expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

ORION

TOKEN OVERVIEW

Risk Findings

Severity	Found
🔴 High	0
🟠 Medium	0
🟡 Low	1
🔵 Informational	2

Centralization Risks

Owner Privileges	Description
🟢 Can Owner Set Taxes >25% ?	Not 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	Passed
KYC Verification	-
Audit Date	27 March 2024

CONTRACT DETAILS

Token Address: 0x1d3032FBeaF715232c8A02f3453a94E92AFb95C1

Name: ORION

Symbol: ORI

Decimals: 18

Network: BscScan

Token Type: BEP-20

Owner: 0x6AfB3cC3EB10E4ABcd45c659Bb2b6a91A3A4d450

Deployer: 0x6AfB3cC3EB10E4ABcd45c659Bb2b6a91A3A4d450

Token Supply: 100,000,000,000

Checksum: A9032c616934aeb47e6039f76b20d2e4

Testnet:
https://testnet.bscscan.com/address/0xc0a98a6495b78d1bc4d9aa1a68e134994514f899#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

- 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.

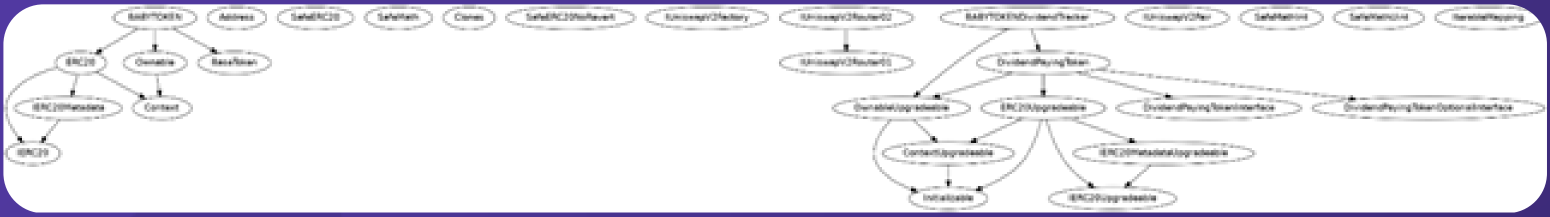
Low Risk

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

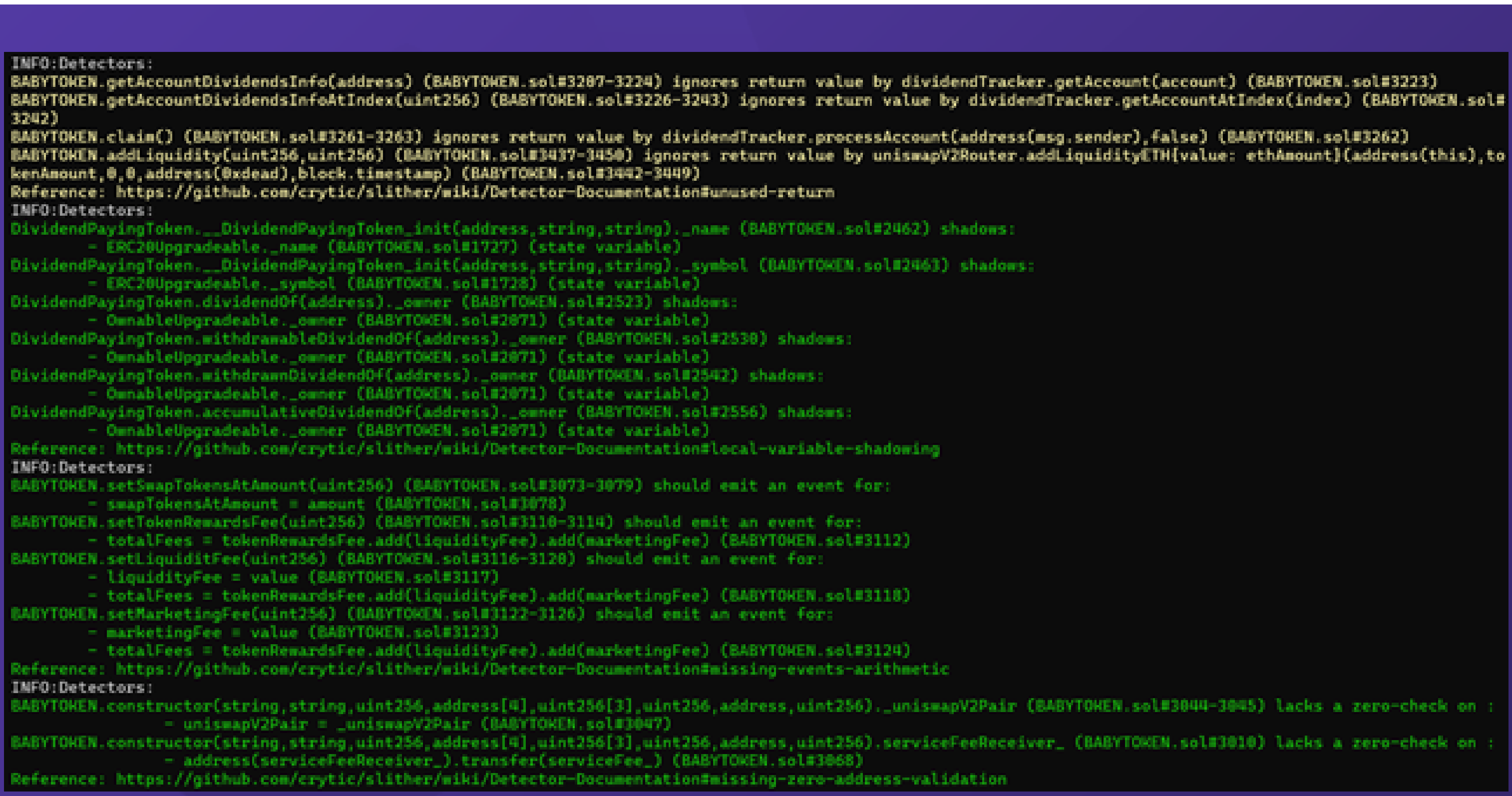
Informational

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

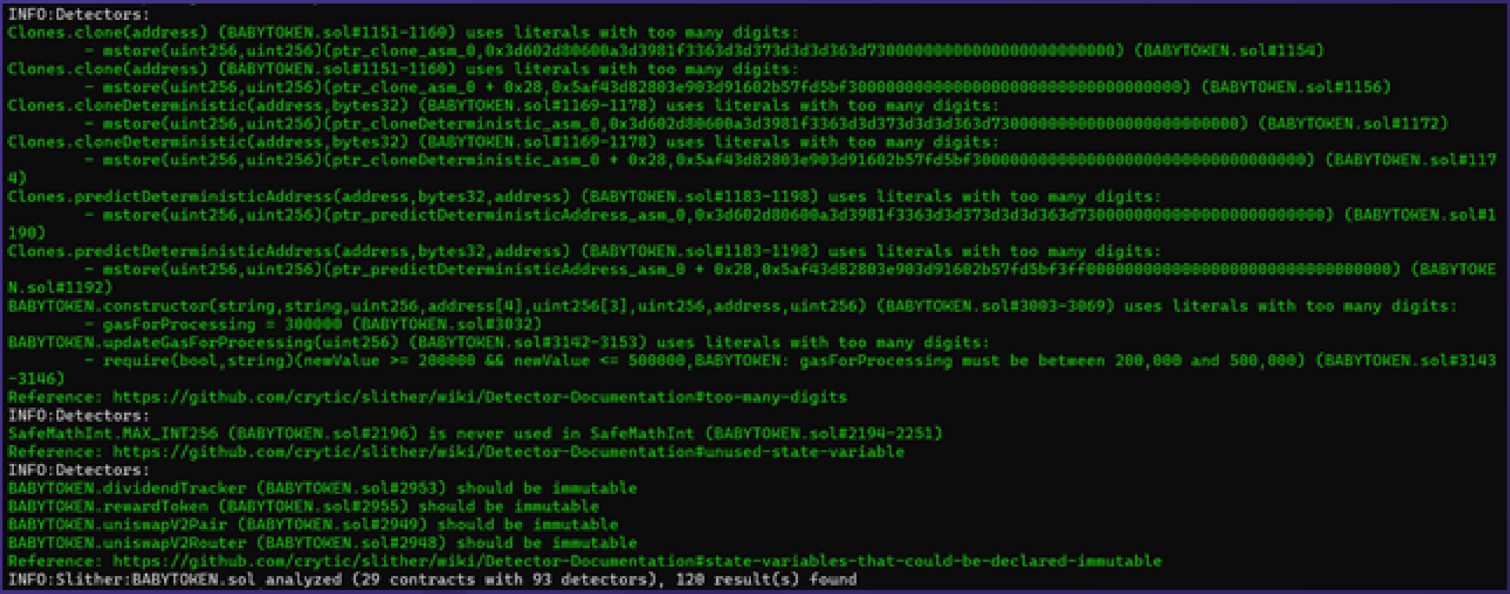
INHERITANCE TREE



STATIC ANALYSIS



```
INFO:Detectors:
BABYTOKEN.getAccountDividendsInfo(address) (BABYTOKEN.sol#3207-3224) ignores return value by dividendTracker.getAccount(account) (BABYTOKEN.sol#3223)
