expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Shiba Inu 2.0

TABLE OF CONTENTS

OVERVIEW

The Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Audit Result	Passed
KYC Verification	No
Audit Date	12 May 2023

PROJECT DESCRIPTION

In recent years, decentralized cryptocurrencies have gained a lot of attention and popularity as people seek alternatives to traditional financial systems. Shiba Inu token is one such cryptocurrency that has been making waves in the crypto world.

SOCIAL MEDIA PROFILES

Shiba Inu 2.0

https://t.me/ShibaInu_20

https://twitter.com/shibainu_20

https://www.shibainu20.com

It's always good to check the social profiles of the project, before making your investment.

Team Expelee

CONTRACT DETAILS

Token Name: Shiba Inu 2.0

Symbol: $SHIBVT

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0x8e7642E3BD96eE82FB8C97f4d46a19CCb9355484

Total Supply: 100000000000

Contract SHA-256 Checksum: -

Owner's Wallet:
0xB62e671c48E6188E43A350dd86810D68819DC25e

Deployer's Wallet:
0x8EAce3E7294D6a091030cA9FB8c08361e069aEFc

Testnet:
https://testnet.bscscan.com/address/0x9a0000a721817d5d0ed866e89c3c0edbc67cbbfe

OWNER PRIVILEGES

- Owner can exclude accounts from rewards
- Owner can change currency but not check valid address
- Owner can exclude accounts from fees
- Owner can change buy/sell fees with limit up to 10%
- Trading must be enabled by the owner
- Owner can change the swap tokens at amount within reasonable limit
- Owner can withdraw any token(except native token) from the contract
- Owner can enable pinkantibot system
- Owner can change dividend tracker
- Owner can change updateMinimumBalanceForDividends
- Owner can change updateClaimWait
- Owner can change updateGasForProcessing
- Owner can change updateLastProcessedIndex
- Owner can change the marketing wallet
- Owner can change the buyback wallet

Important Notice:
swapAndSendBuybackburn is going to an externally owned account

AUDIT METHODOLOGY

Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch , that lead to scams and rugpulls.

Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability

Tools

- DE
- Open Zeppelin
- Code Analyzer
- Solidity Code
- Compiler
- Hardhat

VULNERABILITY CHECKS

Design Logic	Passed
Compiler warnings	Passed
Private user data leaks	Passed
Timestamps dependence	Passed
Integer overflow and underflow	Passed
Race conditions & reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front Running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zepplin module	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and acces control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

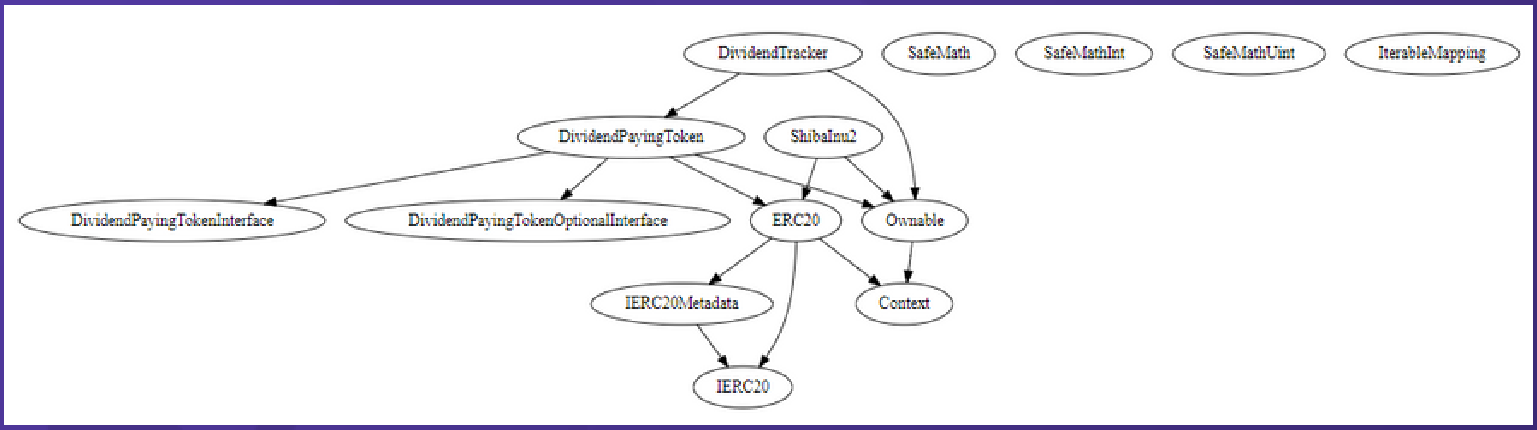
Low Risk

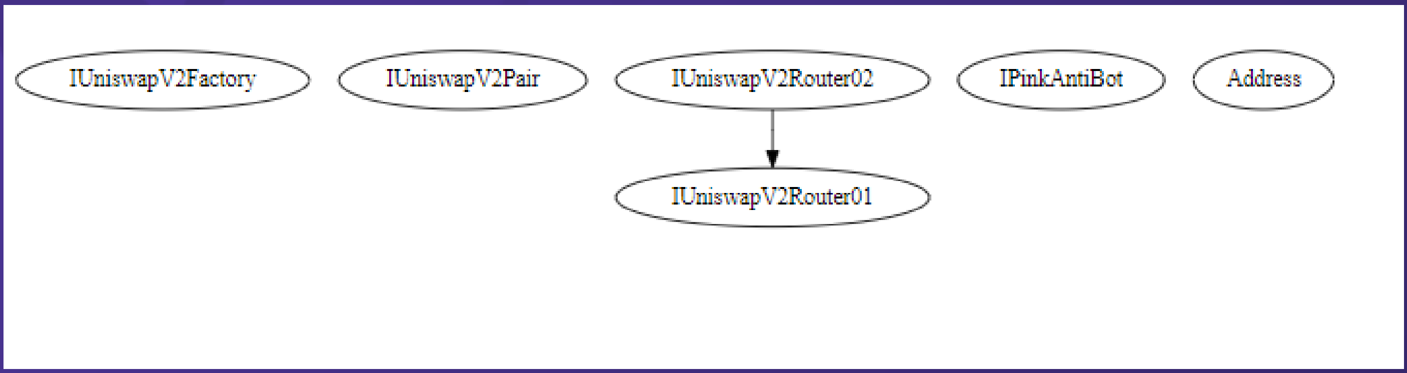
Issues on this level are minor details and warning that can remain unfixed.

