expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Baby Wolves Inu

TOKEN OVERVIEW

Risk Findings

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

Centralization Risks

Owner Privileges	Description
● Can Owner Set Taxes >25% ?	Detected
● Owner needs to enable trading ?	Not Detected
● 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

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	**High-Risk Major Flag**
Audit Date	**17 June 2024**

CONTRACT DETAILS

Token Address: 0xC87ebF9261b6d2687901508b1a76E4b2E9C7Baf3

Name: Baby Wolves Inu

Symbol: WOLVES

Decimals: 15

Network: BscScan

Token Type: BEP-20

Owner: 0x97f2404e56A17De6C2E4b622d6dD5468576B1B5C

Deployer: 0x97f2404e56A17De6C2E4b622d6dD5468576B1B5C

Token Supply: 5000000000000000

Checksum: A17acbefe2a12642d388659dffd20311

Testnet:
https://testnet.bscscan.com/address/0xca3ce1e071af49e32c84b6d9e9659f51b9880fa9#code

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

- Manual Review: The code has undergone a line-by-line review by the Ace team.
- 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.
- Slither: The code has undergone static analysis using Slither.

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.

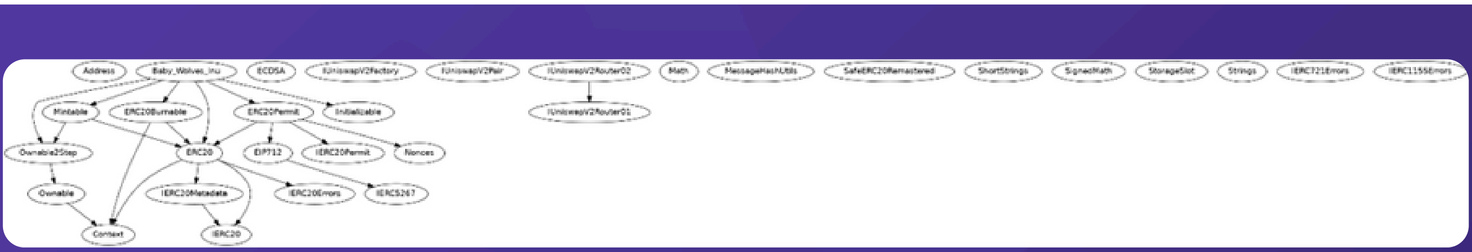
Low Risk

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

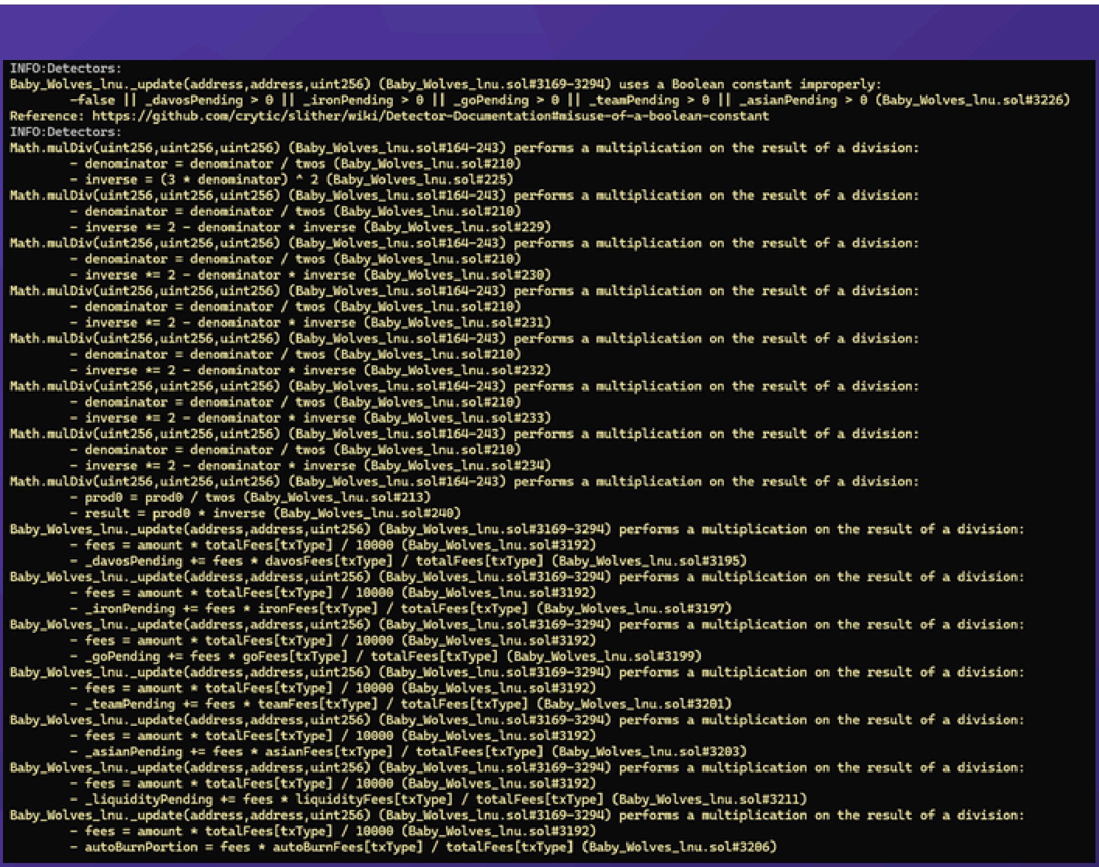
Informational

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

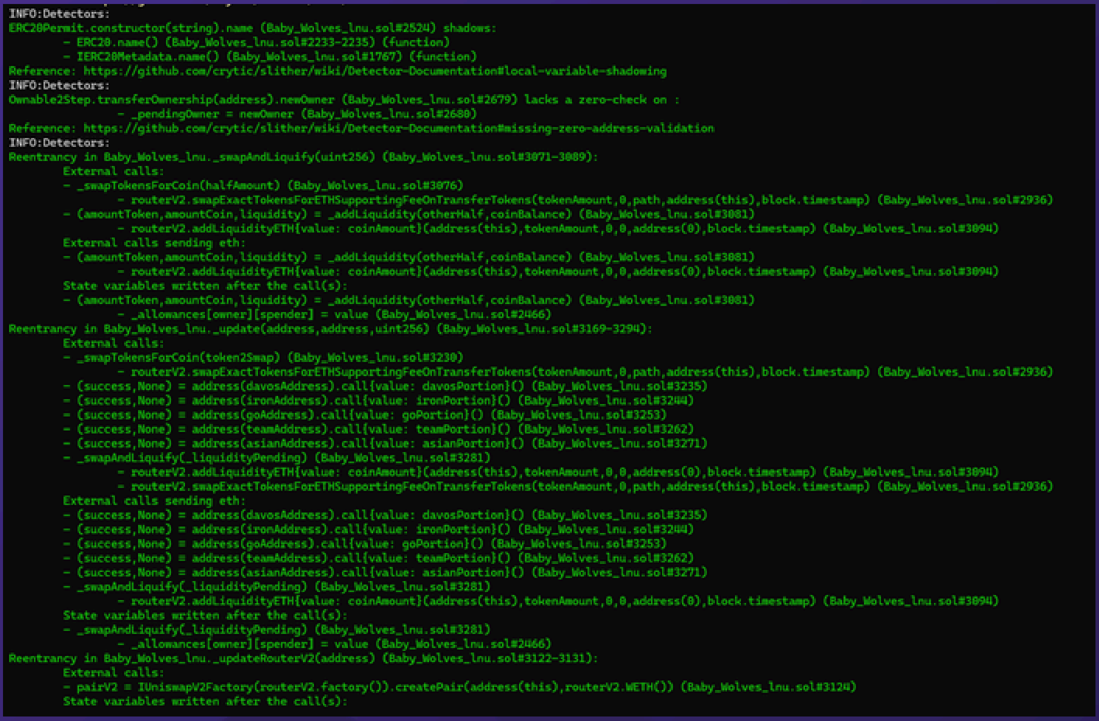
INHERITANCE TREE



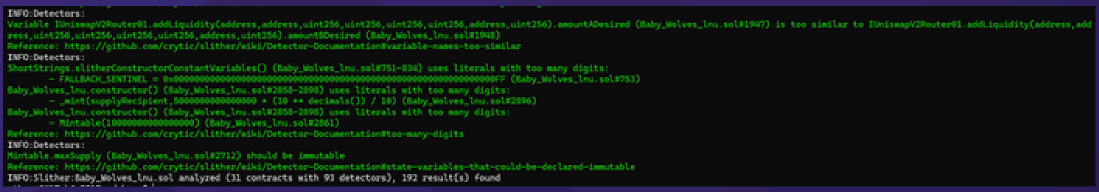
STATIC ANALYSIS



```
INFO:Detectors:
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) uses a Boolean constant improperly:
        -false || _davosPending > 0 || _ironPending > 0 || _goPending > 0 || _teamPending > 0 || _asianPending > 0 (Baby_Wolves_lnu.sol#3226)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#misuse-of-a-boolean-constant
INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse = (3 * denominator) ^ 2 (Baby_Wolves_lnu.sol#225)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse *= 2 - denominator * inverse (Baby_Wolves_lnu.sol#229)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse *= 2 - denominator * inverse (Baby_Wolves_lnu.sol#230)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse *= 2 - denominator * inverse (Baby_Wolves_lnu.sol#231)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse *= 2 - denominator * inverse (Baby_Wolves_lnu.sol#232)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse *= 2 - denominator * inverse (Baby_Wolves_lnu.sol#233)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - denominator = denominator / twos (Baby_Wolves_lnu.sol#210)
        - inverse *= 2 - denominator * inverse (Baby_Wolves_lnu.sol#234)
Math.mulDiv(uint256,uint256,uint256) (Baby_Wolves_lnu.sol#164-243) performs a multiplication on the result of a division:
        - prod0 = prod0 / twos (Baby_Wolves_lnu.sol#213)
