



# Smart Contract Audit

FOR

# Pharaohs Gems

DATED : 29 April 23'



# AUDIT SUMMARY

**Project name - Pharaohs Gems**

**Date:** 29 April, 2023

**Scope of Audit-** Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

**Audit Status: Passed**

## Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	2	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0



# USED TOOLS

---

## Tools:

### 1- Manual Review:

A line by line code review has been performed by audit ace team.

**2- BSC Test Network:** All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

### 3- Slither :

The code has undergone static analysis using Slither.

### Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0xff8116808592f933eb003f57affb25f30d790a2f>

---



# Token Information

---

**Token Name :** Pharaohs Gems

**Token Symbol:** PHARAOH

**Decimals:** 18

**Token Supply:** 1,000,000,000

**Token Address:**

0x20b0c5F0f3a2DC1EEeC3BC55341C1e2Fd811cEcD

**Checksum:**

4a763e723e5ce238f3865ef4e6084da724c5cdf0

**Owner:**

0xD4e88838f3B3FD1bA182a92cF7353e58D9ac814

**(at the time of writing the audit)**

**Deployer:**

0xe13aa455ae125B2E90436C3c23a896856Fc56a25

---



# TOKEN OVERVIEW

---

**Fees:**

Buy Fees: up to 10%

Sell Fees: up to 10%

Transfer Fees: 0%

---

**Fees Privilege:** None

---

**Ownership:** Owned

---

**Minting:** No mint function

---

**Max Tx Amount/ Max Wallet Amount:** No

---

**Blacklist:** No

---

**Other Privileges:** modifying swap threshold - toggling internal swap - excluding wallets from fee - including wallets in fee - modifying fees - changing buyback threshold

