expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

PRO

TOKEN OVERVIEW

Risk Findings

Severity	Found
● High	1
● Medium	0
● Low	0
● Informational	0

Centralization Risks

Owner Privileges	Description
● Can Owner Set Taxes >25% ?	Not Detected
● Owner needs to enable trading ?	Yes, owner needs to enable trades
● Can Owner Disable Trades ?	Not Detected
● Can Owner Mint ?	Not Detected
● Can Owner Blacklist ?	Not Detected
● Can Owner set Max Wallet amount ?	Not Detected
● Can Owner Set Max TX amount ?	Not Detected

HIGH RISK FINDING

Enabling trades

Category: Centralization
Status: Open
Impact: High

Overview:
Owner must enable trades manually for investors. If trades remain disabled, no one would be able to buy/sell/transfer tokens.

```
function enableTrading() external onlyOwner {
  require(!tradingEnabled, "Trading is already enabled");
  tradingEnabled = true;
  startTradingBlock = block.number;
}
```

Suggestion:
To mitigate this issue, its suggested to :
1- enable trades before presale
2- transfer ownership of the contract to a trusted 3rd party (e.g. pinksale safu developer), this ensures that trades will be enabled at right time
3- implement a more decentralized method for enabling trades at right moment, such as using a cooldown.

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 with high risk
KYC Verification	-
Audit Date	4 August 2023

CONTRACT DETAILS

Token Name: The Pro

Symbol: PRO

Network: Ethereum mainnet

Language: Solidity

Contract Address:
0xD153539F893d549B4Bb6D746F609f6e46EEB8dE8

Total Supply: 1,000,000

Owner's Wallet:
0xCf87FDfEE0A2F85951E54f7c9a69974fa3a85354

Deployer's Wallet:
0xCf87FDfEE0A2F85951E54f7c9a69974fa3a85354

Testnet.
https://testnet.bscscan.com/token/0xb09CA63F320E6e73bCCb0076350eC656acbf1add

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.

