expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Sonic StakingEthFactory

TOKEN OVERVIEW

Risk Findings

Severity	Found
● High	0
● Medium	1
● Low	4
● Informational	0

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

02 Token Overview
03 Table of Contents
04 Overview
05 Contract Details
06 Audit Methodology
07 Vulnerabilities Checklist
08 Risk Classification
09 Inheritance Tree
10 Points to Note
11 Manual Review
17 About Expelee
18 Disclaimer

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	**Medium Risk Detected**
Audit Date	**13 April 2025**

CONTRACT DETAILS

Contract Address:
0xe94237Dad08770A0169D04Db0406a2e7c395Ce73

Contract Name: SonicxSwapEthStakingRewards

Blockchain: Sonic

Contract Type: ERC-20

Contract Creator:
0xa4c576e2373282e94ae08ee4212f552d9555b986

Compiler Version: v0.8.20+commit.a1b79de6

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.

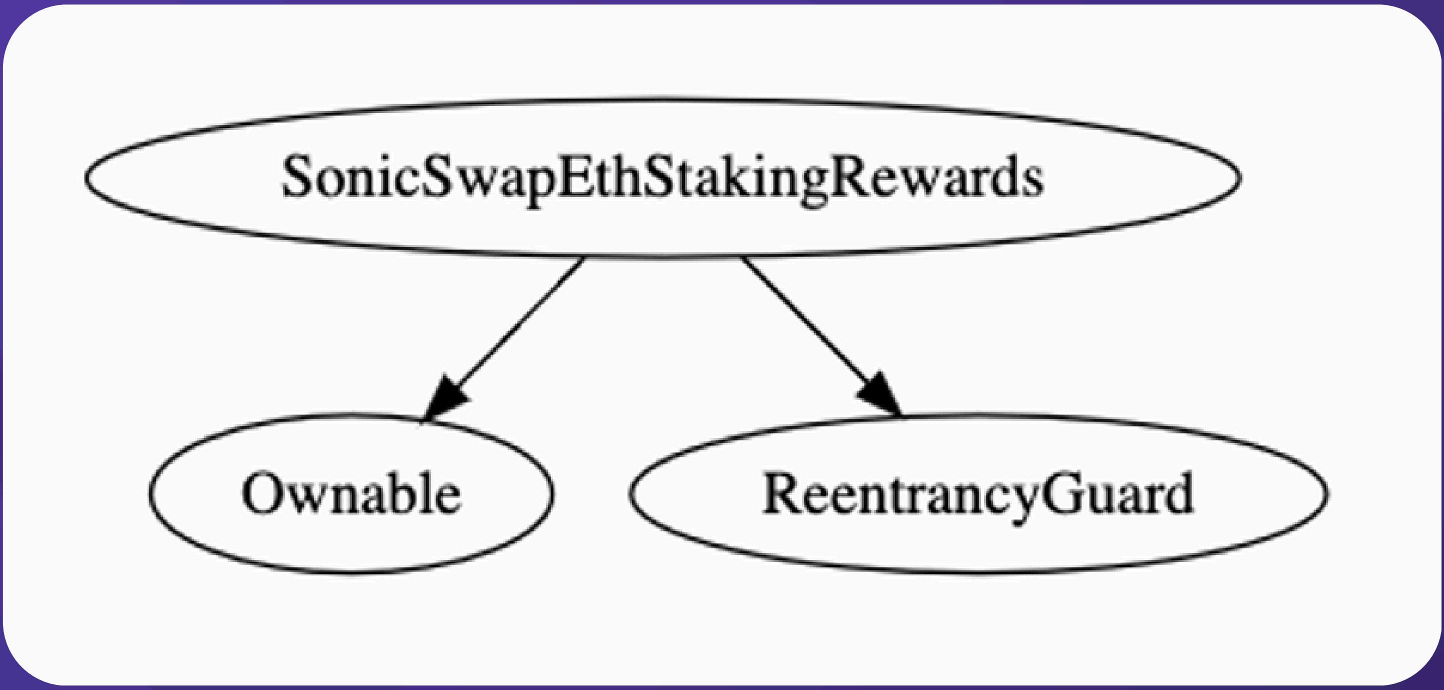
Low Risk

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



POINTS TO NOTE

- The owner can notify reward amounts
- The owner can withdraw tokens from the contract
- The owner can transfer ownership

