expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Green Whale Challenge

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	**24 April 2023**

PROJECT DESCRIPTION

To know more about Green Whale Challenge visit their Social Media handles.

SOCIAL MEDIA PROFILES

GREEN WHALE CHALLENGE

https://t.me/greenwhalechallenge

https://twitter.com/gwctoken

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

Team Expelee

CONTRACT DETAILS

Token Name: Green Whale Challenge

Symbol: GWC

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0x5f7e995ada0B4acCc7506eB98966E3B71eE0C41B

Total Supply: 10,000,000

Contract SHA-256 Checksum:
706a26798c495521786ebd6546ec01ed34758f8d

Owner's Wallet:
0x63355a0b03a3133c649eae244bc57a6849e1f4d7

Testnet:
https://testnet.bscscan.com/token/0x49eab73b24aa104927c29223836a07ab0323b4c1

OWNER PRIVILEGES

- Contract owner is not able to set buy/sell taxes over 12% each
- Contract owner is not able to set transfer tax (0%)
- Contract owner is not able to blacklist an arbitrary wallet
- Contract owner is not able to disable trades/transfers
- Contract owner is not able to mint new tokens
- **Contract owner must enable trades for public**

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

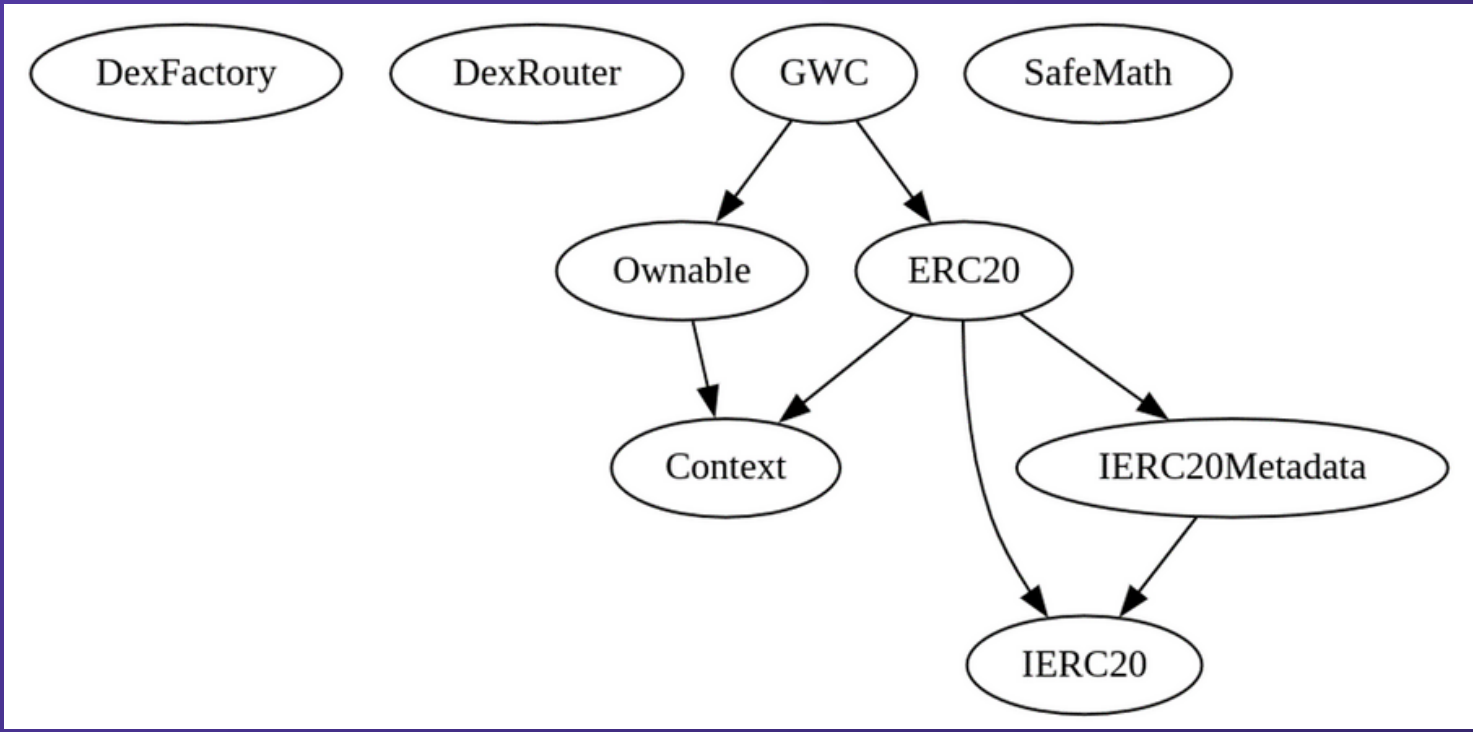
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

expelee

INHERITANCE TREES



expelee

FUNCTION DETAILS

Symbol	Meaning
🔴	Function can modify state
💶	Function is payable

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
GWC	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🔴	ERC20
└	startTrading	External ❗	🔴	onlyOwner
└	setDevWallet	External ❗	🔴	onlyOwner
└	setMarketing	External ❗	🔴	onlyOwner
└	setBuyFees	External ❗	🔴	onlyOwner
└	setSellFees	External ❗	🔴	onlyOwner
└	setSwapTokensAtAmount	External ❗	🔴	onlyOwner
└	toggleSwapping	External ❗	🔴	onlyOwner
└	setWhitelist	External ❗	🔴	onlyOwner
└	checkWhitelist	External ❗	NO ❗	
└	_takeTax	Internal 🔒	🔴	
└	_transfer	Internal 🔒	🔴	
└	manageTaxes	Internal 🔒	🔴	
└	swapAndLiquify	Internal 🔒	🔴	
└	swapToETH	Internal 🔒	🔴	
└	addLiquidity	Private 🔐	🔴	
└	withdrawStuckETH	External ❗	🔴	onlyOwner
└	withdrawStuckTokens	External ❗	🔴	onlyOwner
└	<Receive Ether>	External ❗	💶	NO ❗
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO ❗
└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
└	_transferOwnership	Internal 🔒	🔴	
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		

expelee

FUNCTION DETAILS

└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
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 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_spendAllowance	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	
IERC20Metadata	Interface	IERC20		
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	decimals	External ❗		NO ❗

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			

expelee

FINDINGS

Findings	Severity	Found
High Risk	High	0
Medium Risk	Medium	0
Low Risk	Low	0
Suggestion & discussion	Informational	1
Gas Optimizations	Gas Opt.	0

INFORMATIONAL FINDING

Enabling trades is not guaranteed

Severity : Informational

Overview

The owner of the contract must enable trades for public, otherwise no one would be able to buy/sell/transfer their tokens except whitelisted wallets.

```
  function startTrading() external onlyOwner {
tradingStatus = true;
  }
```

Suggestion :

To mitigate this issue there are several options:
- Enable tradings before presale

Issue Status: Open

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.