BABYTOKEN.getAccountDividendsInfoAtIndex(uint256) (BABYTOKEN.sol#3226-3243) ignores return value by dividendTracker.getAccountAtIndex(index) (BABYTOKEN.sol#
3242)
BABYTOKEN.claim() (BABYTOKEN.sol#3261-3263) ignores return value by dividendTracker.processAccount(address(msg.sender),false) (BABYTOKEN.sol#3262)
BABYTOKEN.addLiquidity(uint256,uint256) (BABYTOKEN.sol#3437-3450) ignores return value by uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),to
kenAmount,0,0,address(0xdead),block.timestamp) (BABYTOKEN.sol#3442-3449)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
DividendPayingToken.__DividendPayingToken_init(address,string,string)._name (BABYTOKEN.sol#2462) shadows:
        - ERC20Upgradeable._name (BABYTOKEN.sol#1727) (state variable)
DividendPayingToken.__DividendPayingToken_init(address,string,string)._symbol (BABYTOKEN.sol#2463) shadows:
        - ERC20Upgradeable._symbol (BABYTOKEN.sol#1728) (state variable)
DividendPayingToken.dividendOf(address)._owner (BABYTOKEN.sol#2523) shadows:
        - OwnableUpgradeable._owner (BABYTOKEN.sol#2071) (state variable)
DividendPayingToken.withdrawableDividendOf(address)._owner (BABYTOKEN.sol#2530) shadows:
        - OwnableUpgradeable._owner (BABYTOKEN.sol#2071) (state variable)
DividendPayingToken.withdrawnDividendOf(address)._owner (BABYTOKEN.sol#2542) shadows:
        - OwnableUpgradeable._owner (BABYTOKEN.sol#2071) (state variable)
DividendPayingToken.accumulativeDividendOf(address)._owner (BABYTOKEN.sol#2556) shadows:
        - OwnableUpgradeable._owner (BABYTOKEN.sol#2071) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
BABYTOKEN.setSwapTokensAtAmount(uint256) (BABYTOKEN.sol#3073-3079) should emit an event for:
        - swapTokensAtAmount = amount (BABYTOKEN.sol#3078)
BABYTOKEN.setTokenRewardsFee(uint256) (BABYTOKEN.sol#3110-3114) should emit an event for:
        - totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee) (BABYTOKEN.sol#3112)
BABYTOKEN.setLiquiditFee(uint256) (BABYTOKEN.sol#3116-3120) should emit an event for:
        - liquidityFee = value (BABYTOKEN.sol#3117)
        - totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee) (BABYTOKEN.sol#3118)
BABYTOKEN.setMarketingFee(uint256) (BABYTOKEN.sol#3122-3126) should emit an event for:
        - marketingFee = value (BABYTOKEN.sol#3123)
        - totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee) (BABYTOKEN.sol#3124)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256,address,uint256)._uniswapV2Pair (BABYTOKEN.sol#3044-3045) lacks a zero-check on :
                - uniswapV2Pair = _uniswapV2Pair (BABYTOKEN.sol#3047)
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256,address,uint256).serviceFeeReceiver_ (BABYTOKEN.sol#3010) lacks a zero-check on :
                - address(serviceFeeReceiver_).transfer(serviceFee_) (BABYTOKEN.sol#3068)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```



