



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



ApesToken
\$AEPS



13/03/2022



TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-7 OWNER PRIVILEGES
- 8 CONCLUSION AND ANALYSIS
- 9 TOKEN DETAILS
- 10 APESTOKEN TOKEN ANALYTICS & 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 ApesToken (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xC0BF954B1d17e292dc5f7D4b6e8266a710292eC8

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 13/03/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance

0 **High**

0 **Medium**

0 **Low**

0 **Optimizations**

0 **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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet from tax

```
function setIsExcludedFromFee(address account, bool newValue)
    public
    onlyOwner
{
    isExcludedFromFee[account] = newValue;
}
```

Contract owner can exclude/include wallet from wallet limitations

```
function setIsWalletLimitExempt(address holder, bool exempt)
    external
    onlyOwner
{
    isWalletLimitExempt[holder] = exempt;
}
```

Contract owner can exclude wallet from wallet limitations and tax

```
function removeTaxFreeAddress(address newAddress) external onlyOwner {
    isExcludedFromFee[address(newAddress)] = false;
    isWalletLimitExempt[address(newAddress)] = false;
}
```

Contract owner can enable/disable wallet limitations

```
function enableDisableWalletLimit(bool newValue) external onlyOwner {
    checkWalletLimit = newValue;
}
```

Contract owner can burn tokens (transfer to dead address)

```
function burn(address account, uint256 amount) public onlyBurner {
    _basicTransfer(account, deadAddress, amount);
}
```

Contract owner can change max tx amount

```
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner {
    _maxTxAmount = maxTxAmount;
}
```

Contract owner can change the fees up to 100%

```
function setTaxes(
    uint256 newLiquidityTax,
    uint256 newMarketingTax,
    uint256 newBuyBackTax,
    uint256 newExtraFeeOnSell
) external onlyOwner {
    _liquidityFee = newLiquidityTax;
    _marketingFee = newMarketingTax;
    _buyBackFee = newBuyBackTax;
    _extraFeeOnSell = newExtraFeeOnSell;

    _totalTaxIfBuying = _liquidityFee.add(_marketingFee).add(_buyBackFee);
    _totalTaxIfSelling = _totalTaxIfBuying.add(_extraFeeOnSell);
}
```

Contract owner can set wallet limit amount

```
function setWalletLimit(uint256 newLimit) external onlyOwner {
    _walletMax = newLimit;
}
```

Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}

function setSwapAndLiquifyByLimitOnly(bool newValue) public onlyOwner {
    swapAndLiquifyByLimitOnly = newValue;
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = 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"
    );
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

Contract owner can change marketingWalletAddress and buyBackWalletAddress addresses

Current values:

marketingWalletAddress : 0x0ba196bd0878e458f4b9da67192f198ea5aab902

buyBackWalletAddress : 0xcb2e13c2818511ecb2e3cd59b81cf82d5a43da84

```
function setMarketingWalletAddress(address newAddress) external onlyOwner {  
    marketingWalletAddress = payable(newAddress);  
}  
  
function setbuyBackWalletAddress(address newAddress) external onlyOwner {  
    buyBackWalletAddress = payable(newAddress);  
}
```



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:	10%
Sell fees:	11%
Max TX:	1,000,000,000
Max Sell:	N/A

Honeypot Risk

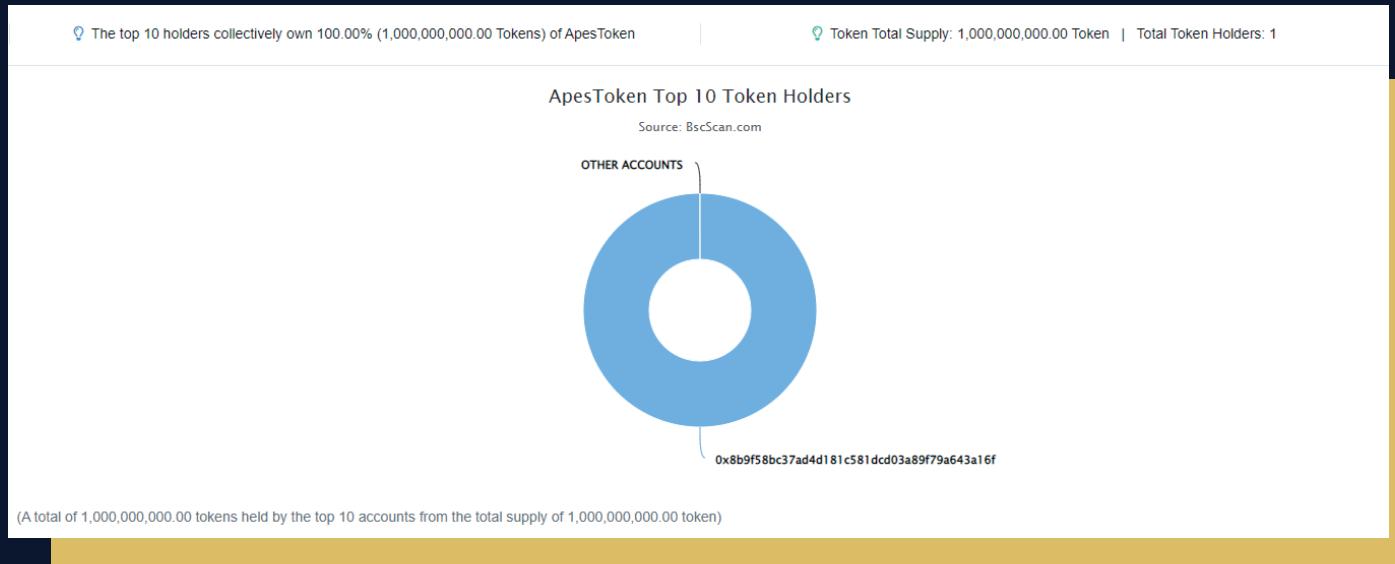
Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



APESTOKEN TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	0x8b9f58bc37ad4d181c581dcd03a89f79a643a16f	1,000,000,000	100.0000%

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.

