APESALE

Smart Contract Security Audit

Project: BatmanInu

Website: *https://batmaninu.me*

🥝 **Low-Risk**

3 low-risk code issues found

🍌 **Medium-Risk**

1 medium-risk code issue found

🔴 **High-Risk**

0 high-risk code issue found

Contract Address
0x518424b7F053eb3f0DC6A854c25Efdf59e838EDD

Disclaimer

Tokenomics

Rank	Address	Quantity (Token)
1	0x518424b7F053eb3f0DC6A854c25Efdf59e838EDD	1,000,000,000

Source Code

ApeSale was comissioned by BatmanInu to perform an audit based on the following smart contract:

https://bscscan.com/address/0x518424b7F053eb3f0DC6A854c25Efdf59e838EDD#code

🍌 **Medium-Risk:** Could be fixed, will not bring problems.

Contract can swap all fee token balance to BNB

Additional information: Contract can swap all fee token balance to BNB, this can cause a dump when variable "swapEnabled" is true and the token balance in the contract is large.

```
uint256 contractTokenBalance = balanceOf(address(this));
if (!inSwap && from != uniswapV2Pair && swapEnabled && contractTokenBalance > 0) {
    swapTokensForEth(contractTokenBalance);
    uint256 contractETHBalance = address(this).balance;
    if(contractETHBalance > 0) {
        sendETHToFee(address(this).balance);
    }
}
```

Recommendation
Swap a certain amount of tokens slowly and cannot be more than 0.5% of total supply

🍃 **Low-Risk:** Could be fixed, will not bring problems.

Several functions are declared public, but are never called internally:

rescureForeignToken, setNewDevAddress, setNewMarketingAddress, setFee, setBot, removeBot, setMM, excludeMultipleAccountsFromFees

Recommendation
These functions should be declared external for additional gas savings on each call.

🍃 Low-Risk: Could be fixed, will not bring problems.

No zero address validation for some functions

Detect missing zero address validation.

```solidity
event devAddressUpdated(address indexed previous, address indexed adr);
function setNewDevAddress(address payable dev) public onlyDev() {
    emit devAddressUpdated(_developmentAddress, dev);
    _developmentAddress = dev;
    _isExcludedFromFee[_developmentAddress] = true;
}

event marketingAddressUpdated(address indexed previous, address indexed adr);
function setNewMarketingAddress(address payable markt) public onlyDev() {
    emit marketingAddressUpdated(_marketingAddress, markt);
    _marketingAddress = markt;
    _isExcludedFromFee[_marketingAddress] = true;
}
```

Recommendation

Check that the new address is not zero.

Exploit scenario

```solidity
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
```

Bob calls updateOwnerwithout specifying the newOwner, soBob loses ownership of the contract.

🍃 **Low-Risk:** Could be fixed, will not bring problems.

Shadowing of inherited state variables should be an error

Detect inherited state variables owner at line 105.

```
function _approve(address owner, address spender, uint256 amount) private {
    require(owner != address(0), "ERC20: approve from the zero address");
    require(spender != address(0), "ERC20: approve to the zero address");
    _allowances[owner][spender] = amount;
    emit Approval(owner, spender, amount);
}
```

```
function allowance(address owner, address spender) public view override returns (uint256) {
    return _allowances[owner][spender];
}
```

```
function owner() public view returns (address) {
    return _owner;
}
```

Recommendation

Change variable "owner" of "_approve" and "allowance" functions

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗️	|NO❗️	
└	balanceOf	External ❗️	|NO❗️	
└	transfer	External ❗️	🔴 |NO❗️	
└	allowance	External ❗️	|NO❗️	
└	approve	External ❗️	🔴 |NO❗️	
└	transferFrom	External ❗️	🔴 |NO❗️	
Token	Interface			
└	transferFrom	External ❗️	🔴 |NO❗️	
└	transfer	External ❗️	🔴 |NO❗️	
IUniswapV2Factory	Interface			
└	createPair	External ❗️	🔴 |NO❗️	
IUniswapV2Router02	Interface			
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗️	🔴 |NO❗️	
└	factory	External ❗️	|NO❗️	
└	WETH	External ❗️	|NO❗️	
└	addLiquidityETH	External ❗️	💵 |NO❗️	
Context	Implementation			
└	_msgSender	Internal 🔒		
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗️	🔴 |NO❗️	
└	owner	Public ❗️	|NO❗️	
└	renounceOwnership	Public ❗️	🔴	onlyOwner	
└	transferOwnership	Public ❗️	🔴	onlyOwner	
BatmanInu	Implementation	Context, IERC20, Ownable		
└	<Constructor>	Public ❗️	🔴 |NO❗️	
└	name	Public ❗️	|NO❗️	
└	symbol	Public ❗️	|NO❗️	
└	decimals	Public ❗️	|NO❗️	

└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🔴	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🔴	NO❗
└	transferFrom	Public ❗	🔴	NO❗
└	tokenFromReflection	Private 🔓		
└	_approve	Private 🔒	🔴	
└	_transfer	Private 🔒	🔴	
└	swapTokensForEth	Private 🔒	🔴	lockTheSwap
└	sendETHToFee	Private 🔒	🔴	
└	_tokenTransfer	Private 🔒	🔴	
└	rescueForeignTokens	Public ❗	🔴	onlyDev
└	setNewDevAddress	Public ❗	🔴	onlyDev
└	setNewMarketingAddress	Public ❗	🔴	onlyDev
└	_transferStandard	Private 🔒	🔴	
└	_takeTeam	Private 🔒	🔴	
└	_reflectFee	Private 🔒	🔴	
└	<Receive Ether>	External ❗	💵	NO❗
└	_getValues	Private 🔒		
└	_getTValues	Private 🔒		
└	_getRValues	Private 🔒		
└	_getRate	Private 🔒		
└	_getCurrentSupply	Private 🔒		
└	manualswap	External ❗	🔴	NO❗
└	manualsend	External ❗	🔴	NO❗
└	setFee	Public ❗	🔴	onlyDev
└	setBot	Public ❗	🔴	onlyDev
└	removeBot	Public ❗	🔴	onlyDev
└	setMM	Public ❗	🔴	onlyDev
└	excludeMultipleAccountsFromFees	Public ❗	🔴	onlyOwner