        - result = prod0 * inverse (Baby_Wolves_lnu.sol#240)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - _davosPending += fees * davosFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3195)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - _ironPending += fees * ironFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3197)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - _goPending += fees * goFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3199)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - _teamPending += fees * teamFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3201)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - _asianPending += fees * asianFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3203)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - _liquidityPending += fees * liquidityFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3211)
Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (Baby_Wolves_lnu.sol#3192)
        - autoBurnPortion = fees * autoBurnFees[txType] / totalFees[txType] (Baby_Wolves_lnu.sol#3206)
```



```
INFO:Detectors:
ERC20Permit.constructor(string).name (Baby_Wolves_lnu.sol#2524) shadows:
        - ERC20.name() (Baby_Wolves_lnu.sol#2233-2235) (function)
        - IERC20Metadata.name() (Baby_Wolves_lnu.sol#1767) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Ownable2Step.transferOwnership(address).newOwner (Baby_Wolves_lnu.sol#2679) lacks a zero-check on :
                - _pendingOwner = newOwner (Baby_Wolves_lnu.sol#2680)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in Baby_Wolves_lnu._swapAndLiquify(uint256) (Baby_Wolves_lnu.sol#3071-3089):
        External calls:
        - _swapTokensForCoin(halfAmount) (Baby_Wolves_lnu.sol#3076)
                - routerV2.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (Baby_Wolves_lnu.sol#2936)
        - (amountToken,amountCoin,liquidity) = _addLiquidity(otherHalf,coinBalance) (Baby_Wolves_lnu.sol#3081)
                - routerV2.addLiquidityETH{value: coinAmount}(address(this),tokenAmount,0,0,address(0),block.timestamp) (Baby_Wolves_lnu.sol#3094)
        External calls sending eth:
        - (amountToken,amountCoin,liquidity) = _addLiquidity(otherHalf,coinBalance) (Baby_Wolves_lnu.sol#3081)
                - routerV2.addLiquidityETH{value: coinAmount}(address(this),tokenAmount,0,0,address(0),block.timestamp) (Baby_Wolves_lnu.sol#3094)
        State variables written after the call(s):
        - (amountToken,amountCoin,liquidity) = _addLiquidity(otherHalf,coinBalance) (Baby_Wolves_lnu.sol#3081)
                - _allowances[owner][spender] = value (Baby_Wolves_lnu.sol#2466)
Reentrancy in Baby_Wolves_lnu._update(address,address,uint256) (Baby_Wolves_lnu.sol#3169-3294):
        External calls:
        - _swapTokensForCoin(token2Swap) (Baby_Wolves_lnu.sol#3230)
                - routerV2.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (Baby_Wolves_lnu.sol#2936)
        - (success,None) = address(davosAddress).call{value: davosPortion}() (Baby_Wolves_lnu.sol#3235)
        - (success,None) = address(ironAddress).call{value: ironPortion}() (Baby_Wolves_lnu.sol#3244)