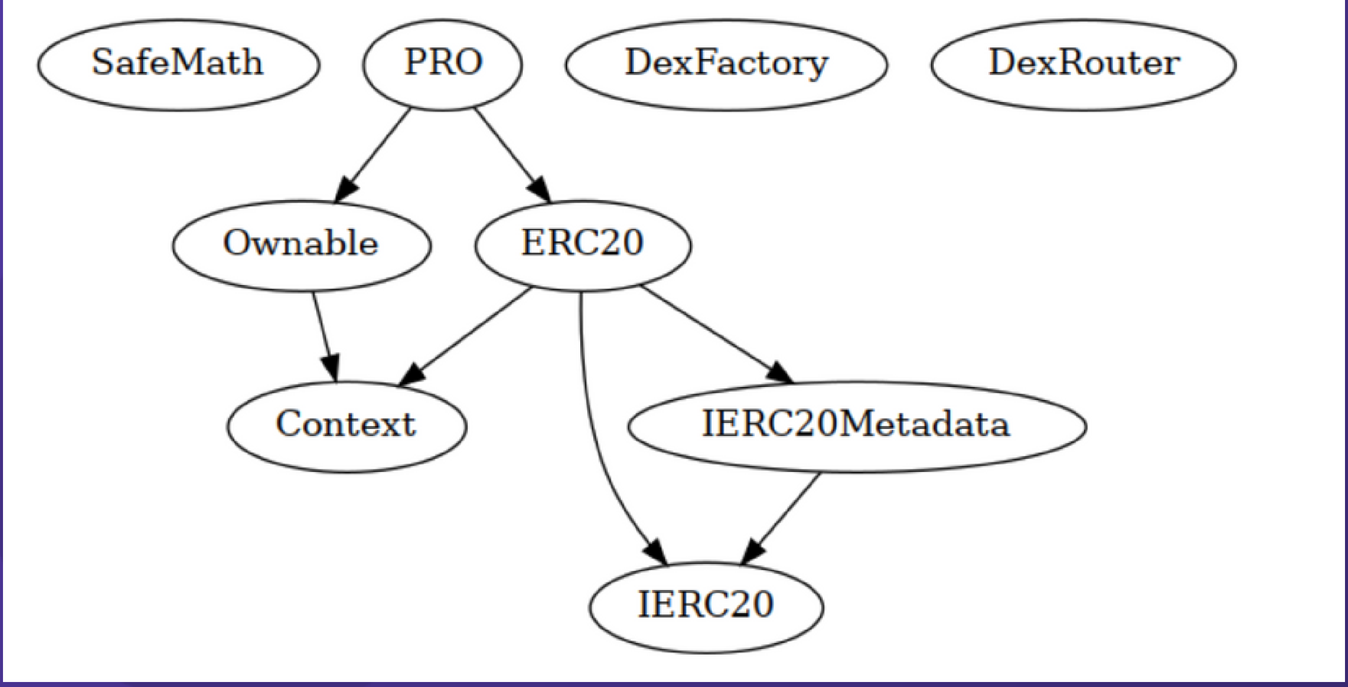
Low Risk

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

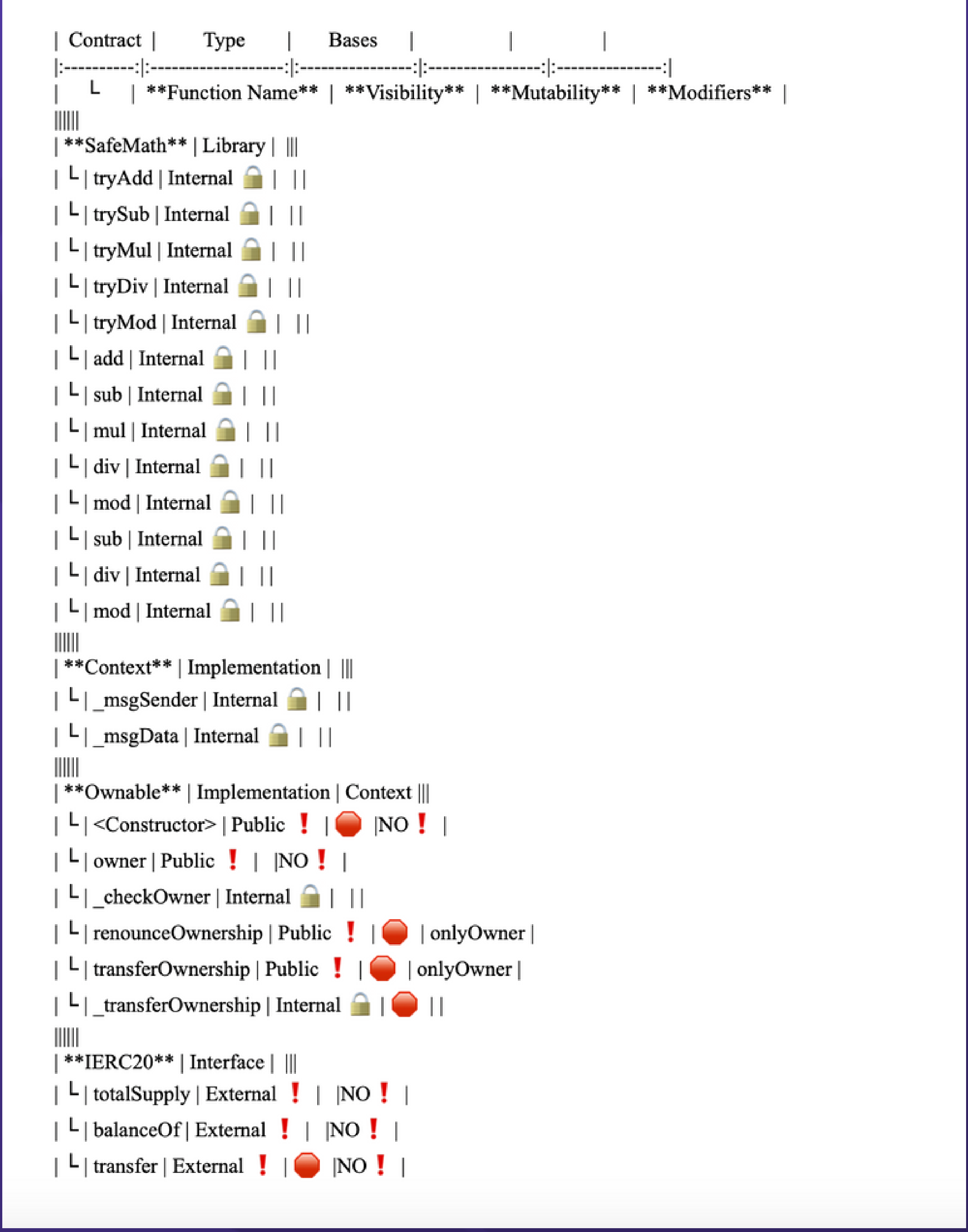
Informational

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

INHERITANCE TREES

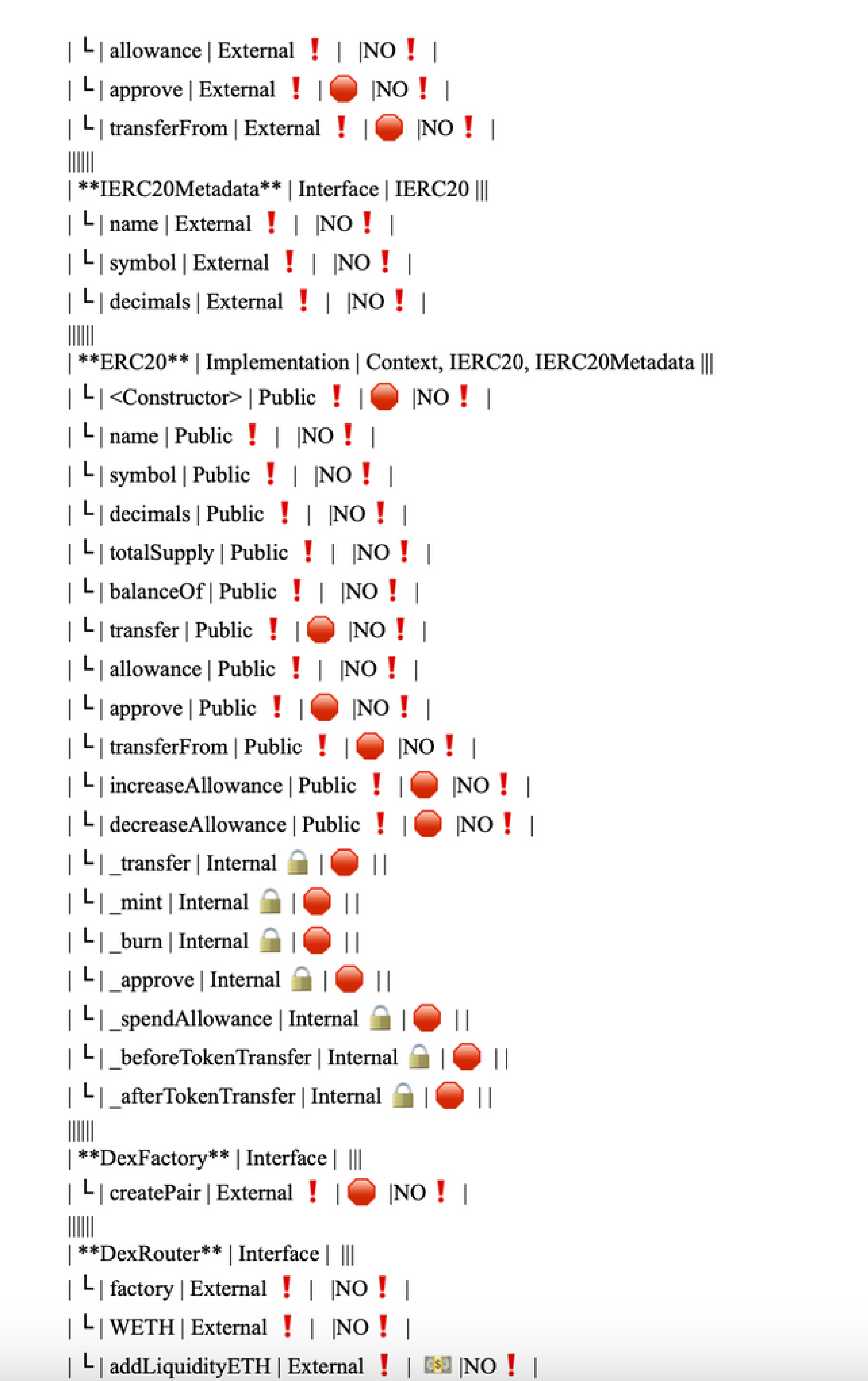


FUNCTION DETAILS