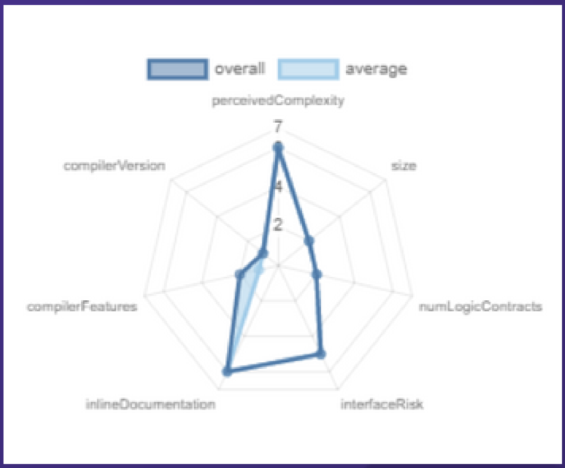
Informational

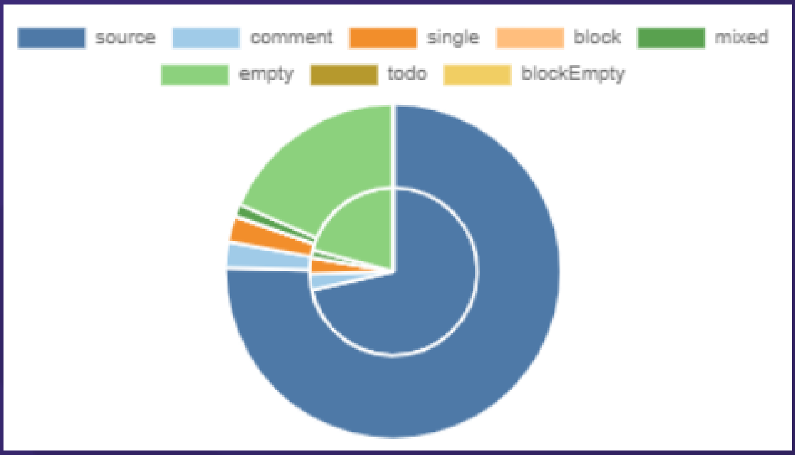
Issues on this level are minor details and warning that can remain unfixed.

