expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Fintyhub

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	0
Medium	2
Low	3
Informational	1

Centralization Risks

Owner Privileges	Description
Can Owner Set Taxes >25% ?	Not Detected
Owner Can 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

expelee

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 Jan, 2024**

CONTRACT DETAILS

Token Name: Fintyhub

Symbol: FTH

Network: BscScan

Decimal: 18

Token Type: BEP - 20

Token Address:
0x3228Ad31679e826E9efE1CAce04Fc0FB85eD482F

Total Supply: 1,000,000,000

Owner's Wallet:
0x76ACA1B7e0d40473D41F2242B3Eb0f38a09bA203

Deployer's Wallet:
0x76ACA1B7e0d40473D41F2242B3Eb0f38a09bA203

CheckSum:
Abff351e1897814d20411ecffceb9017

Testnet.
https://testnet.bscscan.com/address/0x0b4958ee572172f6b83556abdc4e1d28d2571717#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

- 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.

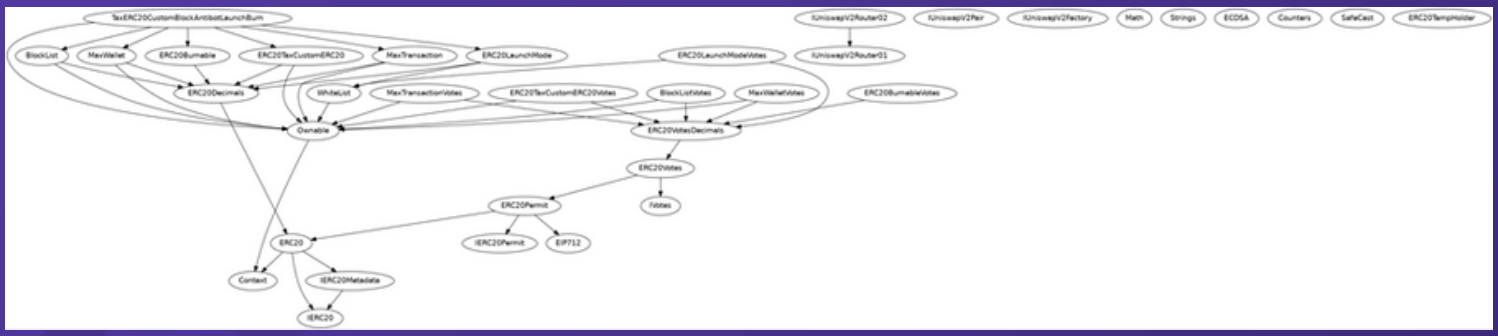
Low Risk

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



STATIC ANALYSIS

```
INFO:Detectors:
ERC20Permit.constructor(string).name (TaxERC20.sol#1615) shadows:
        - ERC20.name() (TaxERC20.sol#188-190) (function)
        - IERC20Metadata.name() (TaxERC20.sol#99) (function)
ERC20Burnable.burnFrom(address,uint256).allowance (TaxERC20.sol#4434) shadows:
        - ERC20.allowance(address,address) (TaxERC20.sol#248-250) (function)
        - IERC20.allowance(address,address) (TaxERC20.sol#54) (function)
ERC20BurnableVotes.burnFrom(address,uint256).allowance (TaxERC20.sol#4452) shadows:
        - ERC20.allowance(address,address) (TaxERC20.sol#248-250) (function)
        - IERC20.allowance(address,address) (TaxERC20.sol#54) (function)
TaxERC20CustomBlockAntibotLaunchBurn.constructor(uint256,uint8,string,string,address,uint256,uint256,uint64,TaxConstructorArgs)._decimals (TaxERC20.sol#4480
) shadows:
        - ERC20Decimals._decimals (TaxERC20.sol#3131) (state variable)
TaxERC20CustomBlockAntibotLaunchBurn.constructor(uint256,uint8,string,string,address,uint256,uint256,uint64,TaxConstructorArgs)._name (TaxERC20.sol#4481) sh
adows:
        - ERC20._name (TaxERC20.sol#168) (state variable)
TaxERC20CustomBlockAntibotLaunchBurn.constructor(uint256,uint8,string,string,address,uint256,uint256,uint64,TaxConstructorArgs)._symbol (TaxERC20.sol#4482)
shadows:
        - ERC20._symbol (TaxERC20.sol#169) (state variable)
TaxERC20CustomBlockAntibotLaunchBurn.constructor(uint256,uint8,string,string,address,uint256,uint256,uint64,TaxConstructorArgs)._owner (TaxERC20.sol#4483) s
hadows:
        - Ownable._owner (TaxERC20.sol#721) (state variable)
TaxERC20CustomBlockAntibotLaunchBurn.constructor(uint256,uint8,string,string,address,uint256,uint256,uint64,TaxConstructorArgs)._maxWallet (TaxERC20.sol#448
4) shadows:
        - MaxWallet._maxWallet (TaxERC20.sol#4047) (state variable)
TaxERC20CustomBlockAntibotLaunchBurn.constructor(uint256,uint8,string,string,address,uint256,uint256,uint64,TaxConstructorArgs)._maxTransaction (TaxERC20.so
l#4485) shadows:
        - MaxTransaction._maxTransaction (TaxERC20.sol#4157) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
```

```
INFO:Detectors:
ERC20TaxCustomERC20Votes.setMinTokensBeforeSwap(uint256) (TaxERC20.sol#3821-3825) should emit an event for:
        - minTokensBeforeSwap = _minTokensBeforeSwap (TaxERC20.sol#3824)
MaxWalletVotes.setMaxWallet(uint256) (TaxERC20.sol#4126-4128) should emit an event for:
        - _maxWallet = _maxWalletAmount (TaxERC20.sol#4127)
MaxTransactionVotes.setMaxTransaction(uint256) (TaxERC20.sol#4262-4264) should emit an event for:
        - _maxTransaction = _maxTransactionAmount (TaxERC20.sol#4263)
MaxTransactionVotes.setPeriod(uint256) (TaxERC20.sol#4270-4272) should emit an event for:
        - _period = _periodInSeconds (TaxERC20.sol#4271)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
ERC20TaxCustomERC20Votes.setCharityAddress(address)._charityAddress (TaxERC20.sol#3749) lacks a zero-check on :
                - charityAddress = _charityAddress (TaxERC20.sol#3750)
ERC20TaxCustomERC20Votes.setMarketingAddress(address)._marketingAddress (TaxERC20.sol#3753) lacks a zero-check on :
                - marketingAddress = _marketingAddress (TaxERC20.sol#3754)
ERC20TaxCustomERC20Votes.setAutoLiquidityAddress(address)._autoLiquidityAddress (TaxERC20.sol#3758) lacks a zero-check on :
                - autoLiquidityAddress = _autoLiquidityAddress (TaxERC20.sol#3760)
ERC20TaxCustomERC20.setCharityAddress(address)._charityAddress (TaxERC20.sol#3366) lacks a zero-check on :
                - charityAddress = _charityAddress (TaxERC20.sol#3367)
ERC20TaxCustomERC20.setMarketingAddress(address)._marketingAddress (TaxERC20.sol#3370) lacks a zero-check on :
                - marketingAddress = _marketingAddress (TaxERC20.sol#3371)
ERC20TaxCustomERC20.setAutoLiquidityAddress(address)._autoLiquidityAddress (TaxERC20.sol#3375) lacks a zero-check on :
                - autoLiquidityAddress = _autoLiquidityAddress (TaxERC20.sol#3377)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```

```
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (TaxERC20.sol#524) is too similar t
o IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (TaxERC20.sol#525)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Slither:TaxERC20.sol analyzed (36 contracts with 93 detectors), 230 result(s) found
```

TESTNET VERSION

1- Approve (passed):
https://testnet.bscscan.com/tx/0xbbc373450e4370af039182fbe42de43a9f911a40fb636e388c9799dbc897f1af

2- Increase Allowance (passed):
https://testnet.bscscan.com/tx/0x907ec4299025c7da5063c010d625d18d721ac47f4725128ee066326358e1fcc3

3- Decrease Allowance (passed):
https://testnet.bscscan.com/tx/0x123c5da31b6ed1e084d5e25158a002c71e3e04a0348a59e2c677ad222ab51aac

4- Set Marketing Address (passed):
https://testnet.bscscan.com/tx/0xd2fd165061f970f1c8dc8ebf352bc6f378bb1b53f145dafcf1476e6448a76131

5- Set Charity Address (passed):
https://testnet.bscscan.com/tx/0xbd869eb2f47e72d33a8314e1ad68f753d17f77001f549dce94c665a0c782a3e2

6- Set Sell Tax (passed):
https://testnet.bscscan.com/tx/0x3f953f7217ba4dce8fb5176417631c80cd09c505782b3a088403644e000d301e

7- Set Buy Tax (passed):
https://testnet.bscscan.com/tx/0xa8c2143920e91a7b10dfa7c49ddfcd6385df68a59ff9c76bbe97204374f4affc

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			

MEDIUM RISK FINDING

Liquidity is added to EOA.

Category: Centralization
Severity: Medium
Function: addLiquidityETH/addLiquidityERC20
Status:Open

Overview:
Liquidity is added to EOA. It may be drained by the autoLiquidityAddress.

