expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

APPLEVR

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	1
Medium	0
Low	0
Informational	0

Centralization Risks

Owner Privileges	Description
Can Owner Set Taxes >25% ?	Not Detected
Owner needs to enable trading ?	Yes, owner needs to enable trades
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	**10 June 2023**

CONTRACT DETAILS

Token Name: APPLEVR

Symbol: APPLEVR

Network: Binance Smart Chain

Language: Solidity

Contract Address: --

Total Supply: 10,000,000,000

Owner's Wallet: --

Deployer's Wallet: --

Testnet.
https://testnet.bscscan.com/token/0x95e50E9956434b8A2ae29c6b215b684d851A2E3d

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

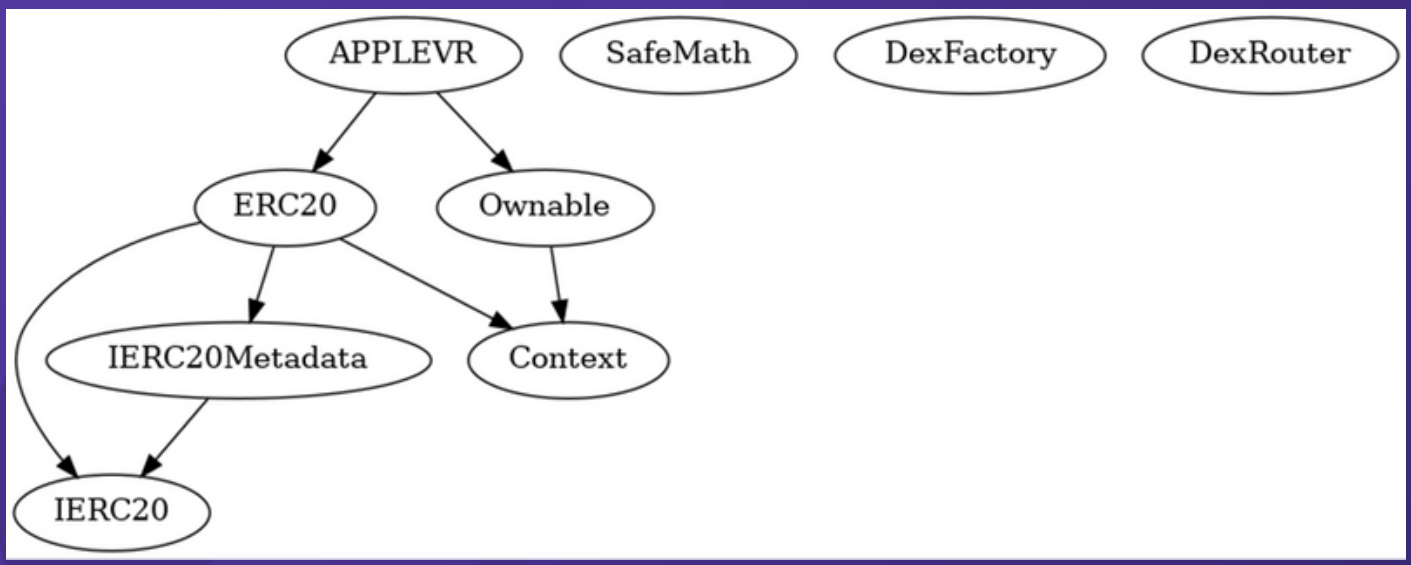
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

expelee

INHERITANCE TREES



FUNCTION DETAILS

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🔴	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🔴	NO ❗
└	transferFrom	External ❗	🔴	NO ❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IERC20Metadata	Interface	IERC20		
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	decimals	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 🔒	🔴	
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		

FUNCTION DETAILS

└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO ❗
└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
└	_transferOwnership	Internal 🔒	🔴	
DexFactory	Interface			
└	createPair	External ❗	🔴	NO ❗
DexRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
APPLEVR	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🔴	ERC20
└	setmarketingWallet	External ❗	🔴	onlyOwner
└	setSwapTokensAtAmount	External ❗	🔴	onlyOwner
└	toggleSwapping	External ❗	🔴	onlyOwner
└	setWhitelistStatus	External ❗	🔴	onlyOwner
└	checkWhitelist	External ❗		NO ❗
└	startTrading	External ❗	🔴	onlyOwner
└	updateBuyTax	External ❗	🔴	onlyOwner
└	updateSellTax	External ❗	🔴	onlyOwner
└	updateTransferTax	External ❗	🔴	onlyOwner
└	_takeTax	Internal 🔒	🔴	
└	_transfer	Internal 🔒	🔴	
└	internalSwap	Internal 🔒	🔴	
└	swapToETH	Internal 🔒	🔴	
└	withdrawStuckETH	External ❗	🔴	onlyOwner
└	withdrawStuckTokens	External ❗	🔴	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗

FUNCTION DETAILS

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0xeb57a95a480c18123a31f1ba03d71127fbe7882bc27c277da444c7f99a8733da

==========

Buying from a fee excluded wallet ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x61be09035dc330c64723b11200ccdb756af92d32a19816f79efa3816cb46abab

==========

Selling from a fee excluded wallet ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xd9943191940680cc827b77d94f5c25ac65b8dc61b3d547d437439469cd4a52ad

==========

Transferring using a fee excluded wallet ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xa65ae351e0f30607704d32ffc0656c29548e5f3d07abbc90b827a47c7b9aaa2c

==========

Buying from a regular wallet ✅
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0xe76788fe041992dacdebd4a06d0a8fe5d492020009b40c3839a81bf71ea80e9a

==========

Selling from a regular wallet ✅
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0xda87350e83b8d0f869efabf22b2094b3528f7a2af783f4753f83b71f5d2fbdaf

TESTNET VERSION

Transferring a regular wallet ✅
Tx (0-10%):
https://testnet.bscscan.com/tx/0x78f1ede735818149f1ea9d00beb7f4b0e53714057d6f8e8f8b37b9a55dd62a56

==========

Internal swap (marketing BNB) ✅
Tx:
https://testnet.bscscan.com/address/0x33707e9bae7cb5843288a08bcaeee0e33c4af000#internaltx

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			

HIGH RISK FINDING

Enabling trades is not guaranteed

Category: Centralization
Status: Open
Severity: High

Overview:

Owner must enable trades for investors manually. If trades remain disabled, holders wont be able to trade their tokens.

```
  function startTrading() external onlyOwner {
require(!tradingEnabled, "Trading already enabled");
      tradingEnabled = true;
  }
```

Suggestion:

to mitigate this issue there are several options:
- Enable trades before end of presale
- Transfer ownership to a trusted 3rd party to guarantee enable of trades

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.