INHERITANCE TREES

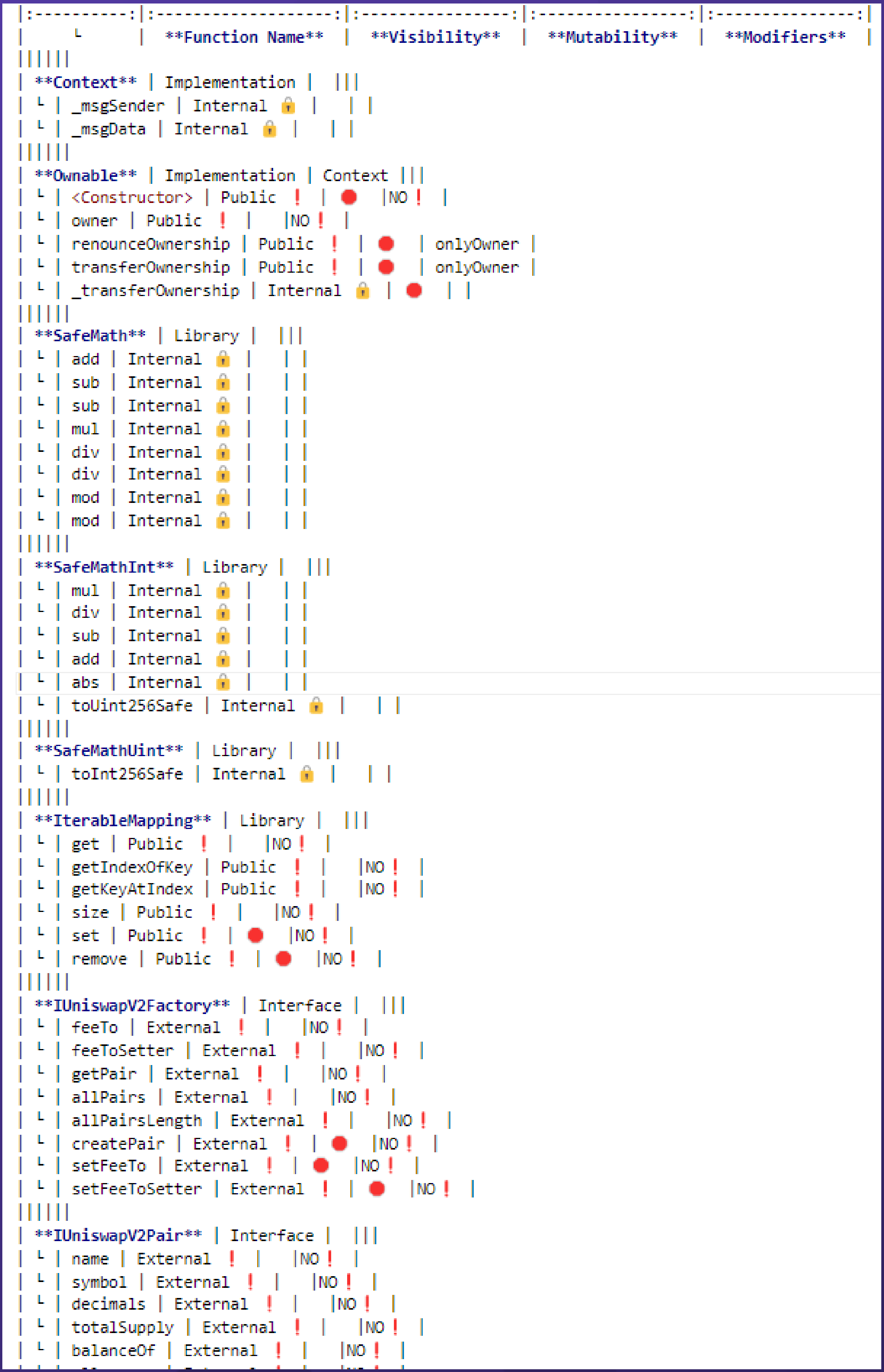






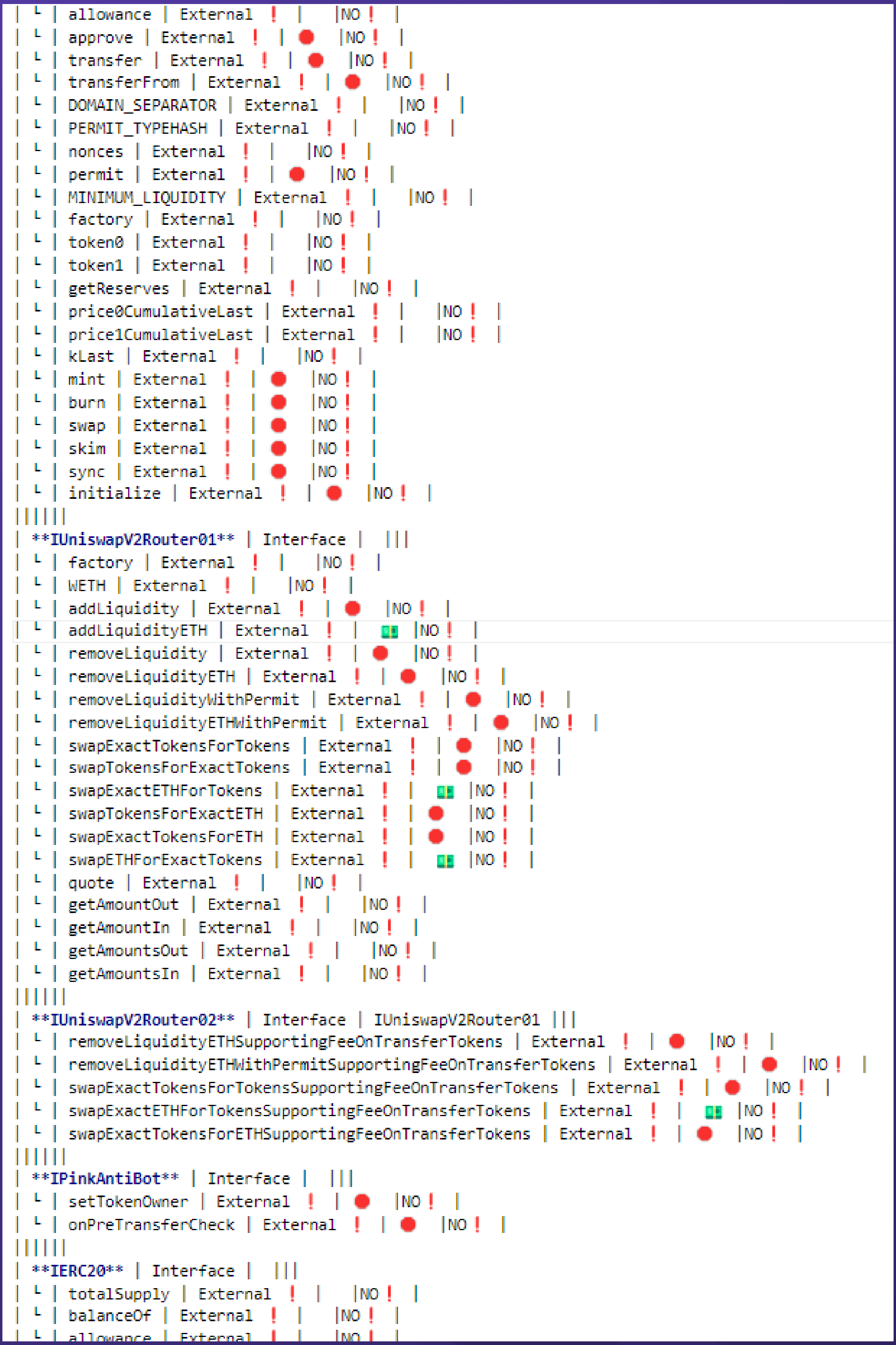


FUNCTION DETAILS



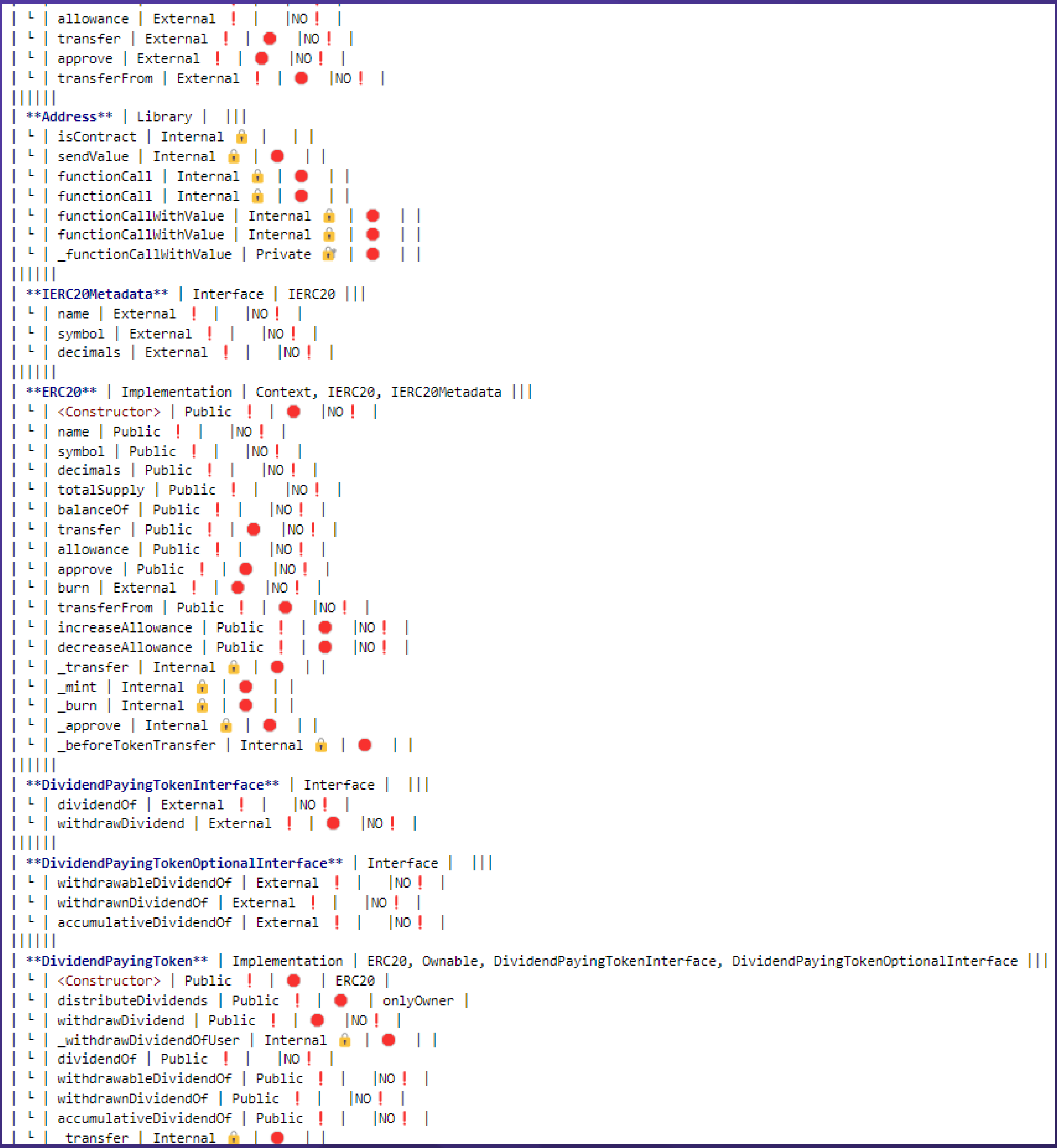
```
|:----------:|:------------------:|:----------------:|:----------------:|:---------------:|
|    └     |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🔴  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | renounceOwnership | Public ❗️ | 🔴  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🔴  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🔴  | |
||||||
| **SafeMath** | Library |  |||
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **SafeMathInt** | Library |  |||
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | add | Internal 🔒 |   | |
| └ | abs | Internal 🔒 |   | |
| └ | toUint256Safe | Internal 🔒 |   | |
||||||
| **SafeMathUint** | Library |  |||
| └ | toInt256Safe | Internal 🔒 |   | |
||||||
| **IterableMapping** | Library |  |||
| └ | get | Public ❗️ |   |NO❗️ |
| └ | getIndexOfKey | Public ❗️ |   |NO❗️ |
| └ | getKeyAtIndex | Public ❗️ |   |NO❗️ |
| └ | size | Public ❗️ |   |NO❗️ |
| └ | set | Public ❗️ | 🔴  |NO❗️ |
| └ | remove | Public ❗️ | 🔴  |NO❗️ |
||||||
| **IUniswapV2Factory** | Interface |  |||
| └ | feeTo | External ❗️ |   |NO❗️ |
| └ | feeToSetter | External ❗️ |   |NO❗️ |
| └ | getPair | External ❗️ |   |NO❗️ |
| └ | allPairs | External ❗️ |   |NO❗️ |
| └ | allPairsLength | External ❗️ |   |NO❗️ |
| └ | createPair | External ❗️ | 🔴  |NO❗️ |
| └ | setFeeTo | External ❗️ | 🔴  |NO❗️ |
| └ | setFeeToSetter | External ❗️ | 🔴  |NO❗️ |
||||||
| **IUniswapV2Pair** | Interface |  |||
| └ | name | External ❗️ |   |NO❗️ |
| └ | symbol | External ❗️ |   |NO❗️ |
| └ | decimals | External ❗️ |   |NO❗️ |
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
```