```
INFO:Detectors:
Clones.clone(address) (BABYTOKEN.sol#1151-1160) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_clone_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000) (BABYTOKEN.sol#1154)
Clones.clone(address) (BABYTOKEN.sol#1151-1160) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_clone_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf30000000000000000000000000000000000) (BABYTOKEN.sol#1156)
Clones.cloneDeterministic(address,bytes32) (BABYTOKEN.sol#1169-1178) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000) (BABYTOKEN.sol#1172)
Clones.cloneDeterministic(address,bytes32) (BABYTOKEN.sol#1169-1178) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf30000000000000000000000000000000000) (BABYTOKEN.sol#117
4)
Clones.predictDeterministicAddress(address,bytes32,address) (BABYTOKEN.sol#1183-1198) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000) (BABYTOKEN.sol#1
190)
Clones.predictDeterministicAddress(address,bytes32,address) (BABYTOKEN.sol#1183-1198) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf3ff00000000000000000000000000000000) (BABYTOKE
N.sol#1192)
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256,address,uint256) (BABYTOKEN.sol#3003-3069) uses literals with too many digits:
        - gasForProcessing = 300000 (BABYTOKEN.sol#3032)
BABYTOKEN.updateGasForProcessing(uint256) (BABYTOKEN.sol#3142-3153) uses literals with too many digits:
        - require(bool,string)(newValue >= 200000 && newValue <= 500000,BABYTOKEN: gasForProcessing must be between 200,000 and 500,000) (BABYTOKEN.sol#3143
-3146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
SafeMathInt.MAX_INT256 (BABYTOKEN.sol#2196) is never used in SafeMathInt (BABYTOKEN.sol#2194-2251)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
INFO:Detectors:
BABYTOKEN.dividendTracker (BABYTOKEN.sol#2953) should be immutable
BABYTOKEN.rewardToken (BABYTOKEN.sol#2955) should be immutable
BABYTOKEN.uniswapV2Pair (BABYTOKEN.sol#2949) should be immutable
BABYTOKEN.uniswapV2Router (BABYTOKEN.sol#2948) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:BABYTOKEN.sol analyzed (29 contracts with 93 detectors), 120 result(s) found
```

TESTNET VERSION

1- Approve (passed):
https://testnet.bscscan.com/tx/0x200c319786bd933d1d04dee133e73096233733304fb711643681852d76edef1c

2- Increase Allowance (passed):
https://testnet.bscscan.com/tx/0x0f1e5e1b4100dc38cb1b7af289caa51b706bd934e6cae14142e2646f9180bf77

3- Decrease Allowance (passed):
https://testnet.bscscan.com/tx/0x087301982412 86b7db6287114f2f219c0e3bc960ba8d43b07cbf6ad887a3cf52

4- Exclude From Dividends (passed):
https://testnet.bscscan.com/tx/0xb8cd20ee06cc9b62f2edcd373f4d6c98270146965fe92a3ae80a42eb1069ac50

5- Exclude From Fees (passed):
https://testnet.bscscan.com/tx/0x72d517694cad5e74ca90520c556ce7a5b121ed128c7b42f27449f8c52e63bba8

6- Transfer Ownership (passed):
https://testnet.bscscan.com/tx/0x9308c1e0cdb6616b77a1ca88be93fc1dce3cf40e0785110b7e27d3f76df58009

MANUAL REVIEW

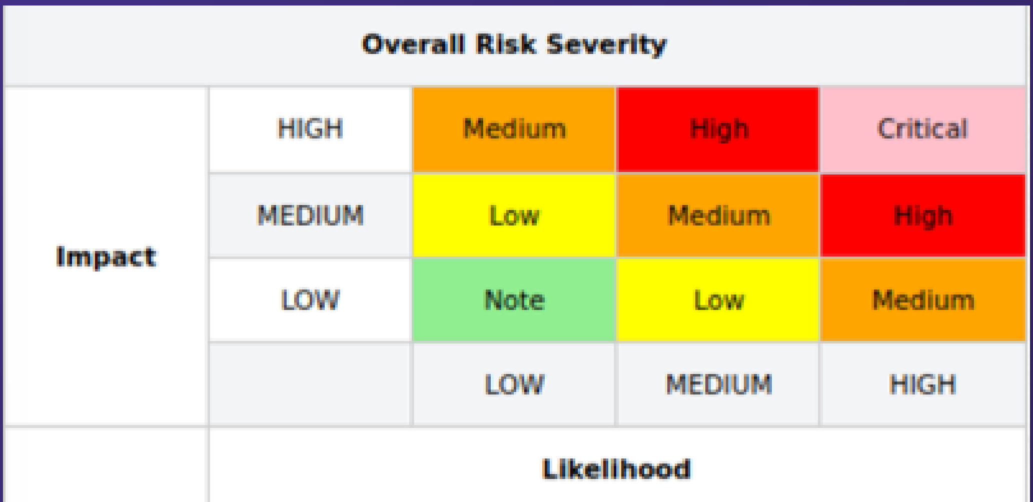
Severity Criteria

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			

LOW RISK FINDING

Centralization – Missing Events

Severity: Low

subject: Missing Events

Status: Open

Overview:
They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setSwapTokensAtAmount(uint256 amount) external
onlyOwner {
require(
    amount > totalSupply() / 10 ** 5,
"BABYTOKEN: Amount must be greater than 0.001% of total supply"
  );
  swapTokensAtAmount = amount;
 }
function setMarketingWallet(address payable wallet) external
onlyOwner {
require(
    wallet != address(0),
"BABYTOKEN: The marketing wallet cannot be the value of zero"
  );
require(!wallet.isContract(), "Marketing wallet cannot be a
contract");
  _marketingWalletAddress = wallet;
 }
function setTokenRewardsFee(uint256 value) external onlyOwner {
  tokenRewardsFee = value;
```

