expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

BOCAT

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	0
Medium	0
Low	2
Informational	1

Centralization Risks

Owner Privileges	Description
Can Owner Set Taxes >25% ?	Not Detected
Owner needs to enable trading ?	Not Detected
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**
Audit Date	**29 April 2024**

CONTRACT DETAILS

Token Address: 0x5e1773eB46E74F3a3511fdfE4ba730B3B50411E5

Name: BOCAT

Symbol: BOCAT

Decimals: 18

Network: BaseScan

Token Type: ERC-20

Owner: 0xD5cFB25EB5f6c608ada6579cFDe0E93Ac2eECFb6

Deployer: 0x8e387BbeF29E8c911B2902bA3f4615F11bEe4b19

Token Supply: 1,000,000,000

Checksum: AEde641126e217b2b455d49e77fc41221

Testnet:
https://testnet.bscscan.com/address/0x78a952bef8f19f409705f10ff1ba44153def24b5#code

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

- Manual Review: The code has undergone a line-by-line review by the Ace team.
- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.
- Slither: The code has undergone static analysis using Slither.

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.

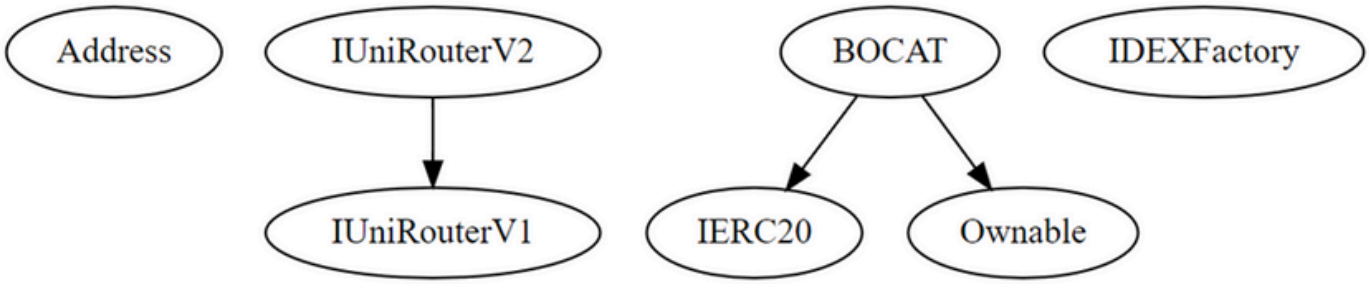
Low Risk

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 TREE



expelee

STATIC ANALYSIS

```
INFO:Detectors:
BOCAT._approve(address,address,uint256).owner (BOCAT.sol#452) shadows:
        - Ownable.owner() (BOCAT.sol#249-251) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
BOCAT.ownerSetSwapThreshold(uint256) (BOCAT.sol#391-397) should emit an event for:
        - _swapTokenThreshold = swapTokenThreshold * 10 ** _decimals (BOCAT.sol#396)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Reentrancy in BOCAT._transfer(address,address,uint256) (BOCAT.sol#310-327):
        External calls:
        - _swapContractTokens() (BOCAT.sol#323)
                - _router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BOCAT.sol#359-365)
                - (transferMarketing) = address(MARKETING).call{gas: 30000,value: address(this).balance}() (BOCAT.sol#347)
        External calls sending eth:
        - _swapContractTokens() (BOCAT.sol#323)
                - (transferMarketing) = address(MARKETING).call{gas: 30000,value: address(this).balance}() (BOCAT.sol#347)
        Event emitted after the call(s):
        - Transfer(from,to,amount - taxAmount) (BOCAT.sol#338)
                - _transferTokens(from,to,amount,sellTax) (BOCAT.sol#324)
Reentrancy in BOCAT.transferFrom(address,address,uint256) (BOCAT.sol#439-450):
        External calls:
        - _transfer(sender,recipient,amount) (BOCAT.sol#445)
                - _router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BOCAT.sol#359-365)
                - (transferMarketing) = address(MARKETING).call{gas: 30000,value: address(this).balance}() (BOCAT.sol#347)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (BOCAT.sol#445)
                - (transferMarketing) = address(MARKETING).call{gas: 30000,value: address(this).balance}() (BOCAT.sol#347)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BOCAT.sol#458)
                - _approve(sender,msg.sender,allowance_ - amount) (BOCAT.sol#447)
        - Transfer(sender,recipient,amount) (BOCAT.sol#448)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Address.isContract(address) (BOCAT.sol#9-15) uses assembly
        - INLINE ASM (BOCAT.sol#11-13)
Address._verifyCallResult(bool,bytes,string) (BOCAT.sol#73-90) uses assembly
        - INLINE ASM (BOCAT.sol#82-85)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
INFO:Detectors:
Function IUniRouterV1.WETH() (BOCAT.sol#95) is not in mixedCase
Constant BOCAT.buyTax (BOCAT.sol#268) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BOCAT.sellTax (BOCAT.sol#269) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BOCAT._decimals (BOCAT.sol#270) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BOCAT._totalSupply (BOCAT.sol#272) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BOCAT._tokenName (BOCAT.sol#275) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BOCAT._tokenSymbol (BOCAT.sol#276) is not in UPPER_CASE_WITH_UNDERSCORES
Modifier BOCAT.LockTheSwap() (BOCAT.sol#293-297) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable IUniRouterV1.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (BOCAT.sol#100) is too similar to IUniRou
terV1.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (BOCAT.sol#101)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
BOCAT.ownerSetSwapThreshold(uint256) (BOCAT.sol#391-397) uses literals with too many digits:
        - require(bool,string)(swapTokenThreshold <= 5000000,Cannot exceed 50 billion.) (BOCAT.sol#395)
BOCAT.slitherConstructorVariables() (BOCAT.sol#266-495) uses literals with too many digits:
        - _swapTokenThreshold = 5000000 * 10 ** _decimals (BOCAT.sol#273)
BOCAT.slitherConstructorConstantVariables() (BOCAT.sol#266-495) uses literals with too many digits:
        - _totalSupply = 1000000000 * 10 ** _decimals (BOCAT.sol#272)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BOCAT._pairAddress (BOCAT.sol#283) should be immutable
BOCAT._router (BOCAT.sol#282) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:BOCAT.sol analyzed (7 contracts with 93 detectors), 42 result(s) found
```