FUNCTION DETAILS



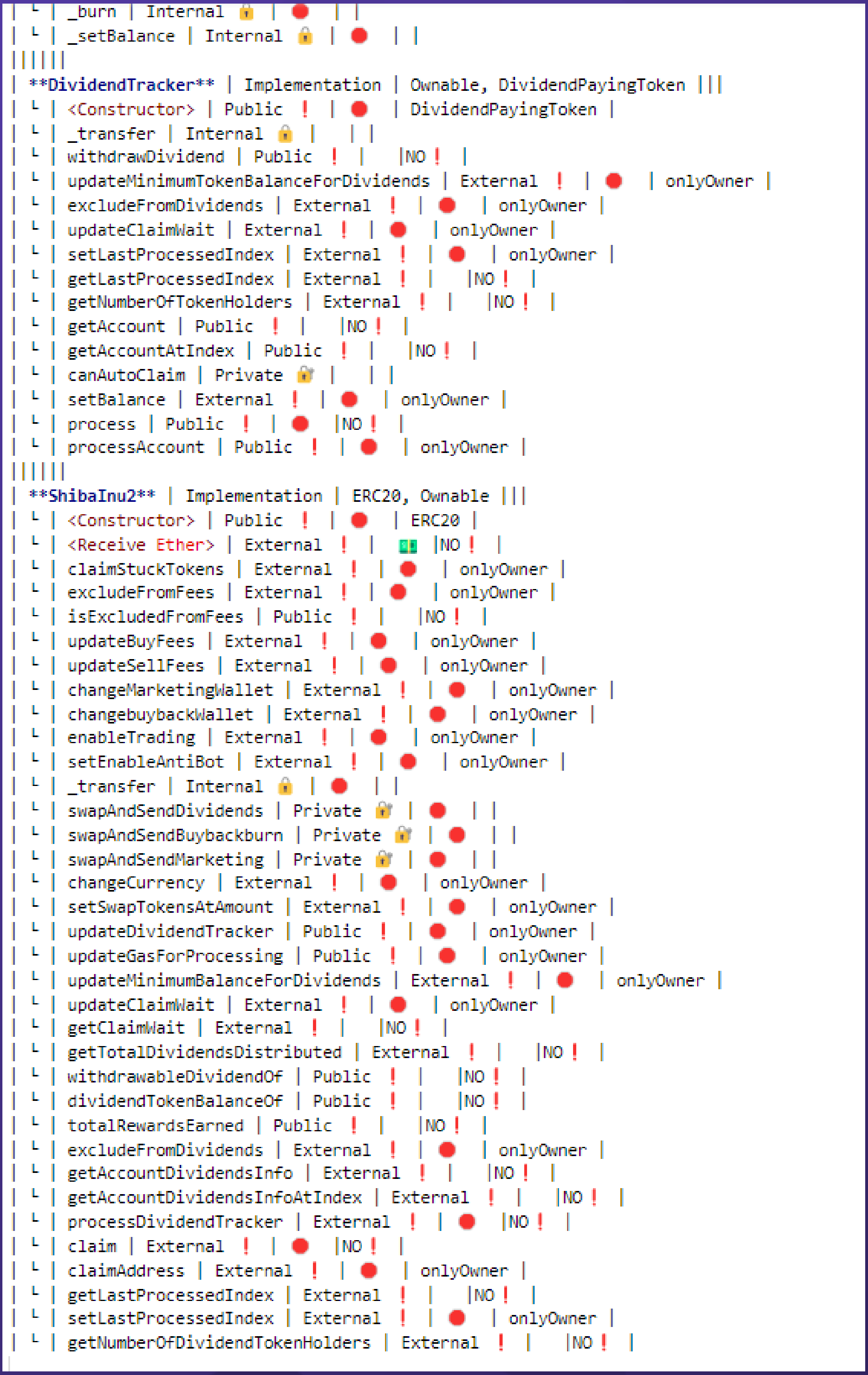
```
| └ | allowance | External ❗ |   |NO❗ |
| └ | approve | External ❗ | 🛑  |NO❗ |
| └ | transfer | External ❗ | 🛑  |NO❗ |
| └ | transferFrom | External ❗ | 🛑  |NO❗ |
| └ | DOMAIN_SEPARATOR | External ❗ |   |NO❗ |
| └ | PERMIT_TYPEHASH | External ❗ |   |NO❗ |
| └ | nonces | External ❗ |   |NO❗ |
| └ | permit | External ❗ | 🛑  |NO❗ |
| └ | MINIMUM_LIQUIDITY | External ❗ |   |NO❗ |
| └ | factory | External ❗ |   |NO❗ |
| └ | token0 | External ❗ |   |NO❗ |
| └ | token1 | External ❗ |   |NO❗ |
| └ | getReserves | External ❗ |   |NO❗ |
| └ | price0CumulativeLast | External ❗ |   |NO❗ |
| └ | price1CumulativeLast | External ❗ |   |NO❗ |
| └ | kLast | External ❗ |   |NO❗ |
| └ | mint | External ❗ | 🛑  |NO❗ |
| └ | burn | External ❗ | 🛑  |NO❗ |
| └ | swap | External ❗ | 🛑  |NO❗ |
| └ | skim | External ❗ | 🛑  |NO❗ |
| └ | sync | External ❗ | 🛑  |NO❗ |
| └ | initialize | External ❗ | 🛑  |NO❗ |
||||||
| **IUniswapV2Router01** | Interface |  |||
| └ | factory | External ❗ |   |NO❗ |
| └ | WETH | External ❗ |   |NO❗ |
| └ | addLiquidity | External ❗ | 🛑  |NO❗ |
| └ | addLiquidityETH | External ❗ |  💵 |NO❗ |
| └ | removeLiquidity | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityETH | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityWithPermit | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityETHWithPermit | External ❗ | 🛑  |NO❗ |
| └ | swapExactTokensForTokens | External ❗ | 🛑  |NO❗ |
| └ | swapTokensForExactTokens | External ❗ | 🛑  |NO❗ |
| └ | swapExactETHForTokens | External ❗ |  💵 |NO❗ |
| └ | swapTokensForExactETH | External ❗ | 🛑  |NO❗ |
| └ | swapExactTokensForETH | External ❗ | 🛑  |NO❗ |
| └ | swapETHForExactTokens | External ❗ |  💵 |NO❗ |
| └ | quote | External ❗ |   |NO❗ |
| └ | getAmountOut | External ❗ |   |NO❗ |
| └ | getAmountIn | External ❗ |   |NO❗ |
| └ | getAmountsOut | External ❗ |   |NO❗ |
| └ | getAmountsIn | External ❗ |   |NO❗ |
||||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ |  💵 |NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO❗ |
||||||
| **IPinkAntiBot** | Interface |  |||
| └ | setTokenOwner | External ❗ | 🛑  |NO❗ |
| └ | onPreTransferCheck | External ❗ | 🛑  |NO❗ |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗ |   |NO❗ |
| └ | balanceOf | External ❗ |   |NO❗ |
| └ | allowance | External ❗ |   |NO❗ |
```

