



expelee

A Secure Place For Web3

SMART CONTRACT AUDIT OF

Kyu Token Presale



Contract Address

0x987a6509A5927dd8046e4ba9485C7e0e24c832A6

www.expelee.com | Page 1 |





Audit Summary

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

Ownership: NOT RENOUNCED

KYC Verification: Done

Audit Date: 02/07/2022

Audit Team: EXPELEE

Be aware that smart contracts deployed on the blockchain aren't resistant to internal exploit, external vulnerability, or hack. For a detailed understanding of risk severity, source code vulnerability, functional hack, and audit disclaimer, kindly refer to the audit.

www.expelee.com | Page 2 |





DISCLAMER

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 guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always Do your own research and protect 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. Always 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.

www.expelee.com | Page 3 |



Contract Review

Contract Name	KUY
Compiler Version	v0.8.10+commit.fc410830
Optimization	No with 200 runs
License	No license
Explorer	https://bscscan.com/address/0x987a6 509A5927dd8046e4ba9485C7e0e24c83 2A6#code
Symbol	KUY
Decimals	18
Total Supply	100,000,000
Domain	https://kuytoken.com/

www.expelee.com | Page 4 |





Project Review

Token Name: KUY

Web Site: https://kuytoken.com/

Twitter: https://twitter.com/kuytoken

Telegram: https://t.me/kuytokenen

Contract Address:

0x987a6509A5927dd8046e4ba9485C7e0e24c832A6

Platform: Binance Smart Chain

Token Type: BEP 20

Language: SOLIDITY

www.expelee.com Page 5 |





Audit Methodology

The scope of this report is to audit the smart contract source code. We have scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category

Smart Contract
Vulnerabilities

- Unhandled Exceptions

- Transaction Order Dependency

- Integer Overflow

- Unrestricted Action

Incorrect Inheritance Order

- Typographical Errors

- Requirement Violation

Source Code Review

- Gas Limit and Loops

- Deployment Consistency

- Repository Consistency

- Data Consistency

- Token Supply Manipulation

Functional Assessment - Operations Trail & Event Generation

- Assets Manipulation

- Liquidity Access

www.expelee.com Page 6 |





Vulnerability Checklist

Νō	Description.	Result
1	Compiler warnings.	Passed
2	Race conditions and Re-entrancy. Cross-function raceconditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

www.expelee.com | Page 7 |



Manual Audit

- Low-Risk
- 2 low-risk code issues found
 - Medium-Risk
- 0 medium-risk code issues found
 - High-Risk
 - 0 high-risk code issues found

www.expelee.com | Page 8 |





1) Unused Return

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    // approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount);

    // add the liquidity
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        address(this),
        block.timestamp
    );
}
```

Recommendation

Ensure that all the return values of the function calls are used.

www.expelee.com | Page 9 |



2) Local variable shadowing

Detection of shadowing using local variables.

```
function transfer(address to, uint256 amount)
        public
        virtual
        override
        returns (bool)
    {
        address owner = _msgSender();
        _transfer(owner, to, amount);
        return true;
    }
     * @dev See {IKUY20-allowance}.
    function allowance(address owner, address spender)
        public
        view
        virtual
        override
        returns (uint256)
    {
        return _allowances[owner][spender];
    }
```

Recommendation

Check that the new address is not zero.

www.expelee.com Page 10 |



Audit Summary

Compiled with solc

Number of lines: 1498 (+ 0 in dependencies, + 0 in tests)

Number of assembly lines: 0

Number of contracts: 10 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 18 Number of informational issues: 42

Number of low issues: 2 Number of medium issues: 0 Number of high issues: 0

ERCs: ERC2612, ERC20

+	.	-	+	+	
Name	# functions	ERCS	ERC20 info	Complex code	Features
SafeMath	13			No	
Address	11			No	Send ETH
					Delegatecall
					Assembly
IUniswapV2Factory	8			No	
IUniswapV2Pair	27	ERC20,ERC2612	∞ Minting	No	
			Approve Race Cond.		
IUniswapV2Router02	24			No	Receive ETH
KUY	37	ERC20	No Minting	No	Send ETH
1	<u> </u>		Approve Race Cond.	I	
1	<u> </u>		T. T	I	
+	+	·	+	+	++

www.expelee.com Page 11 |



Manual Audit (Contract Function)

```
contract KUY is Context, IKUY20, Ownable {
   using SafeMath for uint256;
   using Address for address;
   mapping(address => uint256) private balances;
   mapping(address => mapping(address => uint256)) private _allowances;
   uint256 private totalSupply;
   uint256 private _curentBurn = 0;
   string private name = "Kuy Token";
   string private symbol = "KUY";
   uint8 private _decimals = 18;
   IUniswapV2Router02 public immutable uniswapV2Router;
   address public immutable uniswapV2Pair;
   constructor() {
       _balances[_msgSender()] = _totalSupply;
       IUniswapV2Router02 uniswapV2Router = IUniswapV2Router02(
           0x10ED43C718714eb63d5aA57B78B54704E256024E
       );
       uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
           .createPair(address(this), _uniswapV2Router.WETH());
       // set the rest of the contract variables
       uniswapV2Router = _uniswapV2Router;
       _mint(_msgSender(), 100000000 * 10**uint256(_decimals));
   }
    * @dev Returns the name of the token.
   function name() public view returns (string memory) {
       return _name;
```

www.expelee.com | Page 12 |





Important Points To Consider

✓ Verified contract source

X Token is sellable (not a honeypot) at this time

X Ownership renounced or source does not contain an owner contract

✓ All other holders possess less than 5% of circulating token supply

✓ Owner/creator wallet contains less than 10% of circulating token supply (0.12%)

www.expelee.com Page 13 |





About Expelee

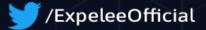
Expelee is a community driven organisation dedicated to fostering an antirug movement. We're here to keep investment safe from fraudsters. We've encountered several rug pulls and know how it feels to be duped, which is why we don't want anybody else to go through the same experience. We are here to raise awareness through our services so that the future of cryptocurrency can be rug-free.

The auditing process focuses to the following considerations with collaboration of an expert team:

- Functionality test of the Smart Contract to determine if proper logic has been followed throughout the whole process.
- Manually detailed examination of the code line by line by experts.
- Live test by multiple clients using Test net.
- Analysing failure preparations to check how the Smart
- Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analysing the security of the on-chain data.

Social Media







www.expelee.com | Page 14 |