expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Nero Token

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	**No**
Audit Date	**6 May 2023**

expelee

PROJECT DESCRIPTION

$Nero is a reliable, decentralized, and hyper-deflationary token for the community, with utility and scalability where its supply decreases with each transaction with automatic Burn! Let's Moon together!$Nero is a reliable, decentralized, and hyper-deflationary token for the community, with utility and scalability where its supply decreases with each transaction with automatic Burn! Let's Moon together!

eXpelee

SOCIAL MEDIA PROFILES

NERO TOKEN

https://t.me/nerotokenglobal

https://twitter.com/nero_token

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

Team Expelee

CONTRACT DETAILS

Token Name: Nero Token

Symbol: NERO

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0x624b9B1aC0Fb350aed8389a51b26E36147E158c3

Total Supply: 1,000,000,000

Contract SHA-256 Checksum:
90e7db9a1c6a4f64f07771a54cd29308664efaa8

Owner's Wallet:
0xF52EAf4CFfFD2F746430AA1a3D9857164E9E1707

Deployer's Wallet:
 0x04d031ECa639fb7B40C830faB45471a19E7cA0C7

Testnet:
https://testnet.bscscan.com/token/0x1566859b993d0c6e3
08666e463a490d1dc3e7398

OWNER PRIVILEGES

- Contract owner is able to set buy/sell tax each one more than 12%
- Contract owner is able to set tarnsfer tax (0% tax)
- Contract owner is not able to blacklist an arbitrary wallet
- Contract owner is not able to set limit for buy/sell/transfer/holding amounts
- Contract owner is not able to mint new tokens
- Contract owner is not able to disable trades

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0xbf66e094fb38984cb0a7a44debd6bd78bc9e72376e4d59301d2684719ec9d36d

==

Buying from a fee excluded wallet ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x07a8c9dc9e77a5a2562162e1ad1b85545a6e520441578a81b724d41771cf0b31

==

Selling from a fee excluded wallet ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xc3338591b18c871ef7d79fb7746153ea257bdc4703b288bba47bb3c9548fb662

==

Transferring using a fee excluded wallet ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x5d2a910c4e58f5a0e7b95bc264e0eb0473b652adb06ad46a846f9586163e9b46

expelee

TESTNET VERSION

Buying from a regular wallet ✅
Tx (up to 12% tax):
https://testnet.bscscan.com/tx/0x668b4488b4cbb944000b09478f4248337a1691468ca0c9660d09f99fac9f159c

===

Selling from a regular wallet ✅
Tx (up to 12% tax):
https://testnet.bscscan.com/tx/0xbd66b783e2fcffe4366fb7cfdb3c441e9f64bc6e608e66ed290de861b1cc4410

===

Transferring a regular wallet ✓ ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x06ff9ab86c21d344b6ccdebe8c2a1b128187cb792e43140c1be79eba6c911ce5

===

Internal swap (BNB to marketing wallet, Auto liqui✅**ty, burning, rewards)** ✓
Tx :
https://testnet.bscscan.com/tx/0x948ee33ef9a41a7f95fa40be52bff3a6f912f0dea13d01ad4464f2ecdaff9b4a