FUNCTION DETAILS



```
| └ | allowance | External ❗ |   |NO❗ |
| └ | transfer | External ❗ | 🛑  |NO❗ |
| └ | approve | External ❗ | 🛑  |NO❗ |
| └ | transferFrom | External ❗ | 🛑  |NO❗ |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | _functionCallWithValue | Private 🔐 | 🛑  | |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
| └ | name | External ❗ |   |NO❗ |
| └ | symbol | External ❗ |   |NO❗ |
| └ | decimals | External ❗ |   |NO❗ |
||||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | name | Public ❗ |   |NO❗ |
| └ | symbol | Public ❗ |   |NO❗ |
| └ | decimals | Public ❗ |   |NO❗ |
| └ | totalSupply | Public ❗ |   |NO❗ |
| └ | balanceOf | Public ❗ |   |NO❗ |
| └ | transfer | Public ❗ | 🛑  |NO❗ |
| └ | allowance | Public ❗ |   |NO❗ |
| └ | approve | Public ❗ | 🛑  |NO❗ |
| └ | burn | External ❗ | 🛑  |NO❗ |
| └ | transferFrom | Public ❗ | 🛑  |NO❗ |
| └ | increaseAllowance | Public ❗ | 🛑  |NO❗ |
| └ | decreaseAllowance | Public ❗ | 🛑  |NO❗ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **DividendPayingTokenInterface** | Interface |  |||
| └ | dividendOf | External ❗ |   |NO❗ |
| └ | withdrawDividend | External ❗ | 🛑  |NO❗ |
||||||
| **DividendPayingTokenOptionalInterface** | Interface |  |||
| └ | withdrawableDividendOf | External ❗ |   |NO❗ |
| └ | withdrawnDividendOf | External ❗ |   |NO❗ |
| └ | accumulativeDividendOf | External ❗ |   |NO❗ |
||||||
| **DividendPayingToken** | Implementation | ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface |||
| └ | <Constructor> | Public ❗ | 🛑  | ERC20 |
| └ | distributeDividends | Public ❗ | 🛑  | onlyOwner |
| └ | withdrawDividend | Public ❗ | 🛑  |NO❗ |
| └ | _withdrawDividendOfUser | Internal 🔒 | 🛑  | |
| └ | dividendOf | Public ❗ |   |NO❗ |
| └ | withdrawableDividendOf | Public ❗ |   |NO❗ |
| └ | withdrawnDividendOf | Public ❗ |   |NO❗ |
| └ | accumulativeDividendOf | Public ❗ |   |NO❗ |
| └ | _transfer | Internal 🔒 | 🛑  | |
```

FUNCTION DETAILS



```
| └ | _burn | Internal 🔒 | 🛑 | |
| └ | _setBalance | Internal 🔒 | 🛑 | |
||||||
| **DividendTracker** | Implementation | Ownable, DividendPayingToken |||
| └ | <Constructor> | Public ❗ | 🛑 | DividendPayingToken |
| └ | _transfer | Internal 🔒 | | |
| └ | withdrawDividend | Public ❗ | |NO❗ |
| └ | updateMinimumTokenBalanceForDividends | External ❗ | 🛑 | onlyOwner |
| └ | excludeFromDividends | External ❗ | 🛑 | onlyOwner |
| └ | updateClaimWait | External ❗ | 🛑 | onlyOwner |
| └ | setLastProcessedIndex | External ❗ | 🛑 | onlyOwner |
| └ | getLastProcessedIndex | External ❗ | |NO❗ |
| └ | getNumberOfTokenHolders | External ❗ | |NO❗ |
| └ | getAccount | Public ❗ | |NO❗ |
| └ | getAccountAtIndex | Public ❗ | |NO❗ |
| └ | canAutoClaim | Private 🔐 | | |
| └ | setBalance | External ❗ | 🛑 | onlyOwner |
| └ | process | Public ❗ | 🛑 |NO❗ |
| └ | processAccount | Public ❗ | 🛑 | onlyOwner |
||||||
| **ShibaInu2** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗ | 🛑 | ERC20 |
| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |
| └ | claimStuckTokens | External ❗ | 🛑 | onlyOwner |
| └ | excludeFromFees | External ❗ | 🛑 | onlyOwner |
| └ | isExcludedFromFees | Public ❗ | |NO❗ |
| └ | updateBuyFees | External ❗ | 🛑 | onlyOwner |
| └ | updateSellFees | External ❗ | 🛑 | onlyOwner |
| └ | changeMarketingWallet | External ❗ | 🛑 | onlyOwner |
| └ | changebuybackWallet | External ❗ | 🛑 | onlyOwner |
| └ | enableTrading | External ❗ | 🛑 | onlyOwner |
| └ | setEnableAntiBot | External ❗ | 🛑 | onlyOwner |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | swapAndSendDividends | Private 🔐 | 🛑 | |
| └ | swapAndSendBuybackburn | Private 🔐 | 🛑 | |
| └ | swapAndSendMarketing | Private 🔐 | 🛑 | |
| └ | changeCurrency | External ❗ | 🛑 | onlyOwner |
| └ | setSwapTokensAtAmount | External ❗ | 🛑 | onlyOwner |
| └ | updateDividendTracker | Public ❗ | 🛑 | onlyOwner |
| └ | updateGasForProcessing | Public ❗ | 🛑 | onlyOwner |
| └ | updateMinimumBalanceForDividends | External ❗ | 🛑 | onlyOwner |
| └ | updateClaimWait | External ❗ | 🛑 | onlyOwner |
| └ | getClaimWait | External ❗ | |NO❗ |
| └ | getTotalDividendsDistributed | External ❗ | |NO❗ |
| └ | withdrawableDividendOf | Public ❗ | |NO❗ |
| └ | dividendTokenBalanceOf | Public ❗ | |NO❗ |
| └ | totalRewardsEarned | Public ❗ | |NO❗ |
| └ | excludeFromDividends | External ❗ | 🛑 | onlyOwner |
| └ | getAccountDividendsInfo | External ❗ | |NO❗ |
| └ | getAccountDividendsInfoAtIndex | External ❗ | |NO❗ |
| └ | processDividendTracker | External ❗ | 🛑 |NO❗ |
| └ | claim | External ❗ | 🛑 |NO❗ |
| └ | claimAddress | External ❗ | 🛑 | onlyOwner |
| └ | getLastProcessedIndex | External ❗ | |NO❗ |
| └ | setLastProcessedIndex | External ❗ | 🛑 | onlyOwner |
| └ | getNumberOfDividendTokenHolders | External ❗ | |NO❗ |
```