MANUAL REVIEW

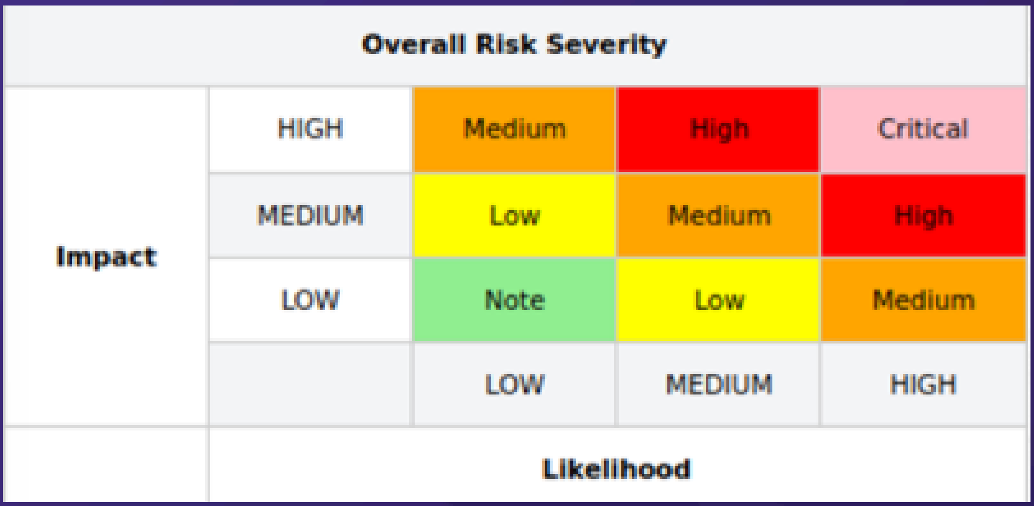
Severity Criteria

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			

MEDIUM RISK FINDING

Centralization – Unsafe ERC20 Transfer in adminWithdraw

Severity: Medium

Overview:
The contract imports and uses SafeERC20 throughout, but specifically in the adminWithdraw function it directly calls transfer() instead of safeTransfer(). This is inconsistent with the rest of the contract and can cause silent failures with certain ERC20 tokens that don't properly return values.

IERC20(rewardsToken).transfer(_toUser, _amount);

Suggestion: Use rewardsToken.safeTransfer(_toUser, _amount); for consistency and safety.

LOW RISK FINDING

Centralization – Missing Event Emission in adminWithdraw

Severity: Low

Suggestion:

Emit events when changing critical parameters, making it impossible to track changes off-chain.

```
 function adminWithdraw(address _toUser, uint256 _amount) external onlyOwner
returns (bool) { //@audit missing events
    require(_toUser != address(0), "Invalid Address");
    require(IERC20(rewardsToken).balanceOf(address(this)) >= _amount,
"StakingFactory: insufficient amount");
    IERC20(rewardsToken).transfer(_toUser, _amount);
    return true;
}
```

LOW RISK FINDING

Centralization – Missing Zero-Address Validation

Severity: Low

Suggestion:

The constructor doesn't validate critical parameters - it doesn't check if _rewardsToken is the zero address or if _rewardDuration is greater than zero.

constructor(address _rewardsToken, uint256 _rewardDuration) Ownable(msg.sender) { rewardsToken = IERC20(_rewardsToken); rewardsDuration = _rewardDuration; }

Recommendation: Add proper validations in the constructor

LOW RISK FINDING

Centralization – Unnecessary SafeMath Usage

Severity: Low

Overview:
Solidity 0.8.x provides built-in overflow/underflow protection, making SafeMath redundant. The contract uses SafeMath methods like add(), sub(), mul(), and div() throughout.

using SafeMath for uint256;

Recommendation: Remove SafeMath dependency and use native arithmetic operations.

LOW RISK FINDING

Centralization – Incorrect Import Path Syntax

Severity: Low

Overview:

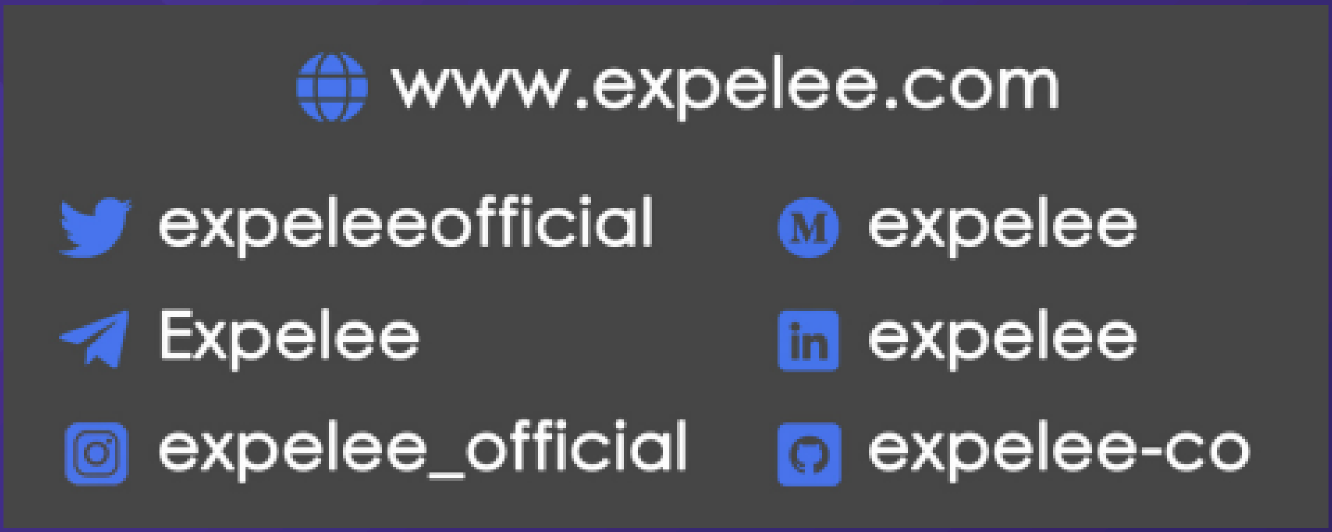
The contract uses backslashes instead of forward slashes in import paths, which will cause compilation errors on most platforms.

import "@openzeppelin\contracts/token/ERC20/IERC20.sol"; import "@openzeppelin\contracts/access/Ownable.sol"; import "@openzeppelin\contracts/token/ERC20/utils/SafeERC20.sol"; import "@openzeppelin\contracts/security/ReentrancyGuard.sol"; import "@openzeppelin\contracts/utils/math/SafeMath.sol"; import "@openzeppelin\contracts/utils/math/Math.sol";

Recommendation: Replace backslashes with forward slashes

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

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