exp̂elee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

MILADY BSC

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	**13 May 2023**

PROJECT DESCRIPTION

MILADY BSC ORIGINAL is the next trend of meme coin. You can make NFT via our DAPP powered by AI Technology. This is on BB chain!
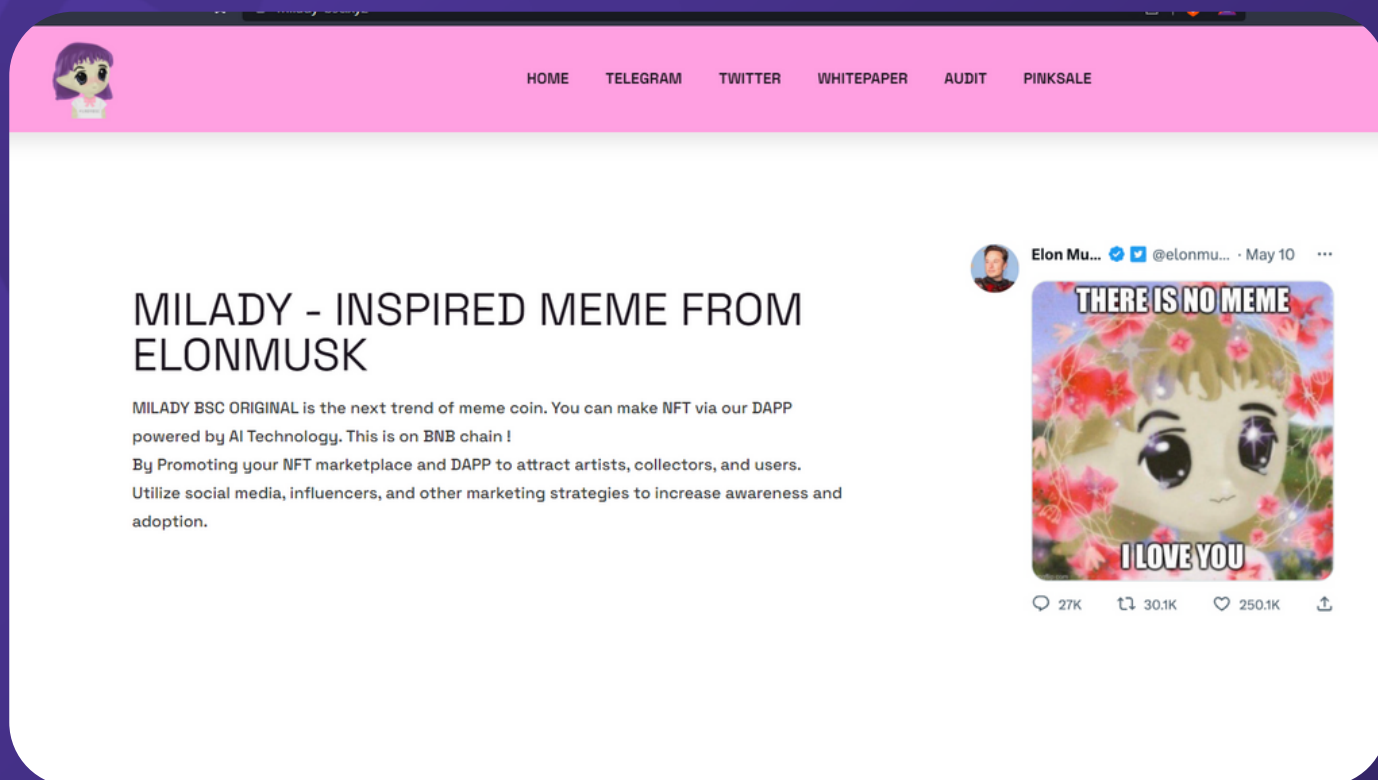
By Promoting your NFT marketplace and DAPP to attract artists, collectors, and users.
Utilize social media, influencers, and other marketing strategies to increase awareness and adoption.