LOW RISK FINDING

```
totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
require(totalFees <= 25, "Total fee is over 25%");
 }
function setLiquiditFee(uint256 value) external onlyOwner {
   liquidityFee = value;
   totalFees =
tokenRewardsFee.add(liquidityFee).add(marketingFee);
require(totalFees <= 25, "Total fee is over 25%");
 }
function setMarketingFee(uint256 value) external onlyOwner {
   marketingFee = value;
   totalFees =
tokenRewardsFee.add(liquidityFee).add(marketingFee);
require(totalFees <= 25, "Total fee is over 25%");
 }
```

Suggestion:

Emit an event for critical changes.

INFORMATIONAL & OPTIMIZATIONS

Optimization

Severity: Optimization

subject: Remove unused code.

Status: Open

Overview:

Unused variables are allowed in Solidity, and they do. not pose a direct security issue. It is the best practice. though to avoid them

```
 function _msgData() internal view virtual returns (bytes calldata) {
return msg.data;
  }
}
function _burn(address account, uint256 amount) internal virtual {
require(account != address(0), "ERC20: burn from the zero address");

   _beforeTokenTransfer(account, address(0), amount);

uint256 accountBalance = _balances[account];
require(accountBalance >= amount, "ERC20: burn amount exceeds
balance");
   unchecked {
     _balances[account] = accountBalance - amount;
     }
   _totalSupply -= amount;

emit Transfer(account, address(0), amount);

   _afterTokenTransfer(account, address(0), amount);
 }
```

INFORMATIONAL & OPTIMIZATIONS

```
function sendValue(address payable recipient, uint256 amount)
internal {
require(address(this).balance >= amount, "Address: insufficient
balance");

   (bool success, ) = recipient.call{value: amount}("");
require(success, "Address: unable to send value, recipient may have
reverted");
 }
function functionCall(address target, bytes memory data) internal
returns (bytes memory) {
return functionCall(target, data, "Address: low-level call failed");
 }
function functionCallWithValue(
address target,
bytes memory data,
uint256 value
 ) internal returns (bytes memory) {
return functionCallWithValue(target, data, value, "Address: low-level
call with value failed");
 }
function functionStaticCall(address target, bytes memory data)
internal view returns (bytes memory) {
return functionStaticCall(target, data, "Address: low-level static call
failed");
 }
function functionDelegateCall(address target, bytes memory data)
internal returns (bytes memory) {
return functionDelegateCall(target, data, "Address: low-level
delegate call failed");
 }
```