---



# AUDIT METHODOLOGY

---

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

# VULNERABILITY CHECKLIST



Return values of low-level calls



**Gasless Send**



Private modifier



Using block.timestamp



Multiple Sends



Re-entrancy



Using Suicide



Tautology or contradiction



Gas Limit and Loops



Timestamp Dependence



Address hardcoded



Revert/require functions



Exception Disorder



Use of tx.origin



Using inline assembly



Integer overflow/underflow



Divide before multiply



Dangerous strict equalities



Missing Zero Address Validation



Using SHA3



Compiler version not fixed



Using throw



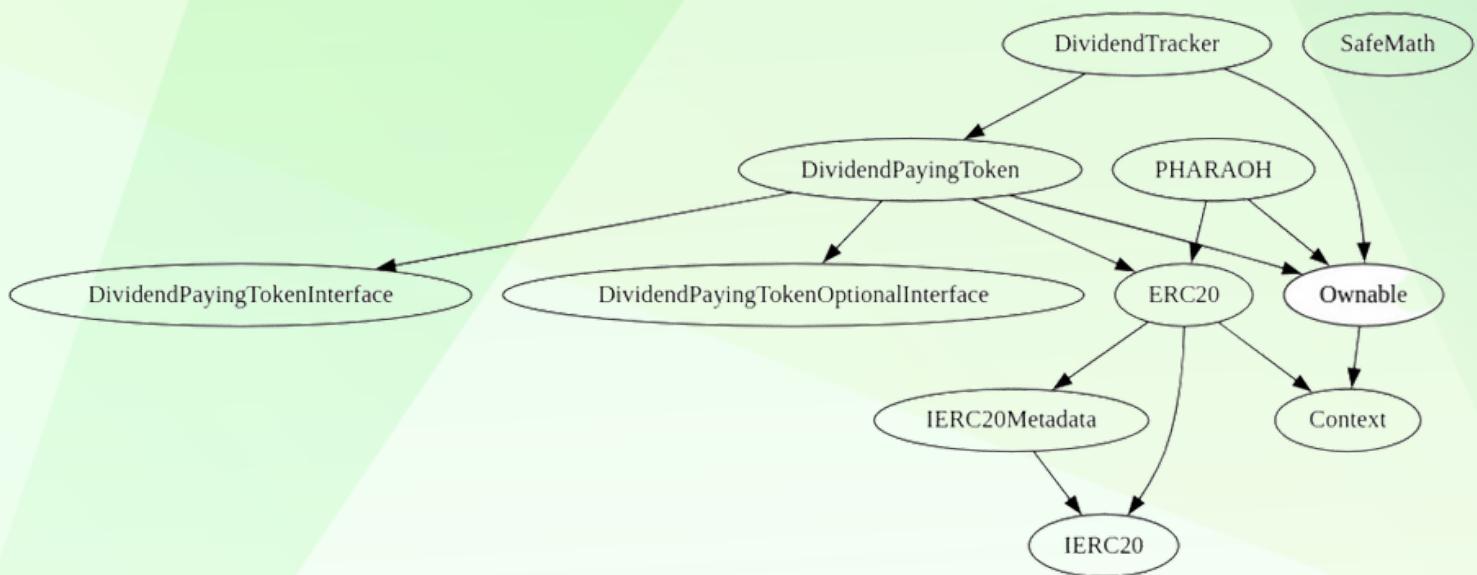
# CLASSIFICATION OF RISK

Severity	Description
◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

## Findings

Severity	Found
◆ Critical	0
◆ High-Risk	2
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

# INHERITANCE TREE





## POINTS TO NOTE

---

- Owner is able to change buy/sell each one up to 10%
- Owner is not able to set transfer taxes (0% forever)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- **Owner has to enable trades manually for holders**
- Owner is not able to mint new tokens
- Token has BNB rewards with auto-distribution
- Fees are used for rewards (BNB), buyback and marketing



# CONTRACT ASSESSMENT

Contract	Type	Bases			
			**Function Name**	**Visibility**	**Mutability**
					**Modifiers**
					**Mutability**
					**Modifiers**
**Context**	Implementation				
L   _msgSender   Internal	🔒				
L   _msgData   Internal	🔒				
**Ownable**	Implementation	Context			
L   <Constructor>   Public	!	●   NO	!		
L   owner   Public	!	NO	!		
L   renounceOwnership   Public	!	●   onlyOwner			
L   transferOwnership   Public	!	●   onlyOwner			
L   _transferOwnership   Internal	🔒	●			
**SafeMath**	Library				
L   add   Internal	🔒				
L   sub   Internal	🔒				
L   sub   Internal	🔒				
L   mul   Internal	🔒				
L   div   Internal	🔒				
L   div   Internal	🔒				
L   mod   Internal	🔒				
L   mod   Internal	🔒				
**SafeMathInt**	Library				
L   mul   Internal	🔒				
L   div   Internal	🔒				
L   sub   Internal	🔒				
L   add   Internal	🔒				
L   abs   Internal	🔒				
L   toUint256Safe   Internal	🔒				
**SafeMathUint**	Library				
L   toInt256Safe   Internal	🔒				
**IterableMapping**	Library				
L   get   Internal	🔒				
L   getIndexOfKey   Internal	🔒				
L   getKeyAtIndex   Internal	🔒				
L   size   Internal	🔒				
L   set   Internal	🔒	●			



# CONTRACT ASSESSMENT

L   remove   Internal	🔒	●
**IUniswapV2Factory**   Interface		
L   feeTo   External	!	NO !
L   feeToSetter   External	!	NO !
L   getPair   External	!	NO !
L   allPairs   External	!	NO !
L   allPairsLength   External	!	NO !
L   createPair   External	!	●   NO !
L   setFeeTo   External	!	●   NO !
L   setFeeToSetter   External	!	●   NO !
**IUniswapV2Pair**   Interface		
L   name   External	!	NO !
L   symbol   External	!	NO !
L   decimals   External	!	NO !
L   totalSupply   External	!	NO !
L   balanceOf   External	!	NO !
L   allowance   External	!	NO !
L   approve   External	!	●   NO !
L   transfer   External	!	●   NO !
L   transferFrom   External	!	●   NO !
L   DOMAIN_SEPARATOR   External	!	NO !
L   PERMIT_TYPEHASH   External	!	NO !
L   nonces   External	!	NO !
L   permit   External	!	●   NO !
L   MINIMUM_LIQUIDITY   External	!	NO !
L   factory   External	!	NO !
L   token0   External	!	NO !
L   token1   External	!	NO !
L   getReserves   External	!	NO !
L   price0CumulativeLast   External	!	NO !
L   price1CumulativeLast   External	!	NO !
L   kLast   External	!	NO !
L   mint   External	!	●   NO !
L   burn   External	!	●   NO !
L   swap   External	!	●   NO !
L   skim   External	!	●   NO !
L   sync   External	!	●   NO !
L   initialize   External	!	●   NO !