        - (success,None) = address(goAddress).call{value: goPortion}() (Baby_Wolves_lnu.sol#3253)
        - (success,None) = address(teamAddress).call{value: teamPortion}() (Baby_Wolves_lnu.sol#3262)
        - (success,None) = address(asianAddress).call{value: asianPortion}() (Baby_Wolves_lnu.sol#3271)
        - _swapAndLiquify(_liquidityPending) (Baby_Wolves_lnu.sol#3281)
                - routerV2.addLiquidityETH{value: coinAmount}(address(this),tokenAmount,0,0,address(0),block.timestamp) (Baby_Wolves_lnu.sol#3094)
                - routerV2.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (Baby_Wolves_lnu.sol#2936)
        External calls sending eth:
        - (success,None) = address(davosAddress).call{value: davosPortion}() (Baby_Wolves_lnu.sol#3235)
        - (success,None) = address(ironAddress).call{value: ironPortion}() (Baby_Wolves_lnu.sol#3244)
        - (success,None) = address(goAddress).call{value: goPortion}() (Baby_Wolves_lnu.sol#3253)
        - (success,None) = address(teamAddress).call{value: teamPortion}() (Baby_Wolves_lnu.sol#3262)
        - (success,None) = address(asianAddress).call{value: asianPortion}() (Baby_Wolves_lnu.sol#3271)
        - _swapAndLiquify(_liquidityPending) (Baby_Wolves_lnu.sol#3281)
                - routerV2.addLiquidityETH{value: coinAmount}(address(this),tokenAmount,0,0,address(0),block.timestamp) (Baby_Wolves_lnu.sol#3094)
        State variables written after the call(s):
        - _swapAndLiquify(_liquidityPending) (Baby_Wolves_lnu.sol#3281)
                - _allowances[owner][spender] = value (Baby_Wolves_lnu.sol#2466)
Reentrancy in Baby_Wolves_lnu._updateRouterV2(address) (Baby_Wolves_lnu.sol#3122-3131):
        External calls:
        - pairV2 = IUniswapV2Factory(routerV2.factory()).createPair(address(this),routerV2.WETH()) (Baby_Wolves_lnu.sol#3124)
        State variables written after the call(s):
```



```
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (Baby_Wolves_lnu.sol#1947) is too similar to IUniswapV2Router01.addLiquidity(address,add
ress,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (Baby_Wolves_lnu.sol#1948)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
ShortStrings.slitherConstructorConstantVariables() (Baby_Wolves_lnu.sol#751-834) uses literals with too many digits:
        - FALLBACK_SENTINEL = 0x00000000000000000000000000000000000000000000000000000000000000FF (Baby_Wolves_lnu.sol#753)
Baby_Wolves_lnu.constructor() (Baby_Wolves_lnu.sol#2858-2898) uses literals with too many digits:
        - _mint(supplyRecipient,5000000000000000 * (10 ** decimals()) / 10) (Baby_Wolves_lnu.sol#2896)
Baby_Wolves_lnu.constructor() (Baby_Wolves_lnu.sol#2858-2898) uses literals with too many digits:
        - Mintable(10000000000000000) (Baby_Wolves_lnu.sol#2861)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
Mintable.maxSupply (Baby_Wolves_lnu.sol#2712) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:Baby_Wolves_lnu.sol analyzed (31 contracts with 93 detectors), 192 result(s) found
```