MANUAL REVIEW

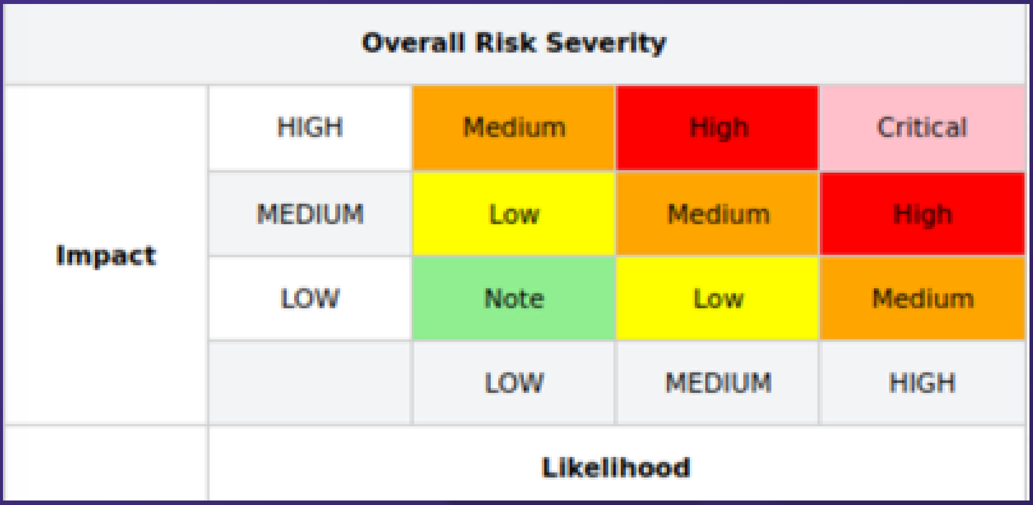
Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to methodology based on OWASP standarts.

Vulnerabilities are dividend into three primary risk categroies:
High
Medium
Low

High-level considerations for vulnerabilities span the following key areas when conducting assessments:

- Malicious input handling
- Escalation of privileges
- Arithmetic
- Gas use



Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

FINDINGS

Findings	Severity	Found
High Risk	● High	0
Medium Risk	● Medium	2
Low Risk	● Low	6
Suggestion & discussion	● Informational	0
Gas Optimizations	● Gas Opt.	0

MEDIUM RISK FINDING

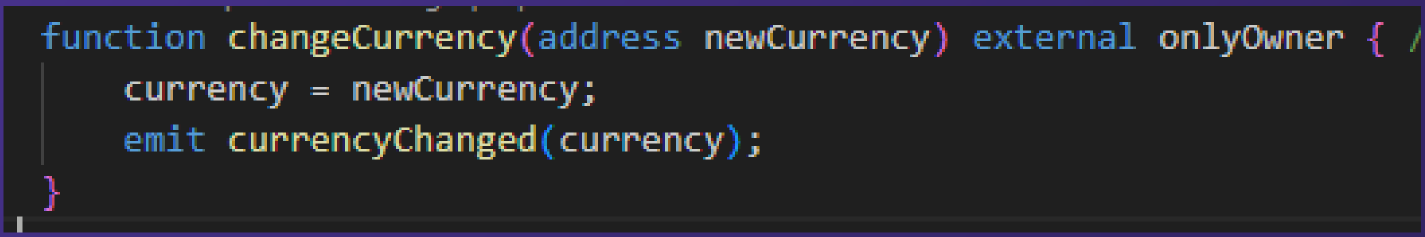
Owner can change currency but not check valid address

Severity : Medium

Overview

The **changeCurrency** function allows the owner of the smart contract to change the currency being used. The swapAndSendBuybackburn and swapAndSendMarketing functions swap ETH for the specified currency using the Uniswap V2 Router and then transfer the resulting tokens to the buybackWallet and marketingWallet, respectively. **But if the owner enters null address or non-valid address instead of newCurreny, no one can sell**

Important notice : Owner himself can't sell if he disable that. Overall this is not Honeypot risk but you should consider that.



```
function changeCurrency(address newCurrency) external onlyOwner { /
    currency = newCurrency;
    emit currencyChanged(currency);
}
```

Recommendation

It is generally considered a good practice to include additional checks to prevent unintended behavior or malicious attacks. A simple check to verify that the new currency is not the zero address could be added to the **changeCurrency** function as an additional safeguard.

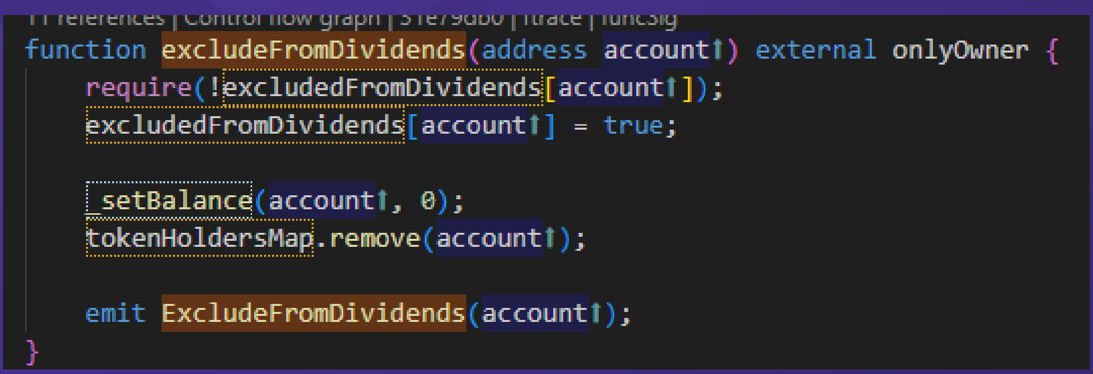
MEDIUM RISK FINDING

Owner can exclude accounts from rewards

Severity : Medium

Overview

Function that allows the owner of the contract to exclude an address from receiving dividends



```
11 references | Control flow graph | 31e79db0 | ftrace | funcsig
function excludeFromDividends(address account) external onlyOwner {
    require(!excludedFromDividends[account]);
    excludedFromDividends[account] = true;

    _setBalance(account, 0);
    tokenHoldersMap.remove(account);

    emit ExcludeFromDividends(account);
}
```

Recommendation

It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions. The contract owner account is well secured and only accessible by authorized parties.

