



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



PugDroid  
\$DROID



01/02/2022



# TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3 WEBSITE DIAGNOSTIC
- 4-5 AUDIT OVERVIEW
- 6-7 OWNER PRIVILEGES
- 8 CONCLUSION AND ANALYSIS
- 9 TOKEN DETAILS
- 10 PUGDROID TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS
- 11 TECHNICAL DISCLAIMER



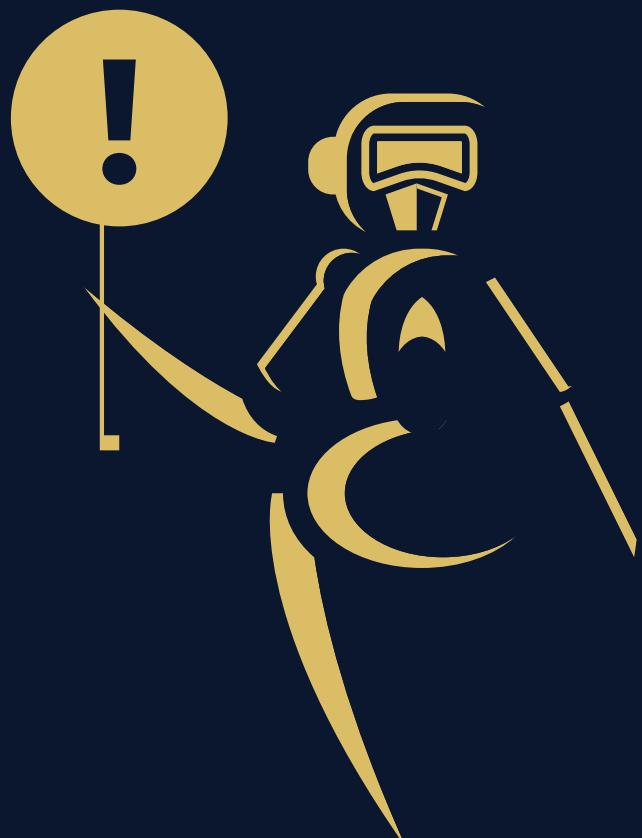
# DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website <https://freshcoins.io>

FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy ( RUG or Honeypot etc )



# INTRODUCTION

**FreshCoins (Consultant) was contracted by  
PugDroid (Customer) to conduct a Smart Contract Code Review  
and Security Analysis.**

**0xc5a2F11D8870f84A585C5cd67357D2cdCb2114D9**

**Network: Binance Smart Chain (BSC)**

**This report presents the findings of the security assessment of  
Customer's smart contract and its code review conducted on 01/02/2022**



# WEBSITE DIAGNOSTIC

<https://www.pugdroid.com/>



0-49



50-89



90-100



Performance



Accessibility



Best Practices



SEO



Progressive  
Web App

## Metrics



First Contentful Paint

**1.7 s**



Time to interactive

**3.7 s**



Speed Index

**4.2 s**



Total Blocking Time

**60 ms**



Large Contentful Paint

**2.6 s**



Cumulative Layout Shift

**0**

# AUDIT OVERVIEW



**Security Score**



**Static Scan**  
Automatic scanning for common vulnerabilities



**ERC Scan**  
Automatic checks for ERC's conformance



**High**



**Medium**



**Low**



**Optimizations**



**Informational**



No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Passed
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Passed
8	Methods execution permissions	Passed
9	Exchange rate impact	Passed
10	Malicious Event	Passed
11	Scoping and Declarations	Passed
12	Uninitialized storage pointers	Passed
13	Design Logic	Passed
14	Safe Zeppelin module	Passed

# OWNER PRIVILEGES

Contract owner can't mint tokens after initial contract deploy.

Contract owner can exclude/include wallet(s) from fee

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "BABYTOKEN: Account is already the value of 'excluded'"
    );
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _setOwner(address(0));
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    _setOwner(newOwner);
}
```

Contract owner can change tx amount

```
function setSwapTokensAtAmount(uint256 amount) external onlyOwner {
    swapTokensAtAmount = amount;
}
```

## Contract owner can change fees up to 25%

```
function setTokenRewardsFee(uint256 value) external onlyOwner {  
    tokenRewardsFee = value;  
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);  
    require(totalFees <= 25, "Total fee is over 25%");  
}  
  
function setLiquidityFee(uint256 value) external onlyOwner {  
    liquidityFee = value;  
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);  
    require(totalFees <= 25, "Total fee is over 25%");  
}  
  
function setMarketingFee(uint256 value) external onlyOwner {  
    marketingFee = value;  
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);  
    require(totalFees <= 25, "Total fee is over 25%");  
}
```

## Contract owner can change `_marketingWalletAddress` address

Current address:

`_marketingWalletAddress = 0x68988b553c294123eb5d2686fcdf19bf91ea613e`

```
function setMarketingWallet(address payable wallet) external onlyOwner {  
    _marketingWalletAddress = wallet;  
}
```

### 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. The risk can be prevented by temporarily locking the contract or renouncing ownership.**



# CONCLUSION AND ANALYSIS



Smart Contracts within the scope were manually reviewed and analyzed with static tools.



Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.



Found no issue during the first review.

# TOKEN DETAILS

## Details

Buy fees: 12%

Sell fees: 12%

Max TX: N/A

Max Sell: N/A

## Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Detected

Modify Max Sell: Not detected

Disable Trading: Not detected

## Rug Pull Risk

Liquidity: N/A

Holders: Clean



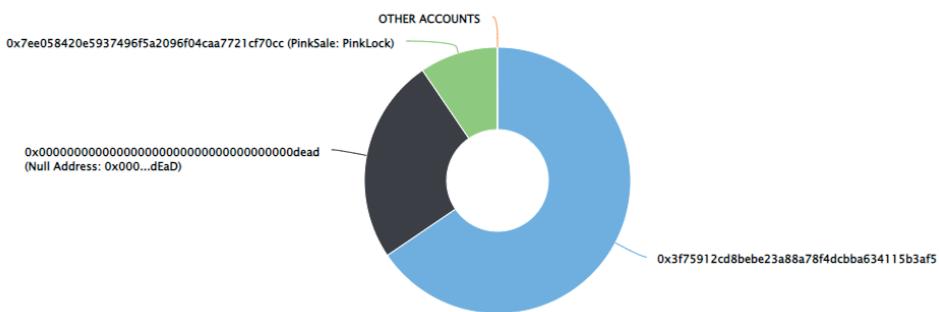
# PUGDROID TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (130,000.00 Tokens) of PugDroid

Token Total Supply: 130,000.00 Token | Total Token Holders: 3

PugDroid Top 10 Token Holders

Source: BscScan.com



(A total of 130,000.00 tokens held by the top 10 accounts from the total supply of 130,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x3f75912cd8bebe23a88a78f4dcba634115b3af5	85,150	65.5000%
2	Null Address: 0x000...dEaD	32,500	25.0000%
3	PinkSale: PinkLock	12,350	9.5000%

# TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.