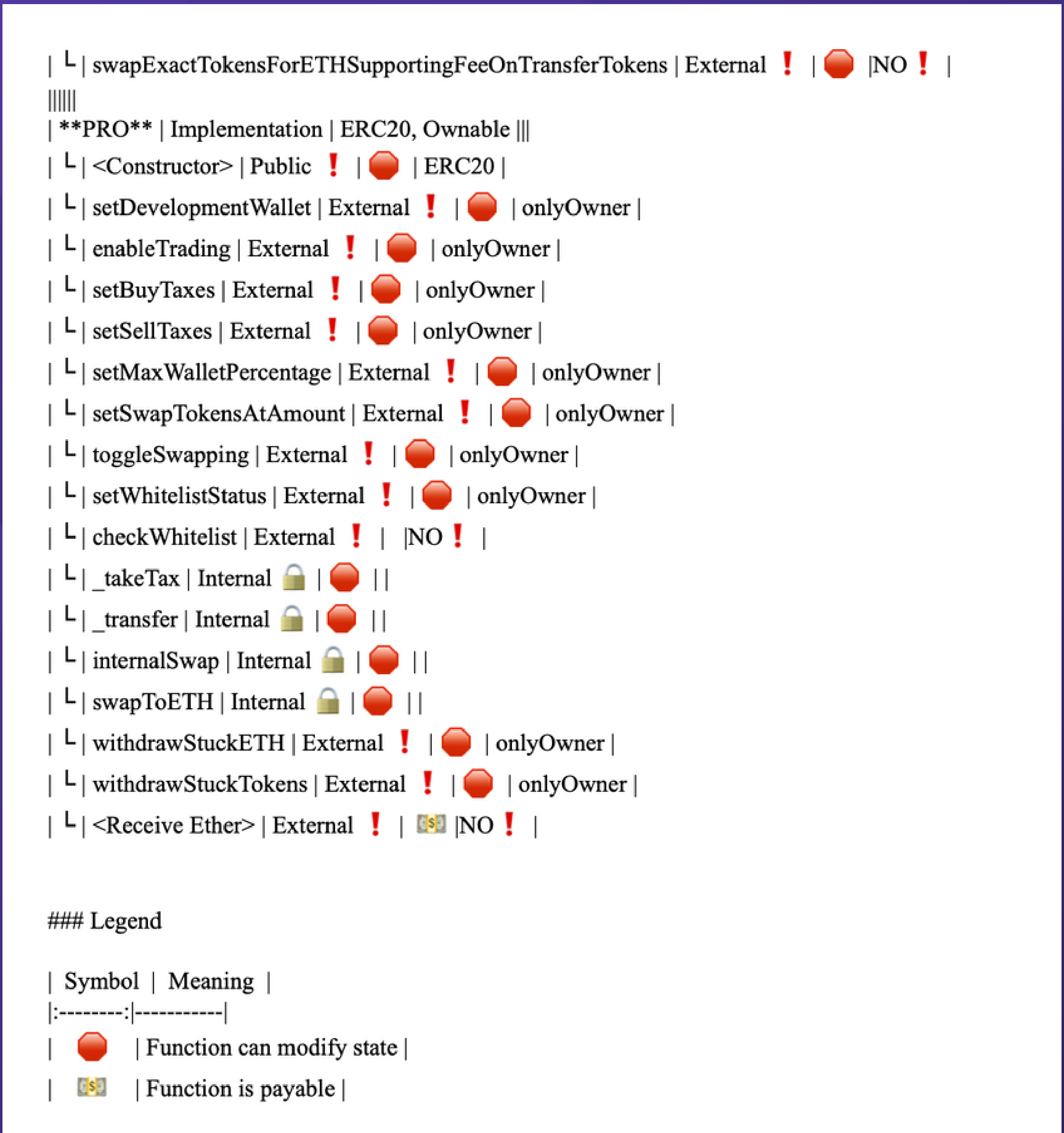
```
| Contract |        Type        |       Bases      |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|    └      | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
||||||
| **SafeMath** | Library |  |||
| └ | tryAdd | Internal 🔒 |   | |
| └ | trySub | Internal 🔒 |   | |
| └ | tryMul | Internal 🔒 |   | |
| └ | tryDiv | Internal 🔒 |   | |
| └ | tryMod | Internal 🔒 |   | |
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | _checkOwner | Internal 🔒 |   | |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
```