expelee

SOCIAL MEDIA PROFILES

MILADY BSC

http://t.me/miladybsc_official

https://milady-bsc.xyz/

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

Team Expelee

expelee

CONTRACT DETAILS

Token Name: MILADY BSC ORIGINAL

Symbol: MILADY BSC ORIGINAL

Network: Binance smart chain

Language: Solidity

Contract Address:
0xa358ECCeC83B5828E7Df715D0bef60cbb6a0C179

Total Supply: 1,000,000,000

Contract SHA-256 Checksum: -
215245066649c0633901f576d206197f1c758d3f

Owner's Wallet:
0xe8218FaA1A1ABC1AFF3d9DDeDeFb4e53548e831E

Deployer's Wallet:
0xe8218FaA1A1ABC1AFF3d9DDeDeFb4e53548e831E

OWNER PRIVILEGES

- Contract owner is not able to set buy/sell/transfer fees more than 5%
- 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
- Contract owner must enable trades for holders manually
- Contract owner can change marketing, staking, and treasury wallets
- Contract owner can exclude an address from fees
- Contract owner can claim stuck tokens (except the contract's own tokens)
- Trading is disabled by default and can be enabled by the contract owner
- The contract has separate fees for buy, sell, and transfer operations
- The contract has separate wallets for marketing, staking, and treasury
- The contract has a swap and send function to swap tokens and send the resulting BNB to the respective wallets

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.
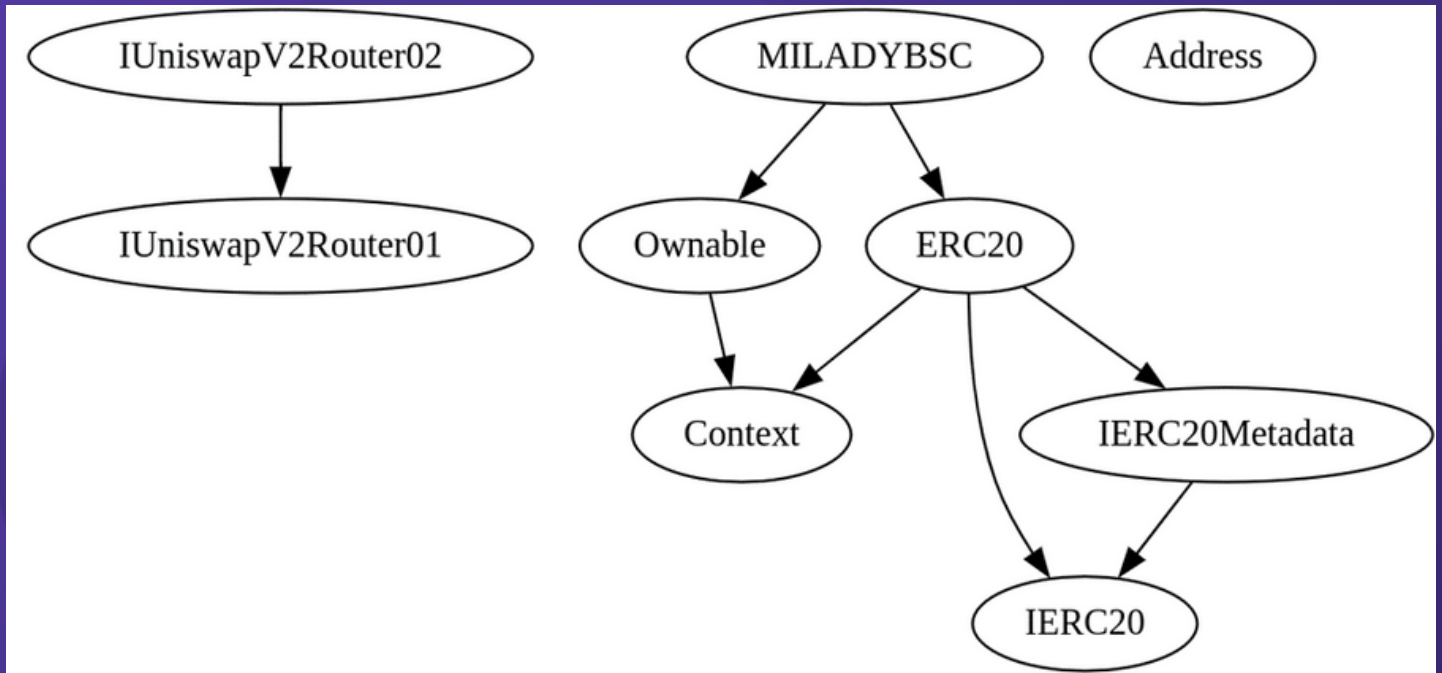
Low Risk

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

TESTNET VERSION

Adding Liquidity ✓
Tx:
https://testnet.bscscan.com/tx/0xb6c3a6d3843c813a8141c05368770c96ca455efef39897207edd8dd6114908fc