# CONTRACT ASSESSMENT

**IUniswapV2Router01**   Interface
L   factory   External !     NO !
L   WETH   External !     NO !
L   addLiquidity   External !     ●   NO !
L   addLiquidityETH   External !     ✅   NO !
L   removeLiquidity   External !     ●   NO !
L   removeLiquidityETH   External !     ●   NO !
L   removeLiquidityWithPermit   External !     ●   NO !
L   removeLiquidityETHWithPermit   External !     ●   NO !
L   swapExactTokensForTokens   External !     ●   NO !
L   swapTokensForExactTokens   External !     ●   NO !
L   swapExactETHForTokens   External !     ✅   NO !
L   swapTokensForExactETH   External !     ●   NO !
L   swapExactTokensForETH   External !     ●   NO !
L   swapETHForExactTokens   External !     ✅   NO !
L   quote   External !     NO !
L   getAmountOut   External !     NO !
L   getAmountIn   External !     NO !
L   getAmountsOut   External !     NO !
L   getAmountsIn   External !     NO !
**IUniswapV2Router02**   Interface   IUniswapV2Router01
L   removeLiquidityETHSupportingFeeOnTransferTokens   External !     ●   NO !
L   removeLiquidityETHWithPermitSupportingFeeOnTransferTokens   External !     ●   NO !
L   swapExactTokensForTokensSupportingFeeOnTransferTokens   External !     ●   NO !
L   swapExactETHForTokensSupportingFeeOnTransferTokens   External !     ✅   NO !
L   swapExactTokensForETHSupportingFeeOnTransferTokens   External !     ●   NO !
**IERC20**   Interface
L   totalSupply   External !     NO !
L   balanceOf   External !     NO !
L   allowance   External !     NO !
L   transfer   External !     ●   NO !
L   approve   External !     ●   NO !
L   transferFrom   External !     ●   NO !
**Address**   Library
L   isContract   Internal 🔒
L   sendValue   Internal 🔒     ●
L   functionCall   Internal 🔒     ●
L   functionCall   Internal 🔒     ●

# CONTRACT ASSESSMENT

L   functionCallWithValue   Internal				
L   functionCallWithValue   Internal				
L   _functionCallWithValue   Private				
**IERC20Metadata**   Interface   IERC20				
L   name   External	!	NO	!	
L   symbol   External	!	NO	!	
L   decimals   External	!	NO	!	
**IPinkAntiBot**   Interface				
L   setTokenOwner   External	!		NO	!
L   onPreTransferCheck   External	!		NO	!
**ERC20**   Implementation   Context, IERC20, IERC20Metadata				
L   <Constructor>   Public	!		NO	!
L   name   Public	!	NO	!	
L   symbol   Public	!	NO	!	
L   decimals   Public	!	NO	!	
L   totalSupply   Public	!	NO	!	
L   balanceOf   Public	!	NO	!	
L   transfer   Public	!		NO	!
L   allowance   Public	!	NO	!	
L   approve   Public	!		NO	!
L   transferFrom   Public	!		NO	!
L   increaseAllowance   Public	!		NO	!
L   decreaseAllowance   Public	!		NO	!
L   _transfer   Internal				
L   _mint   Internal				
L   _burn   Internal				
L   _approve   Internal				
L   _beforeTokenTransfer   Internal				
**DividendPayingTokenInterface**   Interface				
L   dividendOf   External	!	NO	!	
L   withdrawDividend   External	!		NO	!
**DividendPayingTokenOptionalInterface**   Interface				
L   withdrawableDividendOf   External	!	NO	!	
L   withdrawnDividendOf   External	!	NO	!	
L   accumulativeDividendOf   External	!	NO	!	