TESTNET VERSION

1- Approve (passed):
https://testnet.bscscan.com/tx/0xa7c313ff3c08d813b94f0a71ba9c3ceefa734bca7ddb7e62aea494e9b982162d

2- Asian Address Setup (passed):
https://testnet.bscscan.com/tx/0x769f5df2a2cb946de18f82d7009c4e896fa676657f92e65ca9cbc960ce0dba9f

3- Asian Fees Setup (passed):
https://testnet.bscscan.com/tx/0xacbc11b2cbfd6aad8e4667f10ac96078e8b49a8e9c99df9d369e8fc54452a426

4- Devos Address Setup (passed):
https://testnet.bscscan.com/tx/0x0de703321df0df74fdd85740557528df88288f656a5bb07678c8107a59203882

5- Davos Fees Setup (passed):
https://testnet.bscscan.com/tx/0x5fa1acad354c89cbe6a6bc0f9e0475c9a69409a90e82981b73561f6c387a4540

6- Go Address Setup (passed):
https://testnet.bscscan.com/tx/0xc97eca89512547a12dbb2823ed2a34d066e5fb16af773a911978b11e25293c8c

7- Iron Address Setup (passed):
https://testnet.bscscan.com/tx/0xa9dd0a0c54b8146c66efd10fed0b107325f69b2809582f66e1bc238e2cfc5cad

8- Team Address Setup (passed):
https://testnet.bscscan.com/tx/0xba42012310db79d01c201a52095046d89daf15a94883bcdbcf09455629db82ed

