Coinsult

Advanced Manual Smart Contract Audit

BlockSAFU

Project: BlockSafu
Website: https://blocksafu.com/

● **Low-risk**
5 low-risk code issues found

● **Medium-risk**
0 medium-risk code issues found

● **High-risk**
0 high-risk code issues found

Contract address
0x32bFd701655EDF95809EaA5e525F0024ea571267

Disclaimer

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

Tokenomics

Total Supply:	1,000,000,000
Total Holders:	1

Top 10 holders:

Rank	Address	Quantity (Token)	Percentage
1	0x2d0ab0dc18233e4fe32e101124e0a53cdadf83b6	1,000,000,000	100.0000%

The top 100 holders collectively own 100.00% (1,000,000,000.00 Tokens) of BlockSAFU

Note: This is a snapshot of when the audit was performed.

Source code

Coinsult was commissioned by BlockSafu to perform an audit based on the following smart contract:

https://bscscan.com/address/0x32bFd701655EDF95809EaA5e525F0024ea571267#code

Note: This project uses imports. While we do check the full contract for vulnerabilities at the time of the audit, we can not ensure the correctness of these imported modules.

Manual Code Review

Low-risk

5 low-risk code issues found.
Could be fixed, will not bring problems.

- Weak PRNG, do not use block.timestamp as a source of randomness as this can be manipulated by miners.

  Recommendation: Avoid relying on block.timestamp.

```
    function _beforeTransferToken(address sender, address recipient,
uint256 amount) internal view {
      if(isPair[sender]){
        if(isLastTimeBuyEnable && !isExcludeFromTimeBuyLimit[recipient]
&& lastTimeBuy[recipient] > 0){
          require(block.timestamp.sub(lastTimeBuy[recipient]) >=
minimumTimeBuy,"Blocksafu: Minimum Time Buy is Exceed");
        }
      } else {
        if(isLastTimeSellEnable){
          require(block.timestamp.sub(lastTimeSell[sender]) >=
minimumTimeSell,"Blocksafu: Minimum Time Sell is Exceed");
        }
      }
    }
```

- Contract contains Reentrancy vulnerabilities: _transfer(address,address,uint256)

  Additional information: This combination increases risk of malicious intent. While it may be justified by some complex mechanics (e.g. rebase, reflections, buyback). More information: Slither

```
External calls:
- _complexTransfer(sender,recipient,amount) (contracts/Blocksafu.sol#240)
- router.addLiquidityETH{value: amountETH}(address(this),amountToken,0,0,owner,block.timestamp)
(contracts/Blocksafu.sol#535-542)
-router.swapExactTokensForETHSupportingFeeOnTransferTokens(amountSwapForWeth,amountForSwap,path,addr
ess(this),block.timestamp) (contracts/Blocksafu.sol#496-502)
- IBlockstaking(routerStakingAddress).deposit{value: amountStaking}() (contracts/Blocksafu.sol#512)
External calls sending eth:
- _complexTransfer(sender,recipient,amount) (contracts/Blocksafu.sol#240)
- router.addLiquidityETH{value: amountETH}(address(this),amountToken,0,0,owner,block.timestamp)
(contracts/Blocksafu.sol#535-542)
- address(marketingAddress).transfer(amountMarketing) (contracts/Blocksafu.sol#510)
- IBlockstaking(routerStakingAddress).deposit{value: amountStaking}() (contracts/Blocksafu.sol#512)
- address(operationalAddress).transfer(amountDev) (contracts/Blocksafu.sol#514)
State variables written after the call(s):
- _afterTransferToken(sender,recipient,amount) (contracts/Blocksafu.sol#244)
- lastTimeBuy[recipient] = block.timestamp (contracts/Blocksafu.sol#277)
- _afterTransferToken(sender,recipient,amount) (contracts/Blocksafu.sol#244)
- lastTimeSell[sender] = block.timestamp (contracts/Blocksafu.sol#278)
```

- To many digits (Use: Ether suffix, Time suffix, or The scientific notation)

```
    uint256 public percentTaxDenominator = 10000;
    uint256 public minimumSwapForWeth = 1000000000;
    uint256 public minimumWethForTreasuryAndBurn = 1000000000000000000;
//1 BNB
    uint256 public minimumTokenForAddLiquidity = 1000000000;
```

- The return value of an external transfer/transferFrom call is not checked

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

```
    function claimFromContract(address _tokenAddress, address to,
uint256 amount) external onlyOwner {
      IERC20(_tokenAddress).transfer(to, amount);
    }
```

- Calls to a function sending Ether to an arbitrary address.

```
        //distribute
          uint256 amountMarketing =
getAmountPercent(balanceETHAfter,percentSellMarketing,totalTax);
          uint256 amountStaking =
getAmountPercent(balanceETHAfter,percentSellStaking,totalTax);
          uint256 amountDev =
getAmountPercent(balanceETHAfter,percentSellTreasury,totalTax);
        payable(marketingAddress).transfer(amountMarketing);
          if(isStakingEnable){

IBlockstaking(routerStakingAddress).deposit{value:amountStaking}();
          }
        payable(operationalAddress).transfer(amountDev);
```

🟡 Medium-risk

0 medium-risk code issues found.
Should be fixed, could bring problems.

🔴 High-risk

0 high-risk code issues found
Must be fixed, and will bring problems.

Extra notes by the team

Note: This project uses imports. While we do check the full contract for vulnerabilities at the time of the audit, we can not ensure the correctness of these imported modules.