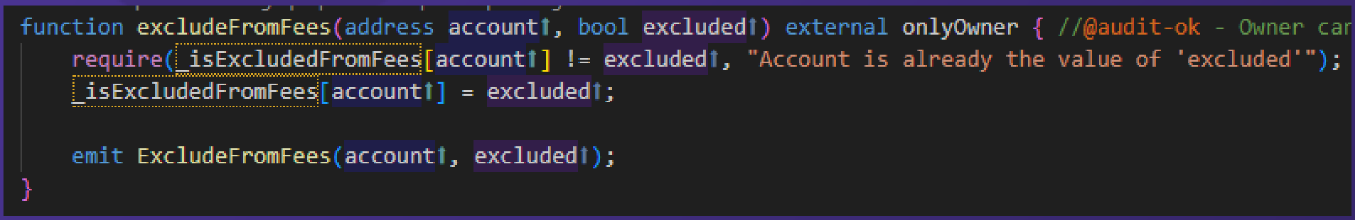
LOW RISK FINDING

Owner can exclude accounts from fees

Severity : Low

Overview

Excludes/Includes an address from the collection of fees



```
function excludeFromFees(address account, bool excluded) external onlyOwner { //@audit-ok - Owner car
    require(_isExcludedFromFees[account] != excluded, "Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

Recommendation

It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions. The contract owner account is well secured and only accessible by authorized parties.

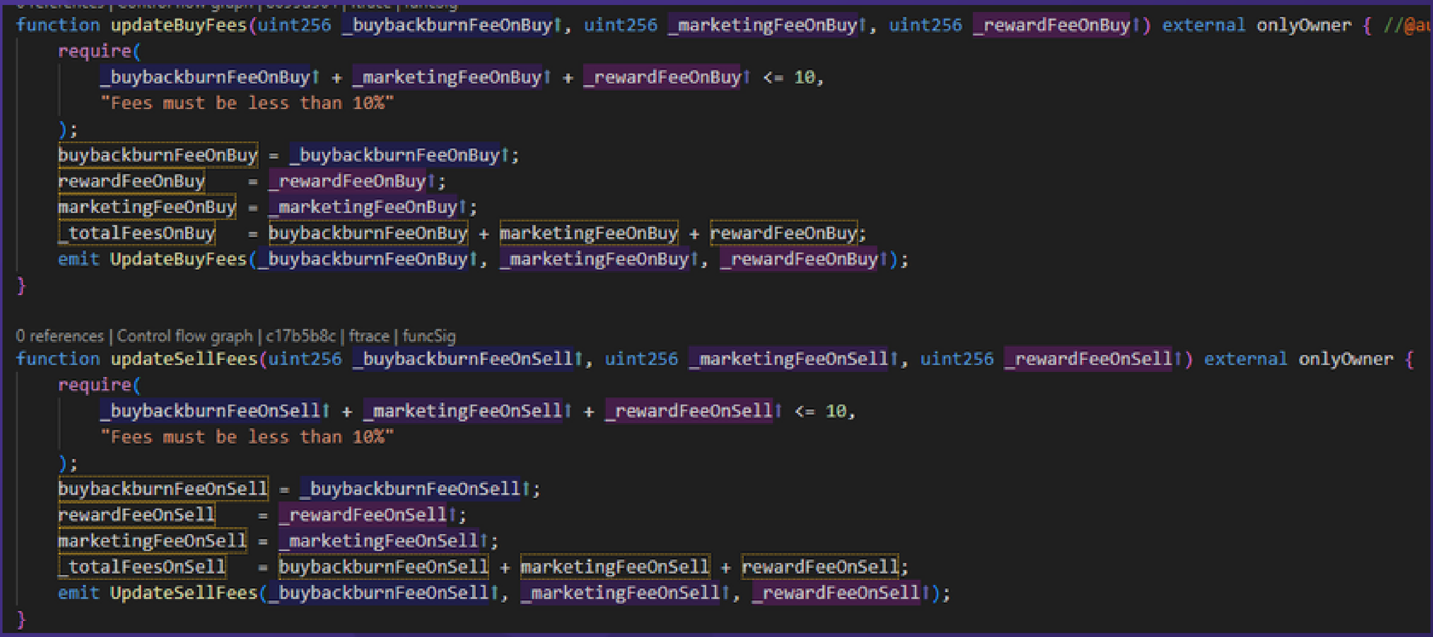
LOW RISK FINDING

Owner can change buy/sell fees with limit up to 10%

Severity : Low

Overview

Functions that allows the owner of the contract to update the buy/sell fees of the contract. These functions assumes that the input parameters are valid and do not exceed the maximum limit of 10%



```
function updateBuyFees(uint256 _buybackburnFeeOnBuy, uint256 _marketingFeeOnBuy, uint256 _rewardFeeOnBuy) external onlyOwner { //@au
    require(
        _buybackburnFeeOnBuy + _marketingFeeOnBuy + _rewardFeeOnBuy <= 10,
        "Fees must be less than 10%"
    );
    buybackburnFeeOnBuy = _buybackburnFeeOnBuy;
    rewardFeeOnBuy      = _rewardFeeOnBuy;
    marketingFeeOnBuy   = _marketingFeeOnBuy;
    _totalFeesOnBuy     = buybackburnFeeOnBuy + marketingFeeOnBuy + rewardFeeOnBuy;
    emit UpdateBuyFees(_buybackburnFeeOnBuy, _marketingFeeOnBuy, _rewardFeeOnBuy);
}