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.
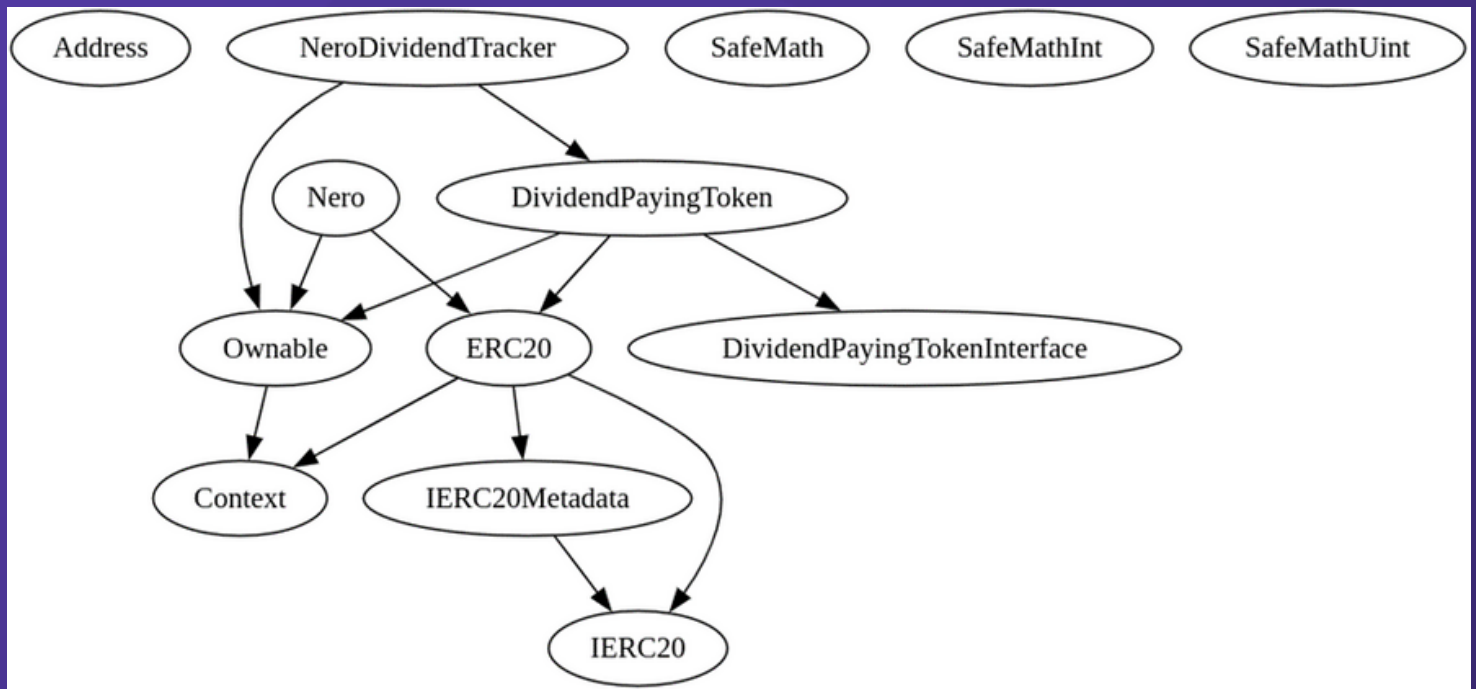
Low Risk

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

Symbol	Meaning
●	Function can modify state
🟩	Function is payable

Contract	Type	Bases			
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	**Mutability**	**Modifiers**
Address	Library				
└	sendValue	Internal 🔒	●		
Nero	Implementation	ERC20, Ownable			
└	<Constructor>	Public ❗	●	ERC20	
└	<Receive Ether>	External ❗	🟩 |NO ❗	
└	processDividendTracker	External ❗	● |NO ❗	
└	setBuyTaxes	External ❗	●	onlyOwner	
└	setSellTaxes	External ❗	●	onlyOwner	
└	claim	External ❗	● |NO ❗	
└	withdrawToken	External ❗	●	onlyOwner	
└	withdrawETH	External ❗	●	onlyOwner	
└	excludeFromFees	Public ❗	●	onlyOwner	
└	excludeMultipleAccountsFromFees	Public ❗	●	onlyOwner	
└	excludeFromDividends	External ❗	●	onlyOwner	
└	setMarketingWallet	External ❗	●	onlyOwner	
└	setSwapTokensAtAmount	External ❗	●	onlyOwner	
└	setSwapEnabled	External ❗	●	onlyOwner	
└	setMinBalanceForDividends	External ❗	●	onlyOwner	
└	_setAutomatedMarketMakerPair	Private 🔒	●		
└	setGasForProcessing	External ❗	●	onlyOwner	
└	setClaimWait	External ❗	●	onlyOwner	
└	getClaimWait	External ❗	|NO ❗	
└	getTotalDividendsDistributed	External ❗	|NO ❗	
└	isExcludedFromFees	Public ❗	|NO ❗	
└	withdrawableDividendOf	Public ❗	|NO ❗	
└	getCurrentRewardToken	External ❗	|NO ❗	
└	dividendTokenBalanceOf	Public ❗	|NO ❗	
└	getAccountDividendsInfo	External ❗	|NO ❗	
└	getAccountDividendsInfoAtIndex	External ❗	|NO ❗	
└	getLastProcessedIndex	External ❗	|NO ❗	
└	getNumberOfDividendTokenHolders	External ❗	|NO ❗	
└	_transfer	Internal 🔒	●		
└	swapAndLiquify	Private 🔒	●		
└	swapTokensForBNB	Private 🔒	●		
└	addLiquidity	Private 🔒	●		
NeroDividendTracker	Implementation	Ownable, DividendPayingToken			
└	<Constructor>	Public ❗	●	DividendPayingToken	
└	_transfer	Internal 🔒			
└	setMinBalanceForDividends	External ❗	●	onlyOwner	

