcSig
function switchToken(address rewardToken) public onlyOwner {
    distributor = new DividendDistributor(address(router), rewardToken);
}
```

- The owner can enable/disable buy cooldown and change cool down time.

```
// enable cooldown between trades
ftrace | funcSig
function cooldownEnabled(bool _status, uint8 _interval) public onlyOwner {
    buyCooldownEnabled = _status;
    cooldownTimerInterval = _interval;
}
```

❖ The owner can add/remove authorized persons.

```
/**
 * Authorize address. Owner only
 */
ftrace | funcSig
function authorize(address adr⬆) public onlyOwner {
    authorizations[adr⬆] = true;
}

/**
 * Remove address' authorization. Owner only
 */
ftrace | funcSig
function unauthorize(address adr⬆) public onlyOwner {
    authorizations[adr⬆] = false;
}

/**
```

❖ The owner can transfer the ownership.

```
/**
 * Transfer ownership to new address. Caller must be owner. Leaves old owner authorized
 */
ftrace | funcSig
function transferOwnership(address payable adr⬆) public onlyOwner {
    owner = adr⬆;
    authorizations[adr⬆] = true;
    emit OwnershipTransferred(adr⬆);
}
```

Authorized persons privileges

(By default, the owner is an authorized person, and the owner has the same privileges given below. The owner can add any wallet with the below privileges.)

- Authorized persons can change max transaction amount**.**

```
ftrace | funcSig
function setTxLimit(uint256 amount⬆) external authorized {
    _maxTxAmount = amount⬆;
}
```

- Authorized persons 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⬆]);
    }
}
```

- Authorized persons can exempt wallets from fees.

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

- Authorized persons can exempt wallets from max transaction limits.

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

❖ Authorized persons can exempt wallets from buy time lock.

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

❖ Authorized persons can change all fees.

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

❖ Authorized persons can change the liquidity wallet and the marketing wallet.

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

❖ Authorized persons can enable/disable buy back and buy back threshold.

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

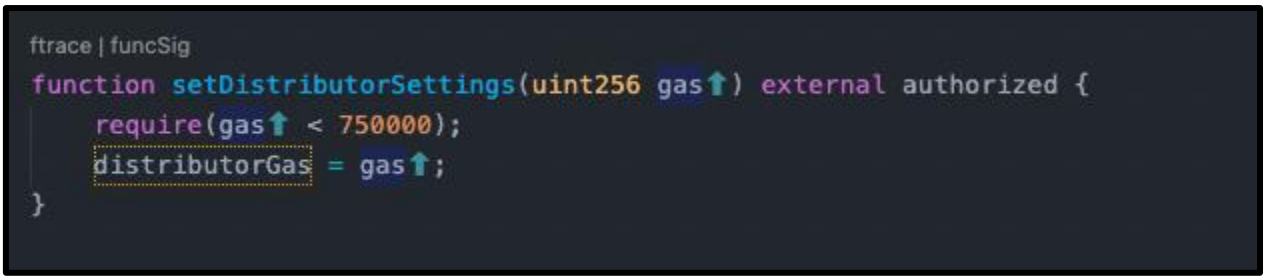
❖ Authorized persons can change target liquidity.

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

❖ Authorized persons can change minimum distribution time and distribution amount.

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

- Authorized persons can change distribution gas fees up to 750000.



Audit conclusion

While conducting the audit of the Baby Avengers Token smart contract, it was observed that there is nothing alarming with the code but only a low severity issue.