MANUAL REVIEW

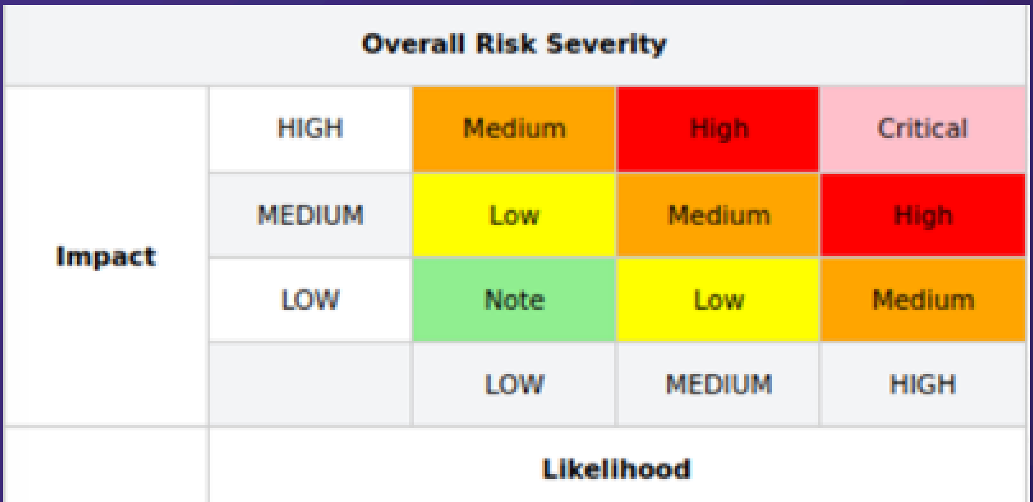
Severity Criteria

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

Centralization – Buy and Sell Fees and transfer

Severity: High

Function: davosFees, ironFees/gofeesSetup/

Status: Open

Overview:

The owner can set the buy and sell fees 100%, which is not recommended.

```
Function davosFeesSetup(uint16 _buyFee, uint16 _sellFee, uint16 _transferFee)
public onlyOwner {
    totalFees[0] = totalFees[0] - davosFees[0] + _buyFee;
    totalFees[1] = totalFees[1] - davosFees[1] + _sellFee;
    totalFees[2] = totalFees[2] - davosFees[2] + _transferFee;
    if (totalFees[0] > 2500 || totalFees[1] > 2500 || totalFees[2] > 2500) revert
CannotExceedMaxTotalFee(totalFees[0], totalFees[1], totalFees[2]);

    davosFees = [_buyFee, _sellFee, _transferFee];

    emit WalletTaxFeesUpdated(1, _buyFee, _sellFee, _transferFee);
  }
function ironFeesSetup(uint16 _buyFee, uint16 _sellFee, uint16 _transferFee) public
onlyOwner {
    totalFees[0] = totalFees[0] - ironFees[0] + _buyFee;
    totalFees[1] = totalFees[1] - ironFees[1] + _sellFee;
    totalFees[2] = totalFees[2] - ironFees[2] + _transferFee;
    if (totalFees[0] > 2500 || totalFees[1] > 2500 || totalFees[2] > 2500) revert
CannotExceedMaxTotalFee(totalFees[0], totalFees[1], totalFees[2]);

    ironFees = [_buyFee, _sellFee, _transferFee];

    emit WalletTaxFeesUpdated(2, _buyFee, _sellFee, _transferFee);
  }
nction goFeesSetup(uint16 _buyFee, uint16 _sellFee, uint16 _transferFee) public
onlyOwner {
    totalFees[0] = totalFees[0] - goFees[0] + _buyFee;
```

HIGH RISK FINDING

```
    totalFees[1] = totalFees[1] - goFees[1] + _sellFee;
    totalFees[2] = totalFees[2] - goFees[2] + _transferFee;
    if (totalFees[0] > 2500 || totalFees[1] > 2500 || totalFees[2] > 2500) revert
CannotExceedMaxTotalFee(totalFees[0], totalFees[1], totalFees[2]);

    goFees = [_buyFee, _sellFee, _transferFee];

    emit WalletTaxFeesUpdated(3, _buyFee, _sellFee, _transferFee);
  }
```

MEDIUM RISK FINDING

Centralization – Missing Require Check.