FUNCTION DETAILS

```
|  └ | excludeFromDividends | External  ❗ |  🔴  | onlyOwner |
|  └ | updateClaimWait | External  ❗ |  🔴  | onlyOwner |
|  └ | getLastProcessedIndex | External  ❗ |  |NO ❗ |
|  └ | getNumberOfTokenHolders | External  ❗ |  |NO ❗ |
|  └ | getCurrentRewardToken | External  ❗ |  |NO ❗ |
|  └ | getAccount | Public  ❗ |  |NO ❗ |
|  └ | getAccountAtIndex | Public  ❗ |  |NO ❗ |
|  └ | canAutoClaim | Private 🔐 |  ||
|  └ | setBalance | Public  ❗ |  🔴  | onlyOwner |
|  └ | process | Public  ❗ |  🔴 |NO ❗ |
|  └ | processAccount | Public  ❗ |  🔴  | onlyOwner |
||||||
| **DividendPayingToken** | Implementation | ERC20, DividendPayingTokenInterface, Ownable |||
|  └ | <Constructor> | Public  ❗ |  🔴  | ERC20 |
|  └ | <Receive Ether> | External  ❗ |  🟩🟩 |NO ❗ |
|  └ | distributeDividends | Public  ❗ |  🟩🟩 |NO ❗ |
|  └ | _withdrawDividendOfUser | Internal 🔒 |  🔴  ||
|  └ | setRewardToken | External  ❗ |  🔴  | onlyOwner |
|  └ | swapBnbForCustomToken | Internal 🔒 |  🔴  ||
|  └ | dividendOf | Public  ❗ |  |NO ❗ |
|  └ | withdrawableDividendOf | Public  ❗ |  |NO ❗ |
|  └ | withdrawnDividendOf | Public  ❗ |  |NO ❗ |
|  └ | accumulativeDividendOf | Public  ❗ |  |NO ❗ |
|  └ | _transfer | Internal 🔒 |  🔴  ||
|  └ | _tokengeneration | Internal 🔒 |  🔴  ||
|  └ | _burn | Internal 🔒 |  🔴  ||
|  └ | _setBalance | Internal 🔒 |  🔴  ||
||||||
| **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 🔒 |  🔴  ||
|  └ | _tokengeneration | Internal 🔒 |  🔴  ||
|  └ | _burn | Internal 🔒 |  🔴  ||
|  └ | _approve | Internal 🔒 |  🔴  ||
|  └ | _beforeTokenTransfer | Internal 🔒 |  🔴  ||
```