FUNCTION DETAILS



```
| └ | allowance | External ❗ |  |NO ❗ |
| └ | approve | External ❗ | 🛑 |NO ❗ |
| └ | transferFrom | External ❗ | 🛑 |NO ❗ |
||||||
| **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 ❗ |
| └ | transferFrom | Public ❗ | 🛑 |NO ❗ |
| └ | increaseAllowance | Public ❗ | 🛑 |NO ❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 |NO ❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | _mint | Internal 🔒 | 🛑 | |
| └ | _burn | Internal 🔒 | 🛑 | |
| └ | _approve | Internal 🔒 | 🛑 | |
| └ | _spendAllowance | Internal 🔒 | 🛑 | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑 | |
||||||
| **DexFactory** | Interface |  |||
| └ | createPair | External ❗ | 🛑 |NO ❗ |
||||||
| **DexRouter** | Interface |  |||
| └ | factory | External ❗ |  |NO ❗ |
| └ | WETH | External ❗ |  |NO ❗ |
| └ | addLiquidityETH | External ❗ | 💵 |NO ❗ |
```

FUNCTION DETAILS



```
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO ❗ |
||||||
| **PRO** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗ | 🛑 | ERC20 |
| └ | setDevelopmentWallet | External ❗ | 🛑 | onlyOwner |
| └ | enableTrading | External ❗ | 🛑 | onlyOwner |
| └ | setBuyTaxes | External ❗ | 🛑 | onlyOwner |
| └ | setSellTaxes | External ❗ | 🛑 | onlyOwner |
| └ | setMaxWalletPercentage | External ❗ | 🛑 | onlyOwner |
| └ | setSwapTokensAtAmount | External ❗ | 🛑 | onlyOwner |
| └ | toggleSwapping | External ❗ | 🛑 | onlyOwner |
| └ | setWhitelistStatus | External ❗ | 🛑 | onlyOwner |
| └ | checkWhitelist | External ❗ |   |NO ❗ |
| └ | _takeTax | Internal 🔐 | 🛑 | |
| └ | _transfer | Internal 🔐 | 🛑 | |
| └ | internalSwap | Internal 🔐 | 🛑 | |
| └ | swapToETH | Internal 🔐 | 🛑 | |
| └ | withdrawStuckETH | External ❗ | 🛑 | onlyOwner |
| └ | withdrawStuckTokens | External ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO ❗ |


### Legend

| Symbol | Meaning |
|:--------:|-----------|
| 🛑 | Function can modify state |
| 💵 | Function is payable |
```

TESTNET VERSION

Adding Liquidity ✓
Tx:
https://testnet.bscscan.com/tx/0x234e6eed7315c197304290c9c5850e7215c6ee9a336719d6906aa76e6ecb18c6z

===

Buying when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x56384b4a12c09f1d123eb4e4498ea95609b55e369115a8afa409ca6fb46d397b

===

Selling when excluded from fees ✓
Tx(0% tax):
https://testnet.bscscan.com/tx/0xf84d11cb9e9978a15872d53a3165f8927345ec9287337d35ca540f216e62a4a2

===

Transferring when excluded from fees ✓
Tx(0% tax):
https://testnet.bscscan.com/tx/0xd87c7d69671b25ae2813bbfa5edf5b1e5292430d19b680701b745d59d304f6e5

===

Buying ✓
Tx(0-10% tax):
https://testnet.bscscan.com/tx/0xf12697e2e7001a4e2ed3b93e87ac602cd3d9209e2519309b2eeba28c423736e3

TESTNET VERSION

Selling ✓
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0x9fbad4b428f246311f930230fc25e8c2d3f5e57160914f28b8f33c0668f93b79

===

Transferring ✓
Tx(0% tax):
https://testnet.bscscan.com/tx/0xdeb5b2037249fc350c151670718e8b0d5f298af486383bc01e2341b663c245a4

===

Internal swap (BNB to marketing wallet | reward token to dividend tracker | reward distribution) ✓
Tx:
https://testnet.bscscan.com/address/0x4d50442757f9f58befb2c43d4f694907a660a298#internaltx

