expelee

Building the Futuristic Blockchain Ecosystem

Audit Report

FOR

ESIK

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	Not Done
Audit Date	27 Sep 2022

Why high risk?

ESIK is the native token of the Esik Swap Dex and powers the ESIK ecosystem. It is a deflationary token, which rewards its community with BNB Smart chain acquired through transaction fees, while halting weighty selloffs with its interesting elements - Hostile to SNIPER-BOT and ESIK LOCK for steady and consistent development.

-Team Expelee

PROJECT DESCRIPTION

ESIK

ESIK is the native token of the Esik Swap Dex and powers the ESIK ecosystem. It is a deflationary token, which rewards its community with BNB Smart chain acquired through transaction fees, while halting weighty selloffs with its interesting elements - Hostile to SNIPER-BOT and ESIK LOCK for steady and consistent development.

expelee

expelee.com

Social Media Profiles

Block Vest

https://esik.io/

https://t.me/EsikNet

@esikswap

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

-Team Expelee

expelee

expelee.com

CONTRACT DETAILS

Token Name

ESIK

Symbol

ESIK

Network

BSC

Language

Solidity

Contract Address (Verified)

0xa9Da3C5E01492b573EfC9B72DFe86E33BF9D1a27

Total Supply

50,000,000

Decimals

18

Compiler

v0.8.9+commit.e5eed63a

License

default license

Contract SHA-256 Checksum:

7c8aebf1b35ee9c72aa0912d9e2c41feda5d0532dedc261b5dcd049ba5bdbbba

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
- Complier
- Hardhat

FUNCTION OVERVIEW

Can Take Back Ownership	Not Detected
Owner Change Balance	Not Detected
Blacklist	Not Detected
Modify Fees	Detected
Proxy	Not Detected
Whitelisted	Not Detected
Anti Whale	Not Detected
Trading Cooldown	Detected
Transfer Pausable	Detected
Cannot Sell All	Not Detected
Hidden Owner	Not Detected
Mint	Not Detected

VULNERABILITY CHECKLIST

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp 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 Zeppelin module	Passed
Fallback function security	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access 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

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

AUDIT SUMMARY

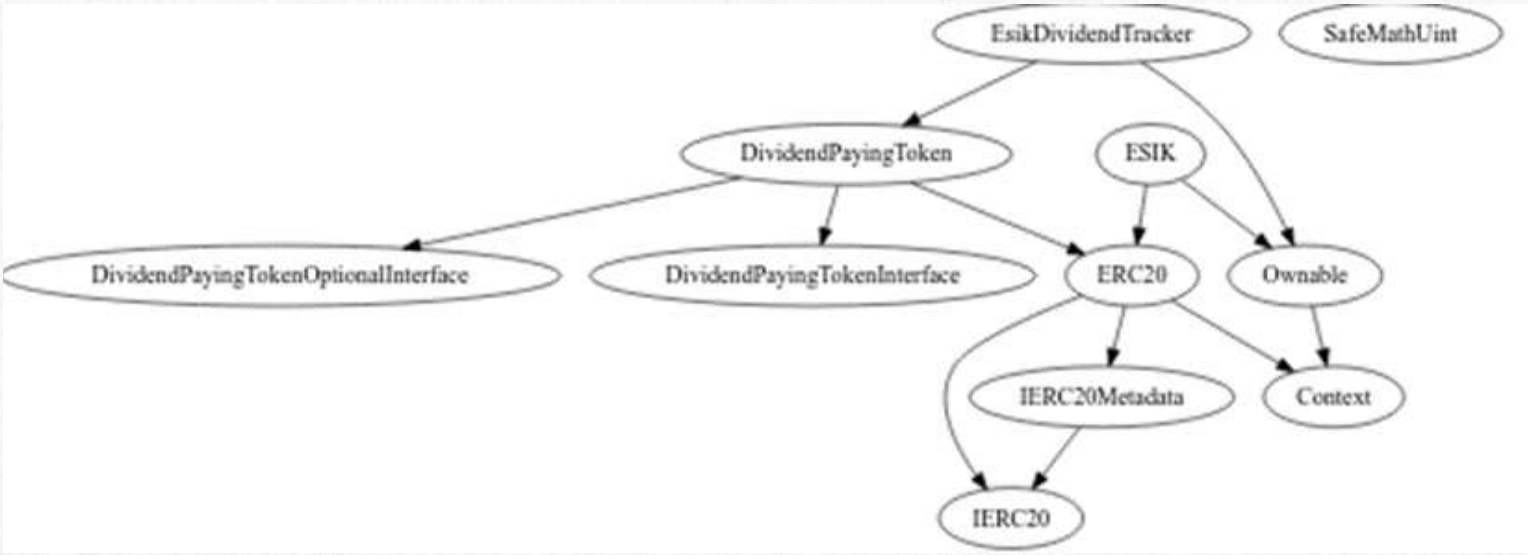
Ownership:

Owner of the contract is:
0x4af08dcf38614c475adc2f97a3998af7c5421a5e
and
0xD14992FAe6377474B1E1bf0944fb59f2f3603094

Contracts & Inheritance Tree:

all of below contractsare in this audit scope

- ESIK
- ESIKDividendTracker



Summary

- Gives rewards in BNB reflections
- Holders are not able to sell more than 50% of their holdings in 24 hour, this time can be modified to any number
- Owner can set Max Wallet, Max Transfer amount to even 0
- Owner is able to set taxes up to 51% on buy and 51% on sell
- Currently there is a 30% tax on buying

MANUAL AUDIT

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to a methodology based on
OWASP standards.
Vulnerabilities are divided into three primary risk categories: high, medium, and low.

High-level considerations for vulnerabilities span the followingkey 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 Summary

- High Risk Findings:3
- Medium Risk Findings:1
- Low Risk Findings:1
- Suggestions & discussion: 4
- Gas Optimizations : 4

High Risk Findings

Centralization Risks

- Owner is able to change trading status, a malicious owner can disable trading at any time, disabling trades will prevent wallets from selling or transering their funds

```
function setTradingIsEnabled(bool status) external onlyOwner {
tradingEnabled = status;
emit UpdateTradingStatus(status);
}
```

- there is not a minimal limit for max transfering amount, this number can be changed to 0 and hence disable all the trades. Same thing is happening with maxHoldingAmount which prevents wallets to hold tokens more than a max amount, setting this number to 0 will disable trades for non-whitelisted wallets
- Owner is able to set buy / sell fees, each one up to 51%
- recommendation : we recommend you to set a reasonable limit for buy / sell taxes

Medium Risk Findings

- Owner is able to withdraw ETH balance of dividend tracker, ETH is the reflection for dividend tracker

Low Risk Findings

Logical

- If creating a new dividendTracker pool address is not excluded from dividends

Suggestions

- !Make sure to use up to 3 indexed arguments for your events
- Owner is able to change token name and symbol later using, this may make confusion for some investors, our suggestion is to declare a final name for the token
- do not use transfer function to send contract ETH, transfer function only forwards 2300 gas, while after EIP-1884 gas cost of some opcodes like SLOAD (200 => 800) changed, if the receiver contract's fallback function uses more gas than 2300, then this transfer() will most likely revert.
- if balance of contract is more than swapTokensAtAmount then swapTokensAtAmount is swapped for BNB, this causes some tokens to be stuck in the contract.

```
 bool canSwap = balanceOf(address(this)) >= swapTokensAtAmount;
 if (tradingEnabled && canSwap && !swapping && swappingEnabled && msg.sender
!pancakePair) {
 swapping = true;
 swapAndDistributeBNB(swapTokensAtAmount);
 swapping = false;
}
```

Gas Optimizations

- deleting unnecessary methods - there is no need to include whole Pancakeswap router and factory interface into the contract, only put the functions that you need, this will reduce contract size and makes it more readable
- redundant check at _transfer

```
if (!_isExcludedFromPeriodLimit[from]) {
```

- SLOAD costs 800 gas, by deleting this redundant check we can save 800 more gas on buys/sells/transfers
- too many reads from storage for tradingEnabled variable, instead declare a memory variable and save up to 3200 gas
- changevisibility withdrawETH() to external

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 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 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. 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 Expelee team disclaims any liability for the resulting losses.