



Building the Futuristic **Blockchain Ecosystem**

SECURITY AUDIT REPORT

USR

TOKEN OVERVIEW

Risk Findings

Severity	Found
● High	2
● Medium	1
● Low	0
● 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 ?	Detected
● Can Owner Mint ?	Not Detected
● Can Owner Blacklist ?	Detected
● Can Owner set Max Wallet amount ?	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 Trees	_____
10	Function Details	_____
13	Testnet Version	_____
15	Manual Review	_____
16	High Risk Finding	_____
18	Medium Risk Finding	_____
19	About Expelee	_____
20	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	Failed
KYC Verification	-
Audit Date	13 august 2023

CONTRACT DETAILS

Token Name: USR

Symbol: USR

Network: Binance Smart Chain

Language: Solidity

Contract Address:

–

Total Supply: 420,000,000,000

Owner's Wallet:

–

Deployer's Wallet:

–

Testnet.

<https://testnet.bscscan.com/token/0x3dDCD61164137d4F32A64d1e1Ad0c6B719041b82>

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 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.

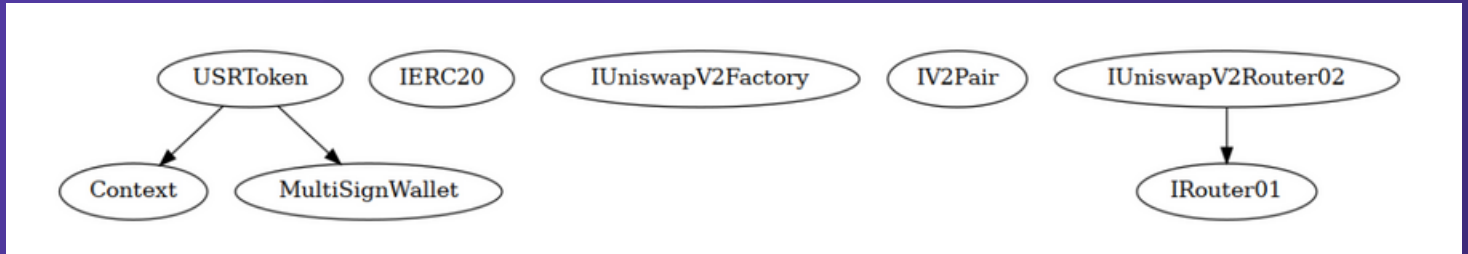
Low Risk

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

Informational

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

INHERITANCE TREES



FUNCTION DETAILS

```

| Contract |      Type      |      Bases      |      |      |
|:-----:|:-----:|:-----:|:-----:|:-----:|
|  L  | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
|||||
| **Context** | Implementation | |||
|  L  | _msgSender | Internal  | ||
|  L  | _msgData | Internal  | ||
|||||
| **MultiSignWallet** | Implementation | |||
|  L  | <Constructor> | Public  |  | NO  |
|  L  | newTransaction | External  |  | onlyOwner |
|  L  | approveTransaction | External  |  | onlyOwner txnExists notApproved notExecuted |
|  L  | _getApprovalCount | Public  | | NO  |
|  L  | executeTransaction | Internal  |  | txnExists notExecuted |
|  L  | revoke | External  |  | onlyOwner txnExists notExecuted |
|||||
| **IERC20** | Interface | |||
|  L  | balanceOf | External  | | NO  |
|  L  | transfer | External  |  | NO  |
|  L  | allowance | External  | | NO  |
|  L  | approve | External  |  | NO  |
|  L  | transferFrom | External  |  | NO  |
|||||
| **IUniswapV2Factory** | Interface | |||
|  L  | getPair | External  | | NO  |
|  L  | createPair | External  |  | NO  |
|||||
| **IV2Pair** | Interface | |||
|  L  | factory | External  | | NO  |
|  L  | getReserves | External  | | NO  |
|  L  | sync | External  |  | NO  |
|||||