MANUAL REVIEW

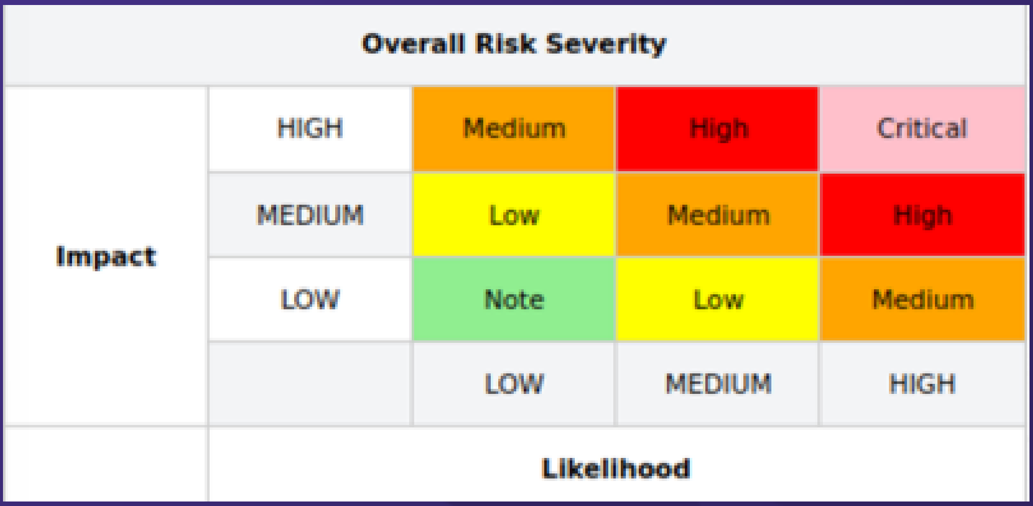
Severity Criteria

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			

HIGH RISK FINDING

Enabling trades

Category: Centralization
Status: Open
Impact: High

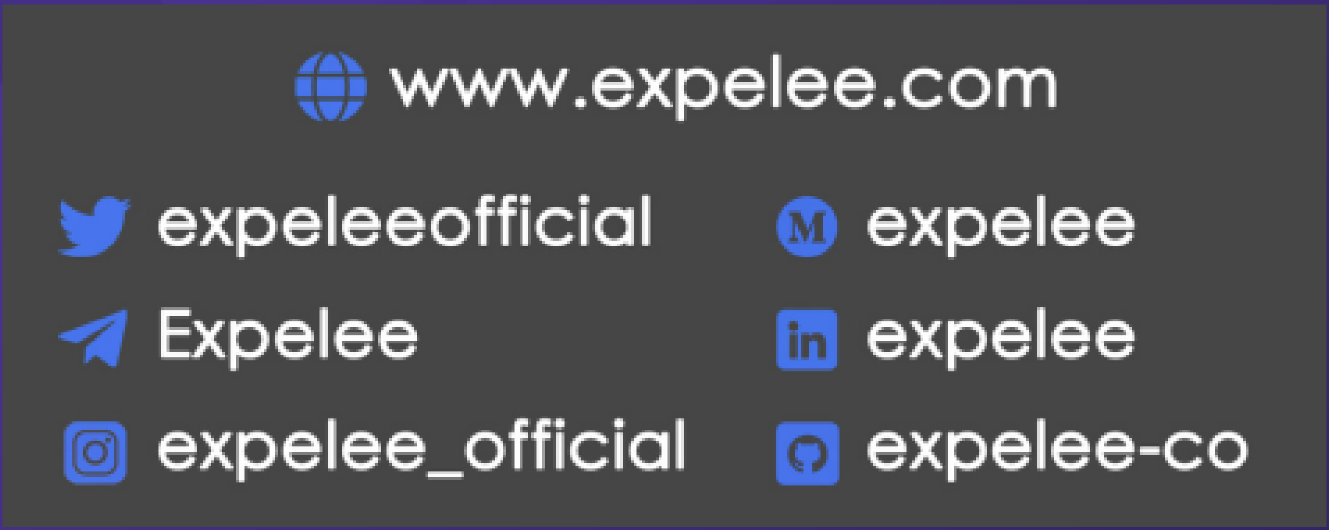
Overview:
Owner must enable trades manually for investors. If trades remain disabled, no one would be able to buy/sell/transfer tokens.

```
function enableTrading() external onlyOwner {
  require(!tradingEnabled, "Trading is already enabled");
  tradingEnabled = true;
  startTradingBlock = block.number;
}
```

Suggestion:
To mitigate this issue, its suggested to :
1- enable trades before presale
2- transfer ownership of the contract to a trusted 3rd party (e.g. pinksale safu developer), this ensures that trades will be enabled at right time
3- implement a more decentralized method for enabling trades at right moment, such as using a cooldown.

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_official

expelee

expelee

expelee-co

expelee

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