Severity: Medium

Function: setTreasuryAddress

Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner sets the address to the contract address, then the ETH will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function davosAddressSetup(address _newAddress) public onlyOwner {
   if (_newAddress == address(0)) revert InvalidTaxRecipientAddress(address(0));

   davosAddress = _newAddress;
   excludeFromFees(_newAddress, true);
   _excludeFromLimits(_newAddress, true);

   emit WalletTaxAddressUpdated(1, _newAddress);
  }
function ironAddressSetup(address _newAddress) public onlyOwner {
   if (_newAddress == address(0)) revert InvalidTaxRecipientAddress(address(0));

   ironAddress = _newAddress;
   excludeFromFees(_newAddress, true);
   _excludeFromLimits(_newAddress, true);

   emit WalletTaxAddressUpdated(2, _newAddress);
  }
function goAddressSetup(address _newAddress) public onlyOwner {
   if (_newAddress == address(0)) revert InvalidTaxRecipientAddress(address(0));

   goAddress = _newAddress;
   excludeFromFees(_newAddress, true);
   _excludeFromLimits(_newAddress, true);
```

MEDIUM RISK FINDING

```
    emit WalletTaxAddressUpdated(3, _newAddress);
  }

 function teamAddressSetup(address _newAddress) public onlyOwner {
    if (_newAddress == address(0)) revert InvalidTaxRecipientAddress(address(0));

    teamAddress = _newAddress;
    excludeFromFees(_newAddress, true);
    _excludeFromLimits(_newAddress, true);

    emit WalletTaxAddressUpdated(4, _newAddress);
  }
function asianAddressSetup(address _newAddress) public onlyOwner {
    if (_newAddress == address(0)) revert InvalidTaxRecipientAddress(address(0));

    asianAddress = _newAddress;
    excludeFromFees(_newAddress, true);
    _excludeFromLimits(_newAddress, true);

    emit WalletTaxAddressUpdated(5, _newAddress);
  }
```

Suggestion:
It is recommended that the address should not be able to be set as a contract address.

INFORMATIONAL FINDINGS

Optimization

Severity: Informational

Subject: Floating Pragma Solidity version

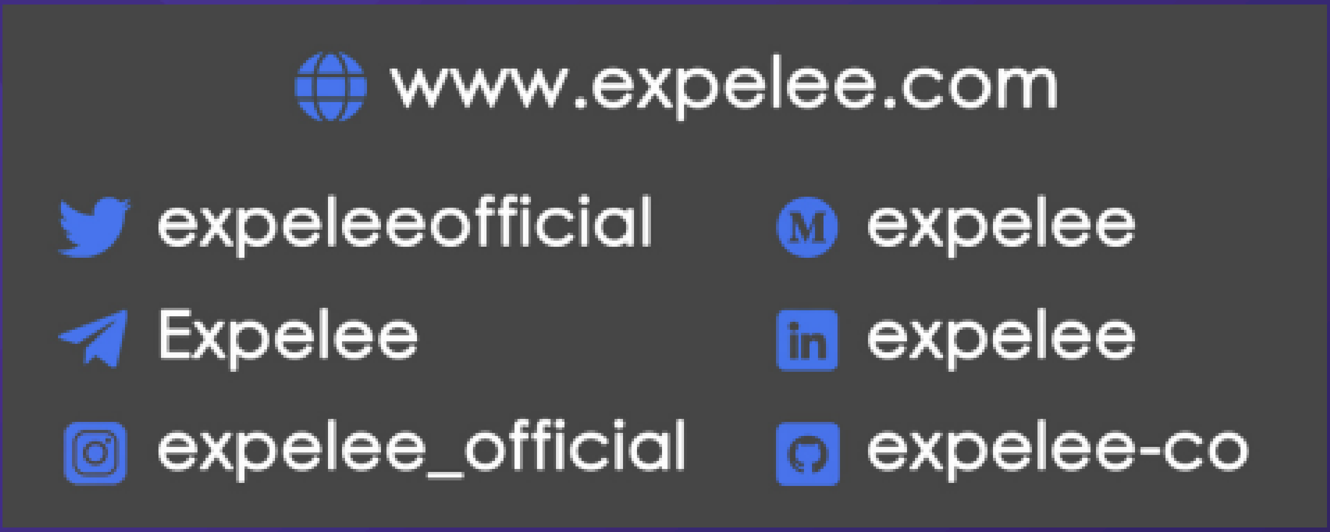
Status: Open

Overview:
It is considered best practice to pick one compiler version and stick with it. With a floating pragma, contracts may accidentally be deployed using an outdated.
pragma solidity ^0.8.20;

Suggestion:
Adding the latest constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.

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

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