# CONTRACT ASSESSMENT

| \*\*DividendPayingToken\*\* | Implementation | ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface |||

| L | <Constructor> | Public ! | ● | ERC20 |

| L | distributeDividends | Public ! | ● | onlyOwner |

| L | withdrawDividend | Public ! | ● | NO ! |

| L | \_withdrawDividendOfUser | Internal 🔒 | ● | | |

| L | dividendOf | Public ! | | NO ! |

| L | withdrawableDividendOf | Public ! | | NO ! |

| L | withdrawnDividendOf | Public ! | | NO ! |

| L | accumulativeDividendOf | Public ! | | NO ! |

| L | \_transfer | Internal 🔒 | ● | | |

| L | \_mint | Internal 🔒 | ● | | |

| L | \_burn | Internal 🔒 | ● | | |

| L | \_setBalance | Internal 🔒 | ● | | |

||||||

| \*\*DividendTracker\*\* | Implementation | Ownable, DividendPayingToken |||

| L | <Constructor> | Public ! | ● | DividendPayingToken |

| L | <Receive Ether> | External ! | | \$ | NO ! |

| L | \_transfer | Internal 🔒 | | |

| L | withdrawDividend | Public ! | | NO ! |

| L | updateMinimumTokenBalanceForDividends | External ! | ● | onlyOwner |

| L | excludeFromDividends | External ! | ● | onlyOwner |

| L | updateClaimWait | External ! | ● | onlyOwner |

| L | setLastProcessedIndex | External ! | ● | onlyOwner |

| L | getLastProcessedIndex | External ! | | NO ! |

| L | getNumberOfTokenHolders | External ! | | NO ! |

| L | getAccount | Public ! | | NO ! |

| L | getAccountAtIndex | Public ! | | NO ! |

| L | canAutoClaim | Private 🔒 | | |

| L | setBalance | External ! | ● | onlyOwner |

| L | process | Public ! | ● | NO ! |

| L | processAccount | Public ! | ● | onlyOwner |

||||||

| \*\*PHARAOH\*\* | Implementation | ERC20, Ownable |||

| L | <Constructor> | Public ! | ● | ERC20 |

| L | <Receive Ether> | External ! | | \$ | NO ! |

| L | claimStuckTokens | External ! | ● | onlyOwner |

| L | excludeFromFees | External ! | ● | onlyOwner |

| L | isExcludedFromFees | Public ! | | NO ! |

| L | updateBuyFees | External ! | ● | onlyOwner |

| L | updateSellFees | External ! | ● | onlyOwner |



# CONTRACT ASSESSMENT

L   changeMarketingWallet   External	!	●	onlyOwner
L   enableTrading   External	!	●	onlyOwner
L   _transfer   Internal	🔒	●	
L   swapAndSendDividends   Private	🔒	●	
L   setSwapEnabled   External	!	●	onlyOwner
L   setBNBValueForBuyBurn   Public	!	●	onlyOwner
L   buyBack   Internal	🔒	●	
L   setSwapTokensAtAmount   External	!	●	onlyOwner
L   updateDividendTracker   Public	!	●	onlyOwner
L   updateGasForProcessing   Public	!	●	onlyOwner
L   updateMinimumBalanceForDividends   External	!	●	onlyOwner
L   updateClaimWait   External	!	●	onlyOwner
L   getClaimWait   External	!	NO	!
L   getTotalDividendsDistributed   External	!	NO	!
L   withdrawableDividendOf   Public	!	NO	!
L   dividendTokenBalanceOf   Public	!	NO	!
L   totalRewardsEarned   Public	!	NO	!
L   excludeFromDividends   External	!	●	onlyOwner
L   isExcludedFromDividends   Public	!	NO	!
L   getAccountDividendsInfo   External	!	NO	!
L   getAccountDividendsInfoAtIndex   External	!	NO	!
L   processDividendTracker   External	!	●	NO
L   claim   External	!	●	NO
L   claimAddress   External	!	●	onlyOwner
L   getLastProcessedIndex   External	!	NO	!
L   setLastProcessedIndex   External	!	●	onlyOwner
L   getNumberOfDividendTokenHolders   External	!	NO	!