```


FUNCTION DETAILS

```

|  | enableTrade | Public | ! | 🔴 | onlyOwner |
|  | pauseTrade | Public | ! | 🔴 | onlyOwner |
|  | disableBurn | Public | ! | 🔴 | onlyOwner |
|  | setUserFoundationWallet | External | ! | 🔴 | onlyOwner |
|  | updateShares | Internal | 🗝️ | 🔴 | |
|  | setBuyUserFoundationPercentage | External | ! | 🔴 | onlyOwner |
|  | setBuyLiquidityPercentage | External | ! | 🔴 | onlyOwner |
|  | setBuyBurnPercentage | External | ! | 🔴 | onlyOwner |
|  | setSellUserFoundationPercentage | External | ! | 🔴 | onlyOwner |
|  | setSellLiquidityPercentage | External | ! | 🔴 | onlyOwner |
|  | setSellBurnPercentage | External | ! | 🔴 | onlyOwner |
|  | setTransferLiquidityPercentage | External | ! | 🔴 | onlyOwner |
|  | setTransferBurnPercentage | External | ! | 🔴 | onlyOwner |
|  | addToBlacklist | Public | ! | 🔴 | onlyOwner |
|  | removeFromBlacklist | Public | ! | 🔴 | onlyOwner |
|  | setTaxThreshold | External | ! | 🔴 | onlyOwner |
|  | setMaxAmount | External | ! | 🔴 | onlyOwner |
|  | recoverTokensFromContract | External | ! | 🔴 | onlyOwner |
|  | recoverETHfromContract | External | ! | 🔴 | onlyOwner |
|  | recoverUSRfromUser | External | ! | 🔴 | onlyOwner |
|  | swapTokensForEth | Private | 🗝️ | 🔴 | |
|  | swapAndLiquify | Internal | 🗝️ | 🔴 | |
|  | addLiquidity | Private | 🗝️ | 🔴 | |
|  | _transfer | Internal | 🗝️ | 🔴 | |
|  | _calculateTax | Internal | 🗝️ | | |
|  | <Fallback> | External | ! | 💵 | NO ! |
|  | <Receive Ether> | External | ! | 💵 | NO ! |

```

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

TESTNET VERSION

Adding Liquidity 

Tx:

<https://testnet.bscscan.com/tx/0x4406f7c6d5499d8a8e7f8d8016af900f44b4120abc805d0f8ae6783d5d5f4855>

=====

Buying when excluded from fees 

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x05f38e26917e0adaaa5a3595f550a111bb9483a6a0176ffc0d410f89f4d3da66>

=====

Selling when excluded from fees 

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x1befa2467408290ae0b51a59125dc01b9c186cd30e7927f3594df5006e7db57e>

=====

Transferring when excluded from fees 

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x2c839826b51650b8772adef0db5a81d14f9aff7b55f0307659a7e784250c5843>

=====

Buying 

Tx (0-5% tax):

<https://testnet.bscscan.com/tx/0x3d4164433ef080f35e3379c3c693dc32780ae42af93027aefdd9b32dc01e1b79>

TESTNET VERSION

Selling ✓

Tx (0-5% tax):

<https://testnet.bscscan.com/tx/0x5aa58144482964119f8a0b1c7f349291b327da992c849e5a4188263c26931e7e>

=====

Transferring ✓

Tx (0-5% tax):

<https://testnet.bscscan.com/tx/0x2fc96db8613d672e5acdfd0ddce0fec3a9d2b8480b25eb1a0bfd8363a614eb21>

=====

Internal swap (BNB to marketing wallet | reward token to dividend tracker | reward distribution) ✓

Tx:

<https://testnet.bscscan.com/tx/0x5aa58144482964119f8a0b1c7f349291b327da992c849e5a4188263c26931e7e>

MANUAL REVIEW

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to methodology based on OWASP standards.

Vulnerabilities are divided into three primary risk categories:

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

Blacklisting

Category: **Centralization**

Status: **Open**

Impact: **High**

Overview:

Owner of the contract is able to blacklist an arbitrary address, blacklisted wallets are not able to buy/sell/transfer tokens.

```
function addToBlacklist(address account) public onlyOwner {  
    require(!blacklisted[account], "Account is already blacklisted");  
    require(_msgSender() != account, "Cannot blacklist self");  
    blacklisted[account] = true;  
}
```

Suggestion:

Delete addToBlacklist function and implement a more decentralized method for blacklisting bad actors such as MEV bots, sniper bots etc.

HIGH RISK FINDING

Enabling/Disabling trades

Category: Centralization

Status: Open

Impact: High

Overview:

Trades are disabled by default owner of the contract is able enable/disable trades at anytime. When trades are disabled, no one would be able to transfer their tokens. Owner must enable trades In order to activate tokens tansfers.

```
function pauseTrade() public onlyOwner {  
    trade_open = false;  
}  
function disableBurn(bool _status) public onlyOwner {  
    burn_disable = _status;  
}
```

Suggestion:

Ensure that trades remain enabled after enabling it

MEDIUM RISK FINDING

LP tokens accumulated in the contract

Category: Centralization

Status: Open

Impact: Medium

Overview:

Contract is receiving LP tokens generated from auto-liquidity. These LP tokens can be withdrawn by owner of the contract. LP tokens can be used to remove a portion of liquidity pool (both ETH and USR tokens)

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    _approve(address(this), address(uniswapV2Router), tokenAmount);
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        address(this),
        block.timestamp
    );
}
```

Suggestion:

It is recommended to burn or lock new LP tokens.

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](https://twitter.com/expeleeofficial)

 [expelee](https://medium.com/expelee)

 [Expelee](https://t.me/Expelee)

 [expelee](https://in.linkedin.com/expelee)

 [expelee_official](https://www.instagram.com/expelee_official)

 [expelee-co](https://github.com/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.

The logo for Expelee, featuring the word "expelee" in a stylized font. The "ex" is in white, and "pelee" is in orange. The letters are bold and modern.

Building the Futuristic **Blockchain Ecosystem**