



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



ApeClockToken
\$APEC

23/04/2022

TABLE OF CONTENTS

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

0x2fE42E717eA2c4438b5a9704A529428B352DDd31

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 23/04/2022



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 mint tokens after initial contract deploy

Contract owner can exclude/include only contract address from transactions

```
function setBotBlacklist(address _botAddress, bool _flag) external onlyOwner {
    require(isContract(_botAddress), "only contract address, not allowed externally owned account");
    blacklist[_botAddress] = _flag;
}
```

Contract owner can change rebase settings

```
function setAutoRebase(bool _flag) external onlyOwner {
    if (_flag) {
        _autoRebase = _flag;
        _lastRebasedTime = block.timestamp;
    } else {
        _autoRebase = _flag;
    }
}
```

Contract owner can change `autoLiquidityReceiver`, `treasuryReceiver`, `apeInsuranceFundReceiver`, `pairAddress` and `pairContract` addresses

```
function setFeeReceivers(
    address _autoLiquidityReceiver,
    address _treasuryReceiver,
    address _apeInsuranceFundReceiver,
    address _firePit
) external onlyOwner {
    autoLiquidityReceiver = _autoLiquidityReceiver;
    treasuryReceiver = _treasuryReceiver;
    apeInsuranceFundReceiver = _apeInsuranceFundReceiver;
    firePit = _firePit;
}

function setPairAddress(address _pairAddress) public onlyOwner {
    pairAddress = _pairAddress;
}

function setLP(address _address) external onlyOwner {
    pairContract = IPancakeSwapPair(_address);
}
```

Contract owner can exclude/include wallet from tax

```
function setWhitelist(address _addr) external onlyOwner {
    _isFeeExempt[_addr] = true;
}
```

Contract owner can change referral settings

Current values:

```
uint256 public minToRefer = 10;
uint256 public minToReferDenominator = 10000;

function approveReferral(address referrer) external {
    require(isReferrerValid(referrer), "Referrer doesn't have enough tokens to refer others");
    require(referrals[msg.sender] == address(0), "You have already been referred");
    referrals[msg.sender] = referrer;
}

function isReferrerValid(address referrer) internal view returns (bool) {
    return _gonBalances[referrer].div(_gonsPerFragment) >= _totalSupply.mul(minToRefer).div(-minToReferDenominator) || approvedReferrers[referrer];
}

function setMinToRefer(uint256 _minToRefer) external onlyOwner {
    require(_minToRefer <= minToReferDenominator, "Invalid value");
    minToRefer = _minToRefer;
}

function setApprovedReferrer(address referrer, bool flag) external onlyOwner {
    approvedReferrers[referrer] = flag;
}

function setDiscounts(uint256[] memory fees) external onlyOwner {
    uint256 _totalDiscount = fees[0].add(fees[1]).add(fees[2]).add(fees[3]);
    require(_totalDiscount >= 20 && totalDiscount <= totalFee.div(2), "Invalid value");
    liquidityFeeDiscount = fees[0];
    treasuryFeeDiscount = fees[1];
    apecInsuranceFundFeeDiscount = fees[2];
    firePitFeeDiscount = fees[3];
    totalDiscount = _totalDiscount;
}

function setReferralComission(uint256 comission) external onlyOwner {
    require(totalDiscount.sub(comission) >= 10, "Invalid value");
    referralFee = comission;
}

function setReferralProgramEnabled(bool flag) external onlyOwner {
    referralProgramEnabled = flag;
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public onlyOwner {
    emit OwnershipRenounced(_owner);
    _owner = address(0);
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public onlyOwner {
    _transferOwnership(newOwner);
}

function _transferOwnership(address newOwner) internal {
    require(newOwner != address(0));
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```



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: 12%

Max TX: N/A

Max Sell: N/A

Honeypot Risk

Ownership: Owned

Blacklist: Detected

Modify Max TX: Not detected

Modify Max Sell: Not detected

Disable Trading: Not detected

Rug Pull Risk

Liquidity: N/A

Holders: Clean



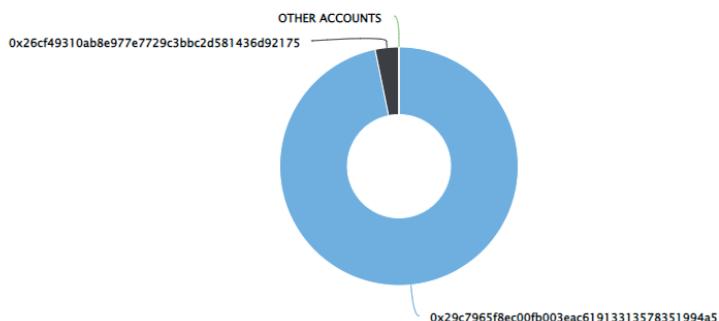
APECLOCKTOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (500,000.00 Tokens) of ApeClockToken

Token Total Supply: 500,000.00 Token | Total Token Holders: 2

ApeClockToken Top 10 Token Holders

Source: BscScan.com



(A total of 500,000.00 tokens held by the top 10 accounts from the total supply of 500,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x29c7965f8ec00fb003eac61913313578351994a5	484,000	96.8000%
2	0x26cf49310ab8e977e7729c3bbc2d581436d92175	16,000	3.2000%

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.

