



# expelee

A Secure Place For Web3

### **SMART CONTRACT AUDIT OF**

## FitApe Presale



**Contract Address** 

0x63a472ec95527dcb79A3aA23170b9ece00A8Ad64

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 (Medium Risk Severity)** 

**Ownership: NOT RENOUNCED** 

KYC Verification: Not done till date of audit

Audit Date: 011/06/2022

**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	FitApe
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	No with 200 runs
License	MIT license
Explorer	https://bscscan.com/address/0x63a47 2ec95527dcb79A3aA23170b9ece00A8A d64#code
Symbol	FitApe
Decimals	9
Total Supply	10,000,000,000,000
Domain	https://fitape.tech/

www.expelee.com | Page 4 |





# **Project Review**

Token Name: FitApe

Web Site: https://fitape.tech/

Twitter: https://twitter.com/FitApeToken

Telegram: https://t.me/FitApe

**Contract Address:** 

0x63a472ec95527dcb79A3aA23170b9ece00A8Ad64

**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

- Unhandled Exceptions

- Transaction Order Dependency

Smart Contract Vulnerabilities - 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**

Nō	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
- 6 low-risk code issues found
  - Medium-Risk
- 1 medium-risk code issues found
  - High-Risk
  - 0 high-risk code issues found

www.expelee.com Page 8 |





#### 1) Contract contains Reentrancy vulnuerabilities

```
function _transfer(address from, address to, uint256 amount) private {require(from
!= address(0), "ERC20: transfer from the zero address"); require(to != address(0),
"ERC20: transfer to the zero address");
require(amount > 0, "Transfer amount must be greater than zero");
require(!_isBlacklisted[from], 'Blacklisted address');
redisFee = 0;
_taxFee = 0;
// No adding liquidity before launchedif
(!liquidityLaunched) {
    if (to == uniswapV2Pair) {
        liquidityLaunched = true;
        lastSnipeTaxBlock = block.number + snipeBlocks;
        swapEnablelastTime = block.timestamp + swapEnableLockTime;
   }
}
//antibot block
if (from != address(_PresaleAddress)) {
    if(liquidityLaunched & amp; & amp; block.number 0) {
```

#### Recommendation

Apply the check-effects-interaction pattern

www.expelee.com | Page 9 |





#### 2) No zero address validation for some functions

Detect missing zero address validation.

```
event appAddressUpdated(address indexed previous, address indexed adr);function
setNewAppAddress(address payable appaddr) public onlyOwner {
   emit appAddressUpdated(_appAddress, appaddr);
   _appAddress = appaddr;
   _isExcludedFromFee[_appAddress] = true;
}
event marketingAddressUpdated(address indexed previous, address indexed adr);function
setNewMarketingAddress(address payable markt) public onlyOwner {
   emit marketingAddressUpdated(_marketingAddress, markt);
   _marketingAddress = markt;
   _isExcludedFromFee[_marketingAddress] = true;
}
event burnAddressUpdated(address indexed previous, address indexed adr);function
setNewBurnAddress(address payable burnaddr) public onlyOwner {
   emit burnAddressUpdated( burnAddress, burnaddr);
   _burnAddress = burnaddr;
    _isExcludedFromFee[_burnAddress] = true;
```

#### Recommendation

Check that the new address is not zero.

www.expelee.com Page 10 |



#### 3) Missing events arithmethics

Detect missing events for critical arithmetic parameters .

#### Recommendation

Emit an event for critical parameter changes.

www.expelee.com | Page 11 |



#### 4) Functions that send Ether to arbitary destinations

Unprotected call to a function sending Ether to arbitary address.

```
function sendETHToFee(uint256 amount) private {uint256
    devAmount = amount.div(2);

uint256 mktAmount = amount.mul(3).div(8);

uint256 burnAmount = amount.sub(devAmount).sub(mktAmount);

_appAddress.transfer(devAmount);

_marketingAddress.transfer(mktAmount);

_burnAddress.transfer(burnAmount);
}
```

#### Recommendation

Ensure that an arbitary user cannot withdraw unauthorized funds

www.expelee.com | Page 12 |



#### 5) Avoid relying on block.timestamp

block.timestamp can be manipulated by miners.

```
// No adding liquidity before launchedif
(!liquidityLaunched) {
   if (to == uniswapV2Pair) {
        liquidityLaunched = true;
        // high tax ends in x blocks
        lastSnipeTaxBlock = block.number + snipeBlocks;
        swapEnablelastTime = block.timestamp + swapEnableLockTime;
   }
}
```

#### Recommendation

Do not use block.timestamp, now or blockhash as a source of randomness

www.expelee.com | Page 13 |





#### 6) Unchecked Transfer

```
function rescueForeignTokens(address _tokenAddr, address _to, uint _amount) public onlyOwner {emit
    tokensRescued(_tokenAddr, _to, _amount);
    IERC20(_tokenAddr).transfer(_to, _amount);
}
```

#### Recommendation

Use SafeERC20, or ensure that the transfer/transferFrom return value is checked.

www.expelee.com | Page 14 |







#### 1) Owner can change Burn address

```
event burnAddressUpdated(address indexed previous, address indexed adr);function
setNewBurnAddress(address payable burnaddr) public onlyOwner {
    emit burnAddressUpdated(_burnAddress, burnaddr);
    _burnAddress = burnaddr;
    _isExcludedFromFee[_burnAddress] = true;
}
```

#### Recommendation

There is no need to be able to change the burn address. Hard-code the burn address for more transperency

www.expelee.com Page 15 |





# Manual Audit (Contract Function)

```
contract FitApe is Context, IERC20, Ownable {
using SafeMath for uint256;
mapping (address => uint256) private _rOwned;
mapping (address => mapping (address => uint256)) private _allowances; mapping
(address => bool) private _isExcludedFromFee;
mapping(address => bool) public _isBlacklisted;
liquidityLaunched = false;
bool public isFirstLaunch = true; uint256
public
        lastSnipeTaxBlock; uint8
                                   public
snipeBlocks = 0;
uint256 private constant MAX = ~uint256(0);
uint256 private constant tTotal = 10000 * (10**12) * (10**9);uint256
private _rTotal = (MAX - (MAX % _tTotal));
uint256 private _redisFeeOnBuy = 2;uint256
private taxFeeOnBuy = 8;
uint256 private _redisFeeOnSell = 2;uint256
private _taxFeeOnSell = 8;
uint256 private redisFee;uint256
private _taxFee;
string private constant _name = "FitApe"; string
private constant _symbol = "FitApe";uint8 private
constant _decimals = 9;
```

www.expelee.com Page 16 |



## Important Points To Consider

- X Source does not contain blacklist capability
  - ✓ The owner cannot stop Trading.
    - ✓ Verified contract source
- √ Token is sellable (not a honeypot) at this time
- X Ownership renounced or source does not contain an owner contract
  - X Source does not contain a fee modifier

The source code contains a function which can modify the transaction fee.

- ✓ Buy fee is less than 10% (10%)
- X Sell fee is less than 10% (24%)
- ✓ Owner/creator wallet contains less than 10% of circulating token supply (0.01%)

www.expelee.com | Page 17 |





# **About Expelee**

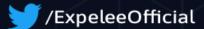
Expelee is a community driven organisation dedicated to fostering an antirug movement. We're here to keep investment safe from fraudsters. We've encountered several rug pulls and know how it feels to be duped, which is why we don't want anybody else to go through the same experience. We are here to raise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

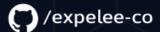
The auditing process focuses to the following considerations with collaboration of an expert team:

- 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 18 |