FUNCTION DETAILS

```
||||||
| **IERC20** | Interface |  |||
|  └ | totalSupply | External  ❗ |  | |NO ❗ |
|  └ | balanceOf | External  ❗ |  | |NO ❗ |
|  └ | transfer | External  ❗ |  ● |NO ❗ |
|  └ | allowance | External  ❗ |  | |NO ❗ |
|  └ | approve | External  ❗ |  ● |NO ❗ |
|  └ | transferFrom | External  ❗ |  ● |NO ❗ |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
|  └ | name | External  ❗ |  | |NO ❗ |
|  └ | symbol | External  ❗ |  | |NO ❗ |
|  └ | decimals | External  ❗ |  | |NO ❗ |
||||||
| **Context** | Implementation |  |||
|  └ | _msgSender | Internal 🔒 |  | |
|  └ | _msgData | Internal 🔒 |  | |
||||||
| **SafeMath** | Library |  |||
|  └ | add | Internal 🔒 |  | |
|  └ | sub | Internal 🔒 |  | |
|  └ | sub | Internal 🔒 |  | |
|  └ | mul | Internal 🔒 |  | |
|  └ | div | Internal 🔒 |  | |
|  └ | div | Internal 🔒 |  | |
|  └ | mod | Internal 🔒 |  | |
|  └ | mod | Internal 🔒 |  | |
||||||
| **SafeMathInt** | Library |  |||
|  └ | mul | Internal 🔒 |  | |
|  └ | div | Internal 🔒 |  | |
|  └ | sub | Internal 🔒 |  | |
|  └ | add | Internal 🔒 |  | |
|  └ | abs | Internal 🔒 |  | |
|  └ | toUint256Safe | Internal 🔒 |  | |
||||||
| **SafeMathUint** | Library |  |||
|  └ | toInt256Safe | Internal 🔒 |  | |
||||||
| **DividendPayingTokenInterface** | Interface |  |||
|  └ | dividendOf | External  ❗ |  | |NO ❗ |
|  └ | distributeDividends | External  ❗ |  🟩 |NO ❗ |
|  └ | withdrawableDividendOf | External  ❗ |  | |NO ❗ |
|  └ | withdrawnDividendOf | External  ❗ |  | |NO ❗ |
|  └ | accumulativeDividendOf | External  ❗ |  | |NO ❗ |
||||||
| **Ownable** | Implementation | Context |||
```

FUNCTION DETAILS

```
| └ | <Constructor> | Public ❗ | ● |NO ❗ |
| └ | owner | Public ❗ | |NO ❗ |
| └ | renounceOwnership | Public ❗ | ● | onlyOwner |
| └ | transferOwnership | Public ❗ | ● | onlyOwner |
||||||
| **IPair** | Interface | |||
| └ | sync | External ❗ | ● |NO ❗ |
||||||
| **IFactory** | Interface | |||
| └ | createPair | External ❗ | ● |NO ❗ |
| └ | getPair | External ❗ | |NO ❗ |
||||||
| **IRouter** | Interface | |||
| └ | factory | External ❗ | |NO ❗ |
| └ | WETH | External ❗ | |NO ❗ |
| └ | addLiquidityETH | External ❗ | 💵 |NO ❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | ● |NO ❗ |
| └ | swapExactETHForTokens | External ❗ | 💵 |NO ❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | ● |NO ❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 |NO ❗ |
||||||
| **IterableMapping** | Library | |||
| └ | get | Internal 🔒 | ||
| └ | getIndexOfKey | Internal 🔒 | ||
| └ | getKeyAtIndex | Internal 🔒 | ||
| └ | size | Internal 🔒 | ||
| └ | set | Internal 🔒 | ● ||
| └ | remove | Internal 🔒 | ● ||
```

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		

FINDINGS

Findings	Severity	Found
High Risk	🔴 High	0
Medium Risk	🟠 Medium	0
Low Risk	🟡 Low	0
Suggestion & discussion	🔵 Informational	0
Gas Optimizations	🟣 Gas Opt.	0

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   in 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.

Building the Futuristic Blockchain Ecosystem