● Owner can not set the sell fee higher than 24%

```
        require(_percentStaking <= 600,"Blocksafu: Maximum 6%");
        require(_percentMarketing <= 600,"Blocksafu: Maximum 6%");
        require(_percentOperational <= 600,"Blocksafu: Maximum 6%");
        require(_percentTreasury <= 600,"Blocksafu: Maximum 6%");
```

● Owner can not set the buy fee higher than 24%

```
        require(_percentStaking <= 600,"Blocksafu: Maximum 6%");
        require(_percentReferral <= 600,"Blocksafu: Maximum 6%");
        require(_percentOperational <= 600,"Blocksafu: Maximum 6%");
        require(_percentTreasury <= 600,"Blocksafu: Maximum 6%");
```

● Owner can change a lot of the contract

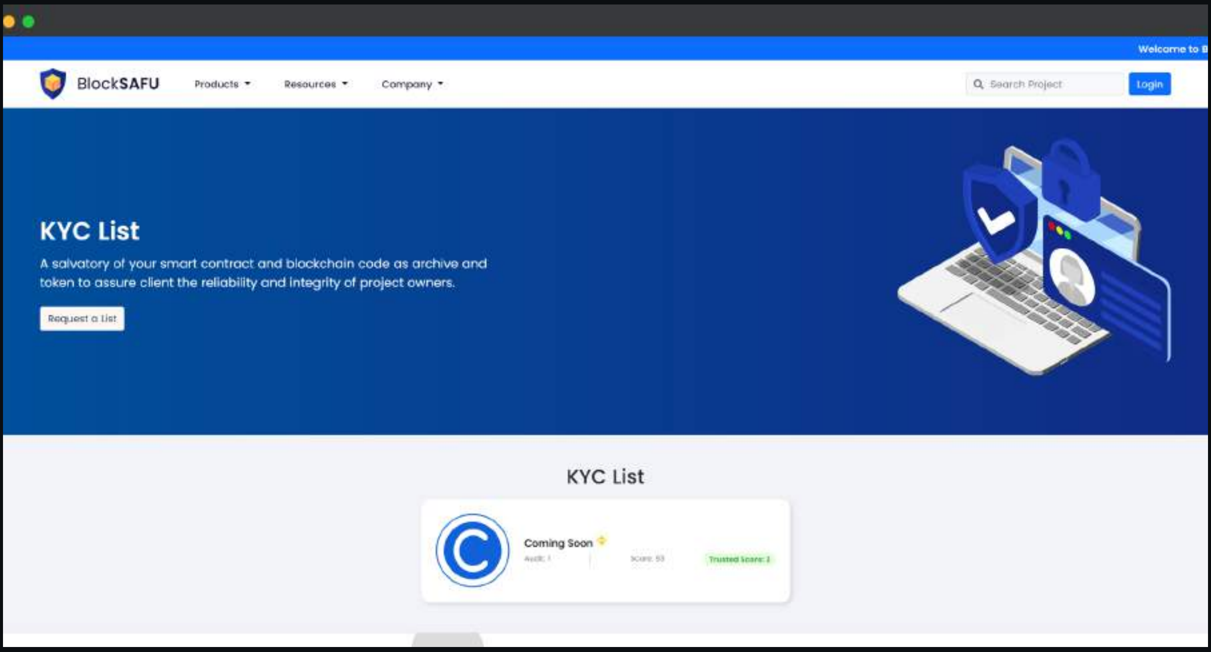
● Owner can change the router address

● Owner can exclude from fees

Contract Snapshot

```
contract Blocksafu is Context, Auth, IERC20 {
    using SafeMath for uint256;

    //ERC20
    uint8 private _decimals = 18;
    uint256 private _totalSupply = 1_000_000_000 * (10**_decimals);
    string private _name = "BlockSAFU";
    string private _symbol = "BSAFU";
    mapping(address => uint256) private _balances;
    mapping(address => mapping(address => uint256)) private
_allowances;

    //Tokenomic Buy
    uint256 public percentBuyStaking = 100;
    uint256 public percentBuyReferral = 100;
    uint256 public percentBuyOperational = 100;
    uint256 public percentBuyTreasury = 100;
    //Tokenomic Sell
    uint256 public percentSellStaking = 250;
    uint256 public percentSellMarketing = 250;
    uint256 public percentSellOperational = 200;
    uint256 public percentSellTreasury = 200;

    uint256 public percentTaxDenominator = 10000;
    uint256 public minimumSwapForWeth = 10000000000;
    uint256 public minimumWethForTreasuryAndBurn = 1000000000000000000;
//1 BNB
    uint256 public minimumTokenForAddLiquidity = 10000000000;
    uint256 public minimumTokenLeft = 1;
    uint256 public minimumTimeBuy = 1 minutes;
    uint256 public minimumTimeSell = 1 minutes;
    uint256 public maximumAmountPerWallet = _totalSupply;

    bool public isAutoSwapForWeth = true;
    bool public isTaxBuyEnable = true;
    bool public isTaxSellEnable = true;
    bool public isAutoTreasuryAndBurn = false;
    bool public isAutoAddLiquidity = false;
    bool public isHasMinimumTokenLeft = true;
    bool public isLastTimeBuyEnable = false;
```

Website Review



KYC List

A salvatory of your smart contract and blockchain code as archive and token to assure client the reliability and integrity of project owners.

KYC List

Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.

● Mobile Friendly
● Contains no jQuery errors
● SSL Secured
● No major spelling errors

Note: The website is graphically dependent on the visitor's graphics card; The website might lag if the visitor has an outdated graphics card.

Loading speed: 98%

Rug-pull Review

Based on the available information analyzed by us, we come to the following conclusions:

🟢 Locked Liquidity (no liquidity yet)

🟢 Large unlocked wallets

- Note: Tokens not distributed yet

🟢 Doxxed Team (KYC)

Honeypot Review

Based on the available information analyzed by us, we come to the following conclusions:

🟢 Ability to sell

🟢 Owner is not able to pause the contract

🟡 Router can be changed

Note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.