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

Contracts Function Graph
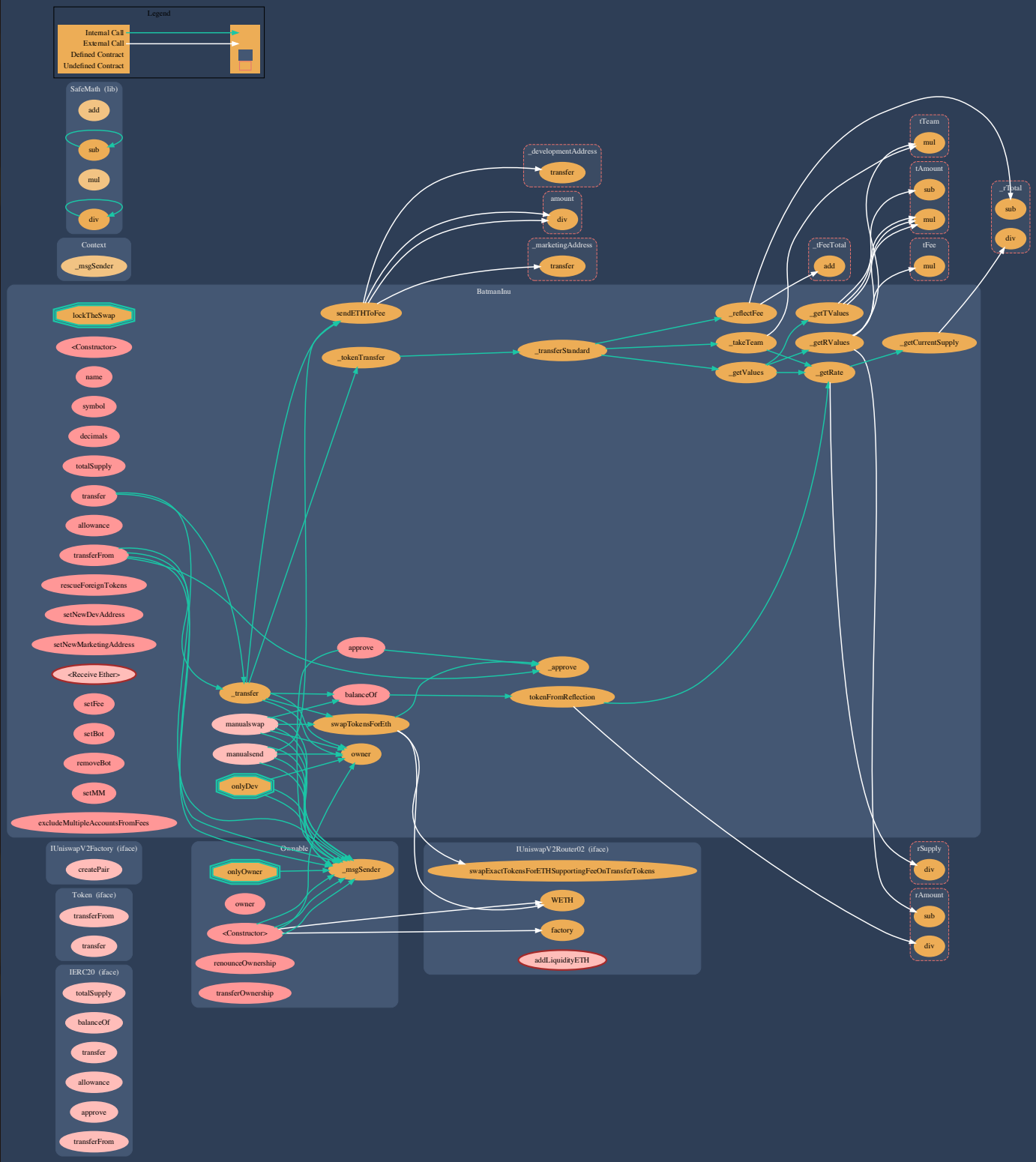
Vulnerability Checklist

№	Description.	Result
1	Compiler warnings.	Passed
2	Race conditions and Re-entrancy. Cross-function race	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

Contract overview

🍌 KYC verified by ApeSale

🍌 Owner can set fees buy to 11%, sell to 16%

🍌 Owner can set bot blacklist, blacklisted accounts are not permitted to transfer tokens.

🍌 The owner can add/remove accounts from the transfer blacklist at any time.

🍌 Users can only buy after the marketMakerAddress buy first

🍌 Owner cannot pause trading

🍌 Owner cannot change max transaction amount

🍌 No burn functions are present though the circulating supply can be decreased by sending tokens to the 0x..dead address.

🍌 At the time of writing this report, 32% of the total supply belongs to ApeSale presale, 17% to add liquidity, 49% burned.

🍌 The owner can set each fee to any value as long as the total fee percentage combined does not exceed 16% on sell.

🍌 The owner can exclude accounts from transfer fees.

🍌 The owner can withdraw any BNB or tokens from the contract at any time.

🍌 There is a Development fee and Marketing fee on all transfers via Pancakeswap where neither the sender nor the recipient is excluded from fees

BATMANINU

Audited by ApeSale.app

Date: 15 Jun 2022

✔ Advanced Manual Smart Contract Audit