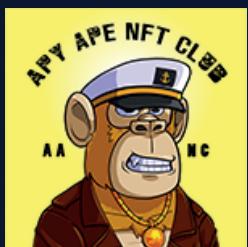




**freshcoins**

## **SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT**



**APYAPENFTCLUB**  
**\$AANC**



**01/05/2022**



# TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 APYAPENFTCLUB TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 10 TECHNICAL DISCLAIMER



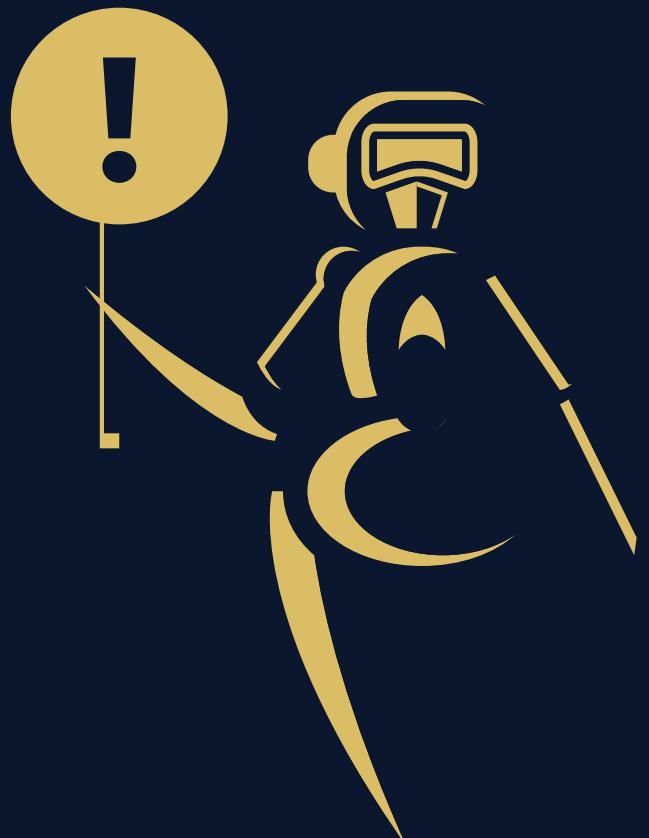
# DISCLAIMER

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 APYAPENFTCLUB (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xC05D272967fca47fefe985ff8b347eE147852aF8

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 01/05/2022



# 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 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`, `AANCInsuranceFundReceiver`, `firePit` and `pairContract` addresses

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

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

Contract owner can exclude 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 exclude wallet from tax

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

## Contract owner can withdraw tokens from smart contract

```
function withdrawAllToTreasury() external swapping onlyOwner {  
    uint256 amountToSwap = _gonBalances[address(this)].div(_gonsPerFragment);  
    require( amountToSwap > 0,"There is no APYAPENFTCLUB token deposited in token contract");  
    address[] memory path = new address[](2);  
    path[0] = address(this);  
    path[1] = router.WETH();  
    router.swapExactTokensForETHSupportingFeeOnTransferTokens(  
        amountToSwap,  
        0,  
        path,  
        treasuryReceiver,  
        block.timestamp  
    );  
}
```

## 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: 14%

Sell fees: 16%

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



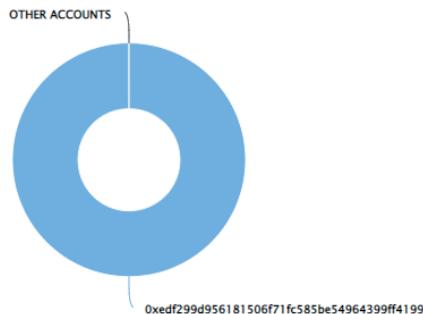
# APYAPENFTCLUB TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

💡 The top 10 holders collectively own 100.00% (650,000.00 Tokens) of APYAPENFTCLUB

💡 Token Total Supply: 650,000.00 Token | Total Token Holders: 1

APYAPENFTCLUB Top 10 Token Holders

Source: BscScan.com



(A total of 650,000.00 tokens held by the top 10 accounts from the total supply of 650,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xedf299d956181506f71fc585be54964399ff4199	650,000	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.