## Legend

Symbol   Meaning	
----- -----	
●   Function can modify state	
💸   Function is payable	



# STATIC ANALYSIS

```
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/fceo/Token.sol#303) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/fceo/Token.sol#305) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/fceo/Token.sol#336) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/fceo/Token.sol#378) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address) .owner (contracts/fceo/Token.sol#969) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address) .owner (contracts/fceo/Token.sol#974) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address) .owner (contracts/fceo/Token.sol#980) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address) .owner (contracts/fceo/Token.sol#986) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/fceo/Token.sol#911) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter DividendTracker.updateMinimumTokenBalanceForDividends(uint256) .newMinimumBalance (contracts/fceo/Token.sol#1087) is not in mixedCase
Parameter DividendTracker.getAccount(address) .account (contracts/fceo/Token.sol#1132) is not in mixedCase
Parameter PHARAOH.updateBuyFees(uint256,uint256,uint256) .autoBuybackBurnFeeOnBuy (contracts/fceo/Token.sol#1467) is not in mixedCase
Parameter PHARAOH.updateBuyFees(uint256,uint256,uint256) .marketingFeeOnBuy (contracts/fceo/Token.sol#1468) is not in mixedCase
Parameter PHARAOH.updateBuyFees(uint256,uint256,uint256) .rewardFeeOnBuy (contracts/fceo/Token.sol#1469) is not in mixedCase
Parameter PHARAOH.updateSellFees(uint256,uint256,uint256) .autoBuybackBurnFeeOnSell (contracts/fceo/Token.sol#1497) is not in mixedCase
Parameter PHARAOH.updateSellFees(uint256,uint256,uint256) .marketingFeeOnSell (contracts/fceo/Token.sol#1498) is not in mixedCase
Parameter PHARAOH.updateSellFees(uint256,uint256,uint256) .rewardFeeOnSell (contracts/fceo/Token.sol#1499) is not in mixedCase
Parameter PHARAOH.changeMarketingWallet(address) .marketingWallet (contracts/fceo/Token.sol#1529) is not in mixedCase
Parameter PHARAOH.setSwapEnabled(bool) .enabled (contracts/fceo/Token.sol#1695) is not in mixedCase
Variable PHARAOH.DEAD (contracts/fceo/Token.sol#1332) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountDesired (contracts/fceo/Token.sol#383) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountDesired (contracts/fceo/Token.sol#384)
Variable DividendPayingToken._withdrawnDividendOfUser(address) .withdrawableDividend (contracts/fceo/Token.sol#948) is too similar to DividendTracker.getAccount(address).withdrawableDividends (contracts/fceo/Token.sol#1140)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

PHARAOH.setSwapTokensAtAmount(uint256) (contracts/fceo/Token.sol#1718-1725) uses literals with too many digits:
- require(bool,string)(newAmount > totalSupply() / 100000,SwapTokensAtAmount must be greater than 0.001% of total supply) (contracts/fceo/Token.sol#1719-1722)
PHARAOH.updateGasForProcessing(uint256) (contracts/fceo/Token.sol#1760-1771) uses literals with too many digits:
- require(bool,string)(newValue >= 200000 && newValue <= 500000,gasForProcessing must be between 200,000 and 500,000) (contracts/fceo/Token.sol#1761-1764)
PHARAOH.slitherConstructorVariables() (contracts/fceo/Token.sol#1306-1892) uses literals with too many digits:
- gasForProcessing = 300000 (contracts/fceo/Token.sol#1345)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SafeMathInt.MAX.INT256 (contracts/fceo/Token.sol#125) is never used in SafeMathInt (contracts/fceo/Token.sol#123-165)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ERC20.antiBotEnabled (contracts/fceo/Token.sol#722) should be constant
ERC20.pinkAntiBot (contracts/fceo/Token.sol#721) should be immutable
PHARAOH.DEAD (contracts/fceo/Token.sol#1332) should be constant
PHARAOH.rewardToken (contracts/fceo/Token.sol#1344) should be constant
PHARAOH.swapWithLimit (contracts/fceo/Token.sol#1337) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

ERC20.antiBotEnabled (contracts/fceo/Token.sol#722) should be immutable
ERC20.pinkAntiBot (contracts/fceo/Token.sol#721) should be immutable
PHARAOH.uniswapV2Pair (contracts/fceo/Token.sol#1330) should be immutable
PHARAOH.uniswapV2Router (contracts/fceo/Token.sol#1329) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

**Result => A static analysis of contract's source code has been performed using slither,  
No major issues were found in the output**



# FUNCTIONAL TESTING

---

## 1- Adding liquidity (**passed**):

<https://testnet.bscscan.com/tx/0xf9c9d661c90de02f17dc4644175024a5375fd56bcad4bc30a5eee1d2108c30e7>

## 2- Buying when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xeaac2c461dc7683b7641baacfbdcad886e83c14f89035a566ce8baad351e978a>

## 3- Selling when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xfa7a9b7fcf905c497a9a987f2470f2e134c7702af68fec8b754f383a551a943d>

## 4- Transferring when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xab0d6ba78f3ae236dda14c0accfe50bb8768de7e6d8df505a251fc5a3dadd60b>

## 5- Buying when not excluded from fees ( up to 10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x94f204ecdc10e593d1aa3108da46da34605ce894070ff249a02e86433bd7c055>

## 6- Selling when not excluded from fees ( up to 10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x29575351b7fa61cb7dcfea33c957faf392aa52d9be473a1fe5e265d404618d53>

---



# FUNCTIONAL TESTING

---

## 7- Transferring when not excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xa8ab7f6072bd486b9b58ff837c04862eaf2aa4a526fc8e4ab1ea26eda4ea1aa>

## 8- Internal swap (passed):

Marketing wallet received BNB

<https://testnet.bscscan.com/address/0xb2e642d02cc2178a0779b4cae00322de26e4e360#internaltx>

## 9- Buyback(passed):

Buyback happened after reaching the threshold, and tokens were sent to dead wallet

[https://testnet.bscscan.com/token/0xff8116808592f933eb003f57affb25f30d790a2f?a=0x00dead](https://testnet.bscscan.com/token/0xff8116808592f933eb003f57affb25f30d790a2f?a=0x00dead)



# MANUAL TESTING

## Logical – Upgradeable dividend tracker

**Severity:** High

**Function:** updateDividendTracker

**Lines:** 1728

**Status:** Not Resolved

### Overview:

Owner of the contract is able to change dividend tracker to another address. If new dividend tracker has issues (spending lots of gas, not compatible with current token implementation etc), buy/sell/transfers would be disabled.