INFORMATIONAL & OPTIMIZATIONS

```
function safeTransferFrom(
   IERC20 token,
address from,
address to,
uint256 value
 ) internal {
   _callOptionalReturn(token,
abi.encodeWithSelector(token.transferFrom.selector, from, to,
value));
 }
function safeApprove(
   IERC20 token,
address spender,
uint256 value
 ) internal {
// safeApprove should only be called when setting an initial
allowance,
// or when resetting it to zero. To increase and decrease it, use
// 'safeIncreaseAllowance' and 'safeDecreaseAllowance'
require(
    (value == 0) || (token.allowance(address(this), spender) == 0),
"SafeERC20: approve from non-zero to non-zero allowance"
   );
   _callOptionalReturn(token,
abi.encodeWithSelector(token.approve.selector, spender, value));
 }
function safeIncreaseAllowance(
   IERC20 token,
address spender,
uint256 value
 ) internal {
```

INFORMATIONAL & OPTIMIZATIONS

```
unchecked {
uint256 oldAllowance = token.allowance(address(this), spender);
require(oldAllowance >= value, "SafeERC20: decreased allowance
below zero");
uint256 newAllowance = oldAllowance - value;
    _callOptionalReturn(token,
abi.encodeWithSelector(token.approve.selector, spender,
newAllowance));
  }
 }
```

INFORMATIONAL & OPTIMIZATIONS

Optimization

Severity: Informational

Subject: Remove Safe Math

Status: Open

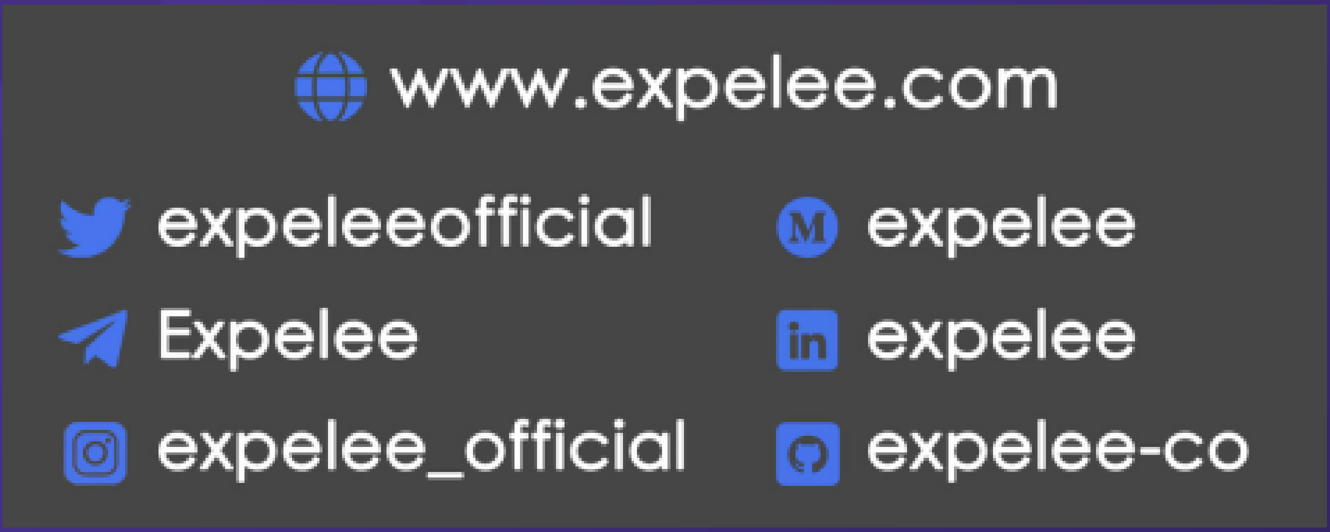
Line: 913-1124

Overview:

compiler version above 0.8.0 can control arithmetic overflow/underflow, it is recommended to remove the unwanted code to avoid high gas fees.

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