```
ffunction addLiquidityETH(
uint256 _tokenAmount,
uint256 _NativeAmount
  ) private returns (uint, uint, uint) {
return
     uniswapV2Router.addLiquidityETH{value: _NativeAmount}(
address(this),
       _tokenAmount,
0,
0,
       autoLiquidityAddress,
       block.timestamp
     );
  }
 function addLiquidityERC20(
uint256 _tokenAmount,
uint256 _counterPartAmount
  ) private returns (uint, uint, uint) {
return
```

MEDIUM RISK FINDING

```
 uniswapV2Router.addLiquidity(
address(this),
 mainLPToken,
 _tokenAmount,
 _counterPartAmount,
0,
0,
 autoLiquidityAddress,
 block.timestamp
 );
 }
```

Suggestion:
It is suggested that the address should be a contract address or a dead address.

MEDIUM RISK FINDING

Missing Require Check.

Category: Centralization
Severity: Medium
Function: setMarketingAddress
Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner will set the address to the contract address, then the Eth will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setMarketingAddress(address _marketingAddress)
external onlyOwner {
        marketingAddress = _marketingAddress;
    }
```

Suggestion:

It is recommended that the address should not be able to be set as a contract address.

LOW RISK FINDING

Missing Events

Category: Centralization
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 setCharityAddress(address _charityAddress)
external onlyOwner {
    charityAddress = _charityAddress;
  }
function setMarketingAddress(address _marketingAddress)
external onlyOwner {
    marketingAddress = _marketingAddress;
  }
 function setTransferTax(
uint64 _charity,
uint64 _autoLiquidity,
uint64 _marketing
  ) external onlyOwner {
```

LOW RISK FINDING

```
require(
 _charity + _autoLiquidity + _marketing <= MAX_TAX,
"Transfer tax too high"
 );
 transferTax = TaxInfo(_charity, _autoLiquidity, _marketing);
 }
}
function setBuyTax(
uint64 _charity,
uint64 _autoLiquidity,
uint64 _marketing
  ) external onlyOwner {
require(
      _charity + _autoLiquidity + _marketing <= MAX_TAX,
"Buy tax too high"
    );
    buyTax = TaxInfo(_charity, _autoLiquidity, _marketing);
  }
function setSellTax(
uint64 _charity,
uint64 _autoLiquidity,
uint64 _marketing
  ) external onlyOwner {
require(
      _charity + _autoLiquidity + _marketing <= MAX_TAX,
"Sell tax too high"
```

LOW RISK FINDING

```
 );
 sellTax = TaxInfo(_charity, _autoLiquidity, _marketing);
 }
 function setMinTokensBeforeSwap(
uint256 _minTokensBeforeSwap
 ) external onlyOwner {
 minTokensBeforeSwap = _minTokensBeforeSwap;
 }
function setMaxTransaction(uint256
_maxTransactionAmount) public onlyOwner {
 _maxTransaction = _maxTransactionAmount;
 }
function setMaxWallet(uint256 _maxWalletAmount) public
onlyOwner {
 _maxWallet = _maxWalletAmount;
 }
 function setPeriod(uint256 _periodInSeconds) public
onlyOwner {
 _period = _periodInSeconds;
 }
```

LOW RISK FINDING

Local Variable Shadowing

Category: Centralization
Severity: Low
Subject: Variable Shadowing.
Status: Open
Overview:

```
constructor(
uint256 _initialSupply,
uint8 _decimals,
string memory _name,
string memory _symbol,
address _owner,
uint256 _maxWallet,
uint256 _maxTransaction,
uint64 _periodInSec,
    TaxConstructorArgs memory _args
  )
    ERC20TaxCustomERC20(_args)
    ERC20Decimals(_decimals)
    ERC20(_name, _symbol)
    Ownable(_owner)
    MaxWallet(_maxWallet)
    MaxTransaction(_periodInSec, _maxTransaction)
  {
    _mint(_owner, _initialSupply);
```

Suggestion:
Rename the local variables that shadow another

LOW RISK FINDING

Category: Centralization
Severity: Low
Status: Open
Overview:
Functions can take a zero address as a parameter (0x00000...). If a function parameter of address type is not properly validated by checking for zero addresses, there could be serious consequences for the contract's functionality.

```
function setCharityAddress(address _charityAddress)
external onlyOwner {
    charityAddress = _charityAddress;
  }
function setMarketingAddress(address
_marketingAddress) external onlyOwner {
    marketingAddress = _marketingAddress;
  }
function setAutoLiquidityAddress(
address _autoLiquidityAddress
  ) external onlyOwner {
    autoLiquidityAddress = _autoLiquidityAddress;
  }
```

Suggestion:
It is suggested that the address should not be zero or dead.

INFORMATIONAL RISK FINDING

Category: Optimization
Severity: Informational
Severity: Low
Subject: floating Pragma
Status: Open
Overview:
It is considered best practice to pick one compiler version and stick with it. With a floating pragma, contracts may accidentally be deployed using an outdated.

pragma solidity ^0.8.18;

Suggestion:
Adding the latest constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.

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.

expelee

Building the Futuristic Blockchain Ecosystem