TESTNET VERSION

1- **Approve (passed):**
https://testnet.bscscan.com/tx/0xfa2b70bd3122b67668d8c12ee14a86c5b8b4dde4eec5668f08610f4f9598556c

2- Owner Enable Trading (passed):
https://testnet.bscscan.com/tx/0x5944b3b8116f18ecac5a58ec3e5f8499d7512cecfdaf609eeef54ee1aaaa6ca4

3- Owner Exclude from Fee (passed):
https://testnet.bscscan.com/tx/0xb4b0656a2a63f5ffb7e4e968b791dab2a57f60d1c73bbdd5f4cc613a9c4e87ba

4- Transfer Ownership (passed):
https://testnet.bscscan.com/tx/0x0c9d6357420d05406f590918cb45eb5f6140c0d168e11c6678de5352303273aa

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			

LOW RISK FINDING

Centralization – Missing Events

Severity: Low

Subject: Missing Events

Status: Open

Overview:

They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function ownerSetSwapThreshold(
    uint256 swapTokenThreshold
  ) public onlyOwner {
    require(_swapTokenThreshold > 0, "Must be greater than zero.");
    require(swapTokenThreshold <= 5000000, "Cannot exceed 50
billion.");
    _swapTokenThreshold = swapTokenThreshold * 10 ** _decimals;
  }
```

LOW RISK FINDING

Centralization – Local Variable Shadowing

Severity: Low

Status: Open

Function: _approve and allowance

Overview:

```
function _approve(
    address owner,
    address spender,
    uint256 amount
  ) private {
    require((owner != address(0) && spender != address(0)),
"Owner/Spender address cannot be 0.");
    _allowances[owner][spender] = amount;
    emit Approval(owner, spender, amount);
  }
```

Suggestion:

Rename the local variable that shadows another component.

INFORMATIONAL & OPTIMIZATIONS

Optimization

Severity: Optimization

Subject: Remove unused code.

Status: Open

Overview:
Unused variables are allowed in Solidity, and they do. not pose a direct security issue. It is the best practice though to avoid them.

```
function sendValue(address payable recipient, uint256 amount)
internal {
    require(address(this).balance >= amount, "Address: insufficient
balance");
    (bool success, ) = recipient.call{value: amount}("");
    require(success, "Address: unable to send value, recipient may have
reverted");
  }
function functionCall(address target, bytes memory data) internal
returns (bytes memory) {
    return functionCall(target, data, "Address: low-level call failed");
  }
function functionCallWithValue(
    address target,
    bytes memory data,
    uint256 value
  ) internal returns (bytes memory) {
    return functionCallWithValue(target, data, value, "Address: low-level
call with value failed");
  }
```

INFORMATIONAL & OPTIMIZATIONS

```
function functionStaticCall(address target, bytes memory data) internal
view returns (bytes memory) {
    return functionStaticCall(target, data, "Address: low-level static call
failed");
  }
function functionDelegateCall(address target, bytes memory data)
internal returns (bytes memory) {
    return functionDelegateCall(target, data, "Address: low-level delegate
call failed");
  }
```

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.