0 references | Control flow graph | c17b5b8c | ftrace | funcSig
function updateSellFees(uint256 _buybackburnFeeOnSell, uint256 _marketingFeeOnSell, uint256 _rewardFeeOnSell) external onlyOwner {
    require(
        _buybackburnFeeOnSell + _marketingFeeOnSell + _rewardFeeOnSell <= 10,
        "Fees must be less than 10%"
    );
    buybackburnFeeOnSell = _buybackburnFeeOnSell;
    rewardFeeOnSell      = _rewardFeeOnSell;
    marketingFeeOnSell   = _marketingFeeOnSell;
    _totalFeesOnSell     = buybackburnFeeOnSell + marketingFeeOnSell + rewardFeeOnSell;
    emit UpdateSellFees(_buybackburnFeeOnSell, _marketingFeeOnSell, _rewardFeeOnSell);
}
```

Recommendation

It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions. The contract owner account is well secured and only accessible by authorized parties.

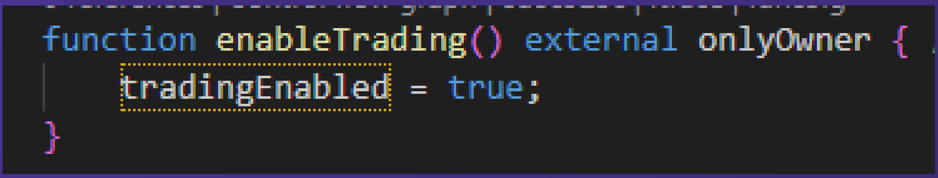
LOW RISK FINDING

Trading must be enabled by the owner

Severity : Low

Overview

Function enables trading by setting the **tradingEnabled** true



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

Recommendation

It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions. The contract owner account is well secured and only accessible by authorized parties.

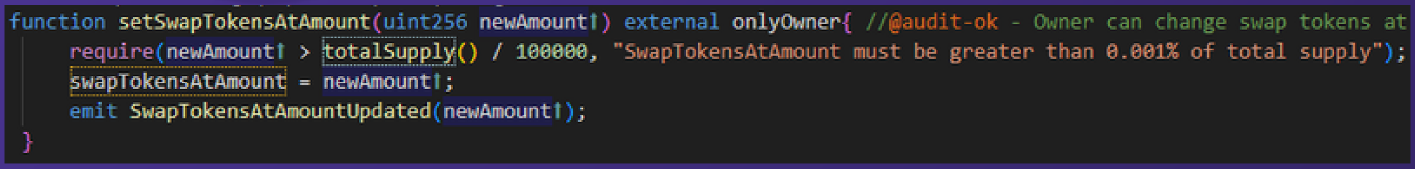
LOW RISK FINDING

Owner can change the swap tokens at amount within reasonable limit

Severity : Low

Overview

setSwapTokensAtAmount function allows the owner to set the minimum number of tokens required to trigger an automatic swap.



```
function setSwapTokensAtAmount(uint256 newAmount↑) external onlyOwner{ //@audit-ok - Owner can change swap tokens at
     require(newAmount↑ > totalSupply() / 100000, "SwapTokensAtAmount must be greater than 0.001% of total supply");
     swapTokensAtAmount = newAmount↑;
     emit SwapTokensAtAmountUpdated(newAmount↑);
 }
```

Recommendation

It's important to ensure that the new **swapTokensAtAmount** value is reasonable and will not adversely affect the functioning of the token or any associated systems.

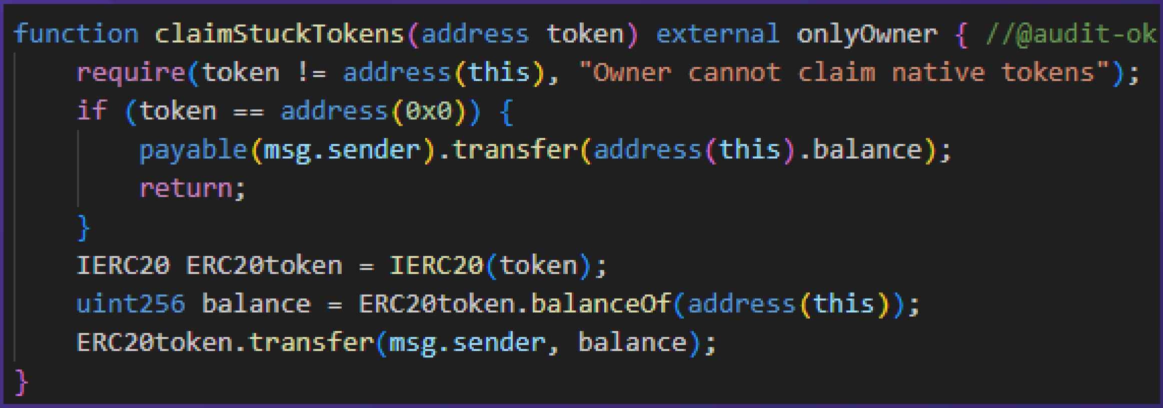
LOW RISK FINDING

Owner can withdraw any token(except native token) from the contract

Severity : Low

Overview

claimStuckTokens function allows the contract owner to recover any ERC20 tokens or BNB that were mistakenly sent to the contract's address. There arerequire statement to prevent the owner from accidentally claiming the native token.



```
function claimStuckTokens(address token) external onlyOwner { //@audit-ok
    require(token != address(this), "Owner cannot claim native tokens");
    if (token == address(0x0)) {
        payable(msg.sender).transfer(address(this).balance);
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

Recommendation

It is generally considered safe for a contract owner to claim stuck tokens, but it's important to ensure that the owner is not abusing this function to steal tokens. In this implementation, there is a require statement that ensures that the **owner cannot claim the native token** of the blockchain on which the contract is deployed.

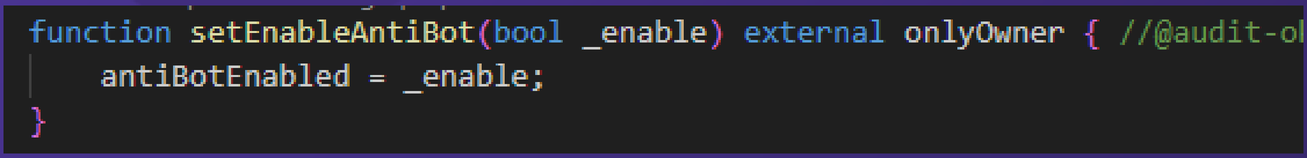
LOW RISK FINDING

Owner can enable pinkantibot system

Severity : Low

Overview

The PinkAntibot smart contract is prevent bot activity in the market by enabling an anti-bot system. The contract includes a function to enable/disable the anti-bot system and a transfer function that includes fees on transactions.



```
function setEnableAntiBot(bool _enable) external onlyOwner { //@audit-ol
    antiBotEnabled = _enable;
}
```

Recommendation

The PinkAntibot contract is a promising with useful features.

ABOUT EXPELEE

Expelee is a product-based aspirational Web3 start-up. Coping up with numerous solutions for blockchain security and constructing a Web3 ecosystem from deal making platform to developer hosting open platform, while also developing our own commercial and sustainable blockchain.

www.expelee.com

expeleeofficial	expelee
Expelee	expelee
expelee_official	expelee-co

Building the Futuristic Blockchain Ecosystem

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 guarantess against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always do your own research and project yourselves from being scammed. The Expelee 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 Expelee receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Alway 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 Expelee team disclaims any liability for the resulting losses.

expelee

Building the Futuristic Blockchain Ecosystem