```
function updateDividendTracker(address newAddress) public onlyOwner {
    require(
        newAddress != address(dividendTracker),
        "The dividend tracker already has that address"
    );
    DividendTracker newDividendTracker = DividendTracker(
        payable(newAddress)
    );
    require(
        newDividendTracker.owner() == address(this),
        "The new dividend tracker must be owned by the token contract"
    );
    try
        newDividendTracker.excludeFromDividends(address(newDividendTracker))
    {} catch {}
    try newDividendTracker.excludeFromDividends(address(this)) {} catch {}
    try newDividendTracker.excludeFromDividends(DEAD) {} catch {}
    try
        newDividendTracker.excludeFromDividends(address(uniswapV2Router))
    {} catch {}
    try
        newDividendTracker.excludeFromDividends(address(uniswapV2Pair))
    {} catch {}
    emit UpdateDividendTracker(newAddress, address(dividendTracker));
    dividendTracker = newDividendTracker;
}
```

### Recommendation:

There are several options to mitigate this issue

- renounce the ownership
- Ensure that dividend tracker is not upgradeable
- Add additional functions to current dividend tracker so be able to change reward token, router address etc. this means we don't need to create a new dividend tracker later in order to update current one.



# MANUAL TESTING

## Centralization - Owner Must Enable Trades

**Severity:** High / Informational

**Function:** enableTrading

**Lines:** 1534

**Status:** Not Resolved

### Overview:

The owner is required to enable trading for investors. If trading remains disabled, token holders will not have the ability to buy, sell, or transfer their tokens.

```
function enableTrading() external onlyOwner {  
    tradingEnabled = true;  
}
```

### Recommendation:

While the presence of this function is considered a feature rather than a flaw, it is crucial to highlight the centralization risk it inherently poses. To address this issue and ensure the enablement of trades, one possible solution would be to transfer the contract's ownership to a trusted third party, such as a Pinksale Safu developer. This would help mitigate the centralization risk associated with this function.



# DISCLAIMER

---

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



# ABOUT AUDITACE

---

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



**<https://auditace.tech/>**



**[https://t.me/Audit\\_Ace](https://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



**<https://github.com/Audit-Ace>**

---