==

Buying from a fee excluded wallet ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x2aa1c9cef64b2f5368eee167396c668fa691e3183e3728425e0a90db83bcfa4a

==

Selling from a fee excluded wallet ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0xa24dfcdfbdea9cefd1bae802dabd0baddac8ec4aad179d9638aa2f3087f4528a

==

Transferring using a fee excluded wallet ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0xe2e583081c5e98cb2c343ca969e4b61329242bdbd1f899f9f0d61291cb64e76e

==

TESTNET VERSION

Buying from a regular wallet ✓
Tx (0–5% tax):
https://testnet.bscscan.com/tx/0x997361341900dc8d77da708524a56c746e2913ab8595b1f105fda2e73d670df8

==

Selling from a regular wallet ✓
Tx (0–5% tax):
https://testnet.bscscan.com/tx/0x30cceac0a6968c885986797bcf4fd7c3da67edb32f2806f1e9fed3c813bb5f15

==

Transferring a regular wallet ✓
Tx (0–5%):
https://testnet.bscscan.com/tx/0xf8dcb7c271f61eaef9fd13d186b508cce6ae73a0877aa91738797e0b39d439e3

==

Internal swap ✓
Tx:
https://testnet.bscscan.com/tx/0x30cceac0a6968c885986797bcf4fd7c3da67edb32f2806f1e9fed3c813bb5f15

==

FUNCTION DETAILS

Contract	Type	Bases			
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IUniswapV2Factory	Interface					
└	feeTo	External	❗		NO ❗	
└	feeToSetter	External	❗		NO ❗	
└	getPair	External	❗		NO ❗	
└	allPairs	External	❗		NO ❗	
└	allPairsLength	External	❗		NO ❗	
└	createPair	External	❗	🔴	|NO ❗	
└	setFeeTo	External	❗	🔴	|NO ❗	
└	setFeeToSetter	External	❗	🔴	|NO ❗	
IUniswapV2Pair	Interface					
└	name	External	❗		NO ❗	
└	symbol	External	❗		NO ❗	
└	decimals	External	❗		NO ❗	
└	totalSupply	External	❗		NO ❗	
└	balanceOf	External	❗		NO ❗	
└	allowance	External	❗		NO ❗	
└	approve	External	❗	🔴	|NO ❗	
└	transfer	External	❗	🔴	|NO ❗	
└	transferFrom	External	❗	🔴	|NO ❗	
└	DOMAIN_SEPARATOR	External	❗		NO ❗	
└	PERMIT_TYPEHASH	External	❗		NO ❗	
└	nonces	External	❗		NO ❗	

FUNCTION DETAILS

└	permit	External ❗		🔴	NO ❗
└	MINIMUM_LIQUIDITY	External ❗			NO ❗
└	factory	External ❗			NO ❗
└	token0	External ❗			NO ❗
└	token1	External ❗			NO ❗
└	getReserves	External ❗			NO ❗
└	price0CumulativeLast	External ❗			NO ❗
└	price1CumulativeLast	External ❗			NO ❗
└	kLast	External ❗			NO ❗
└	mint	External ❗		🔴	NO ❗
└	burn	External ❗		🔴	NO ❗
└	swap	External ❗		🔴	NO ❗
└	skim	External ❗		🔴	NO ❗
└	sync	External ❗		🔴	NO ❗
└	initialize	External ❗		🔴	NO ❗
IUniswapV2Router01	Interface			
└	factory	External ❗			NO ❗
└	WETH	External ❗			NO ❗
└	addLiquidity	External ❗		🔴	NO ❗
└	addLiquidityETH	External ❗		🟩	NO ❗
└	removeLiquidity	External ❗		🔴	NO ❗
└	removeLiquidityETH	External ❗		🔴	NO ❗
└	removeLiquidityWithPermit	External ❗		🔴	NO ❗
└	removeLiquidityETHWithPermit	External ❗		🔴	NO ❗
└	swapExactTokensForTokens	External ❗		🔴	NO ❗
└	swapTokensForExactTokens	External ❗		🔴	NO ❗
└	swapExactETHForTokens	External ❗		🟩	NO ❗
└	swapTokensForExactETH	External ❗		🔴	NO ❗
└	swapExactTokensForETH	External ❗		🔴	NO ❗
└	swapETHForExactTokens	External ❗		🟩	NO ❗
└	quote	External ❗			NO ❗
└	getAmountOut	External ❗			NO ❗
└	getAmountIn	External ❗			NO ❗
└	getAmountsOut	External ❗			NO ❗
└	getAmountsIn	External ❗			NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗		🔴	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗		🔴	NO ❗

└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗		🔴	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗		🟩	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗		🔴	NO ❗
IERC20	Interface			

FUNCTION DETAILS

```
|  └ | 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 ❗ |
||||||
| **Address** | Library | |||
|  └ | isContract | Internal 🔒 |  | ||
|  └ | sendValue | Internal 🔒 | ⬤ ||
|  └ | functionCall | Internal 🔒 | ⬤ ||
|  └ | functionCall | Internal 🔒 | ⬤ ||
|  └ | functionCallWithValue | Internal 🔒 | ⬤ ||
|  └ | functionCallWithValue | Internal 🔒 | ⬤ ||
|  └ | functionStaticCall | Internal 🔒 |  | ||
|  └ | functionStaticCall | Internal 🔒 |  | ||
|  └ | functionDelegateCall | Internal 🔒 | ⬤ ||
|  └ | functionDelegateCall | Internal 🔒 | ⬤ ||
|  └ | verifyCallResultFromTarget | Internal 🔒 |  | ||
|  └ | verifyCallResult | Internal 🔒 |  | ||
|  └ | _revert | Private 🔒 |  | ||
||||||
| **Context** | Implementation | |||
|  └ | _msgSender | Internal 🔒 |  | ||
|  └ | _msgData | Internal 🔒 |  | ||
||||||
| **Ownable** | Implementation | Context |||
|  └ | <Constructor> | Public ❗ | ⬤ |NO ❗ |
|  └ | owner | Public ❗ |  | |NO ❗ |
|  └ | renounceOwnership | Public ❗ | ⬤ | onlyOwner |
|  └ | transferOwnership | Public ❗ | ⬤ | onlyOwner |
||||||
| **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 ❗ |
```

FUNCTION DETAILS

```
| └ | approve | Public ❗ | | 🔴 |NO ❗ |
| └ | transferFrom | Public ❗ | | 🔴 |NO ❗ |
| └ | increaseAllowance | Public ❗ | | 🔴 |NO ❗ |
| └ | decreaseAllowance | Public ❗ | | 🔴 |NO ❗ |
| └ | _transfer | Internal 🔒 | | 🔴 ||
| └ | _mint | Internal 🔒 | | 🔴 ||
| └ | _burn | Internal 🔒 | | 🔴 ||
| └ | _approve | Internal 🔒 | | 🔴 ||
| └ | _beforeTokenTransfer | Internal 🔒 | | 🔴 ||
| └ | _afterTokenTransfer | Internal 🔒 | | 🔴 ||
||||||
| **MILADYBSC** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗ | | 🔴 | ERC20 |
| └ | <Receive Ether> | External ❗ | | 💵 |NO ❗ |
| └ | claimStuckTokens | External ❗ | | 🔴 | onlyOwner |
| └ | excludeFromFees | External ❗ | | 🔴 | onlyOwner |
| └ | isExcludedFromFees | Public ❗ | | |NO ❗ |
| └ | updateFees | External ❗ | | 🔴 | onlyOwner |
| └ | changeMarketingWallet | External ❗ | | 🔴 | onlyOwner |
| └ | changeStakingWallet | External ❗ | | 🔴 | onlyOwner |
| └ | changeTreasuryWallet | External ❗ | | 🔴 | onlyOwner |
| └ | enableTrading | External ❗ | | 🔴 | onlyOwner |
| └ | _transfer | Internal 🔒 | | 🔴 ||
| └ | setSwapEnabled | External ❗ | | 🔴 | onlyOwner |
| └ | setSwapTokensAtAmount | External ❗ | | 🔴 | onlyOwner |
| └ | swapAndSend | Private 🔒 | | 🔴 ||
```

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

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	1
Low Risk	🟡 Low	0
Suggestion & discussion	🔵 Informational	0
Gas Optimizations	🟣 Gas Opt.	0

HIGH RISK FINDING

Category: Centralization
Subject: Enabling trades is not guaranteed
Issue Status: Resolved (Contract is owned by safu dev for 14 days)
Severity : High
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 enableTrading() external onlyOwner {
require(!tradingEnabled, "Trading already enabled.");
tradingEnabled = true;
swapEnabled = true;
  }
```

Suggestion
To mitigate this issue there are several options:
- Enable tradings before presale
- Transfer ownership to safu dev for initial days after presale

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

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