



BlockSAFU

ADVANCE MANUAL SMART CONTRACT AUDIT



Project: LibEarn

Website: earnitcrypto.ml



BlockSAFU Score: 84

Contract Address:

0x452f8Bd3D7Fc9f21DfB18A91f4562e705d054Fe9

**Disclaimer: BlockSAFU is not responsible for any financial losses.
Nothing in this contract audit is financial advice, please do your own reasearch.**

DISCLAIMER

BlockSAFU has completed this report to provide a summary of the Smart Contract functions, and any security, dependency or cybersecurity vulnerabilities. This is often a constrained report on our discoveries based on our investigation and understanding of the current programming versions as at the date of this report. In order to understand the full scope of our analysis, it is vital for you to at the date of this report. In order to understand the full scope of our analysis, it is vital for you to review the complete report. Although we have done our best in conducting our investigation and creating this report, it is vital to note that you should not depend on this report and cannot make any claim against BlockSAFU or its Subsidiaries and Team members on the premise of what has or has not been included in the report. Please remember to conduct your own independent examinations before making any investment choices. We do not provide investment advice or in any way claim to determine if the project will be successful or not.

By perusing this report or any portion of it, you concur to the terms of this disclaimer. In the unlikely situation where you do not concur to the terms, you should immediately terminate reading this report, and erase and discard any and all duplicates of this report downloaded and/or printed by you. This report is given for data purposes as it were and on a non- reliance premise, and does not constitute speculation counsel. No one should have any right to depend on the report or its substance, and BlockSAFU and its members (including holding companies, shareholders, backups, representatives, chiefs, officers and other agents) BlockSAFU and its subsidiaries owe no obligation of care towards you or any other person, nor does BlockSAFU make any guarantee or representation to any individual on the precision or completeness of the report.

ABOUT THE AUDITOR:

BlockSAFU (BSAFU) is an Anti-Scam Token Utility which reviews Smart Contracts and Token information to Identify Rug Pull and Honey Pot scamming activity. BlockSAFU's Development Team consists of a number of Smart Contract creators, Auditors Developers and Blockchain experts. BlockSAFU provides solutions, prevents and hunts down scammers. BSAFU is utility token with features are Audit, KYC, Token Generators and Bounty Scammers. It will enrich the crypto ecosystem.

OVERVIEW

BlockSAFU was commissioned by EARNIT Token to complete a Smart Contract audit. The objective of the Audit is to achieve the following:

- Review the Project and experience and Development team
- Ensure that the Smart Contract functions are necessary and operate as intended.
- Identify any vulnerabilities in the Smart Contract code.

DISCLAIMER: This Audit is intended to inform about token Contract Risks, the result does not imply an endorsement or provide financial advice in any way, all investments are made at your own risk. (<https://blocksafu.com/>)






















SMART CONTRACT REVIEW

Token Name	LibEarn
Token Symbol	LibEarn
Token Decimal	9
Total Supply	1,000,000,000 LibEarn
Contract Address	0x452f8Bd3D7Fc9f21DfB18A91f4562e705d054Fe9
Deployer Address	0x8870a37327d819b57bbe952df890833872d6dc7a
Owner Address	0x8870a37327d819b57bbe952df890833872d6dc7a
Tax Fees Buy	15%
Tax Fees Sell	15%
Gas Used for Buy	835,846
Gas Used for Sell	1,154,821
Contract Created	Apr-01-2022 04:10:46 AM +UTC
Initial Liquidity	1 BNB
Liquidity Status	Locked
Unlocked Date	July-05-2022 06:05:00 AM +UTC
Verified CA	Yes
Compiler	v0.8.13+commit.abaa5c0e
Optimization	Enable with 200 runs
Sol License	MIT License
Top 5 Holders	18.1% Held
Other	default evmVersion

TOKENOMICS

BUY	
Reflection Rewards	10%
Liquidity Fee	2%
Marketing Fee	2%
Buyback Fee	1%
SELL	
Reflection Rewards	10%
Liquidity Fee	2%
Marketing Fee	2%
Buyback Fee	1%

TOP HOLDER

Rank	Address	Quantity	Percentage	Analytics
1	 PinkSale: PinkLock	400,000,000	40.0000%	
2	 PancakeSwap V2: LibEarn 3	152,710,103.960398261	15.2710%	
3	0xabbd79501cf957435dfe2daad15fb37b9a50f04	60,000,000	6.0000%	
4	 0x452f8bd3d7fc9f21dfb18a91f4562e705d054fe9	59,282,534.004281153	5.9283%	
5	0xad74ab0fc9cb2fae4ee8e5a17106d1101b8311c6	40,000,000	4.0000%	
6	0x489a949c30f9281560cd046d2e4e80945c451729	31,803,160.968407636	3.1803%	
7	0x893581d2ba82682ebd70c9f6bea688e6856a215	25,673,733.835398154	2.5674%	
8	0x92d8e8839c8c4481ef3ec350e1847daa78294aeb	24,107,400.745764786	2.4107%	
9	0xfd07a73ff9bfbd7e7fd2517fb83151b12e39fbc	21,713,921.484988201	2.1714%	
10	0x55c53721202ef376af8089cd7ac0687ae256e657	21,128,144.678486126	2.1128%	
11	0x01c83b87854c3dc55c4bfc3c8f3f0eea2a9873d	17,595,000	1.7595%	
12	0xdfd19e0957f34db855e90dc773657ce0fc4434a9	15,315,799.956165848	1.5316%	
13	0x8a296f0a6cbe0be2dc91039f6a899d60ac7644a9	14,179,129.929445785	1.4179%	
14	0x3157ed5e82b495cceb4e7f5318b1c0e8eea25723	12,121,423.384526086	1.2121%	
15	0x6bcfd7ce7745146845dc1fbc7e49fbc74a69a10	11,125,859.264954792	1.1126%	
16	0xab7f5f36b5f247c4fd7dbd7a49bodd1ce647f6bf	10,934,485	1.0934%	
17	0x284dc65c7a70c19f092f79e9881f4bc5486f7a89	10,450,850.516425935	1.0451%	
18	0x9972b89348651ef8620b645cf4114fc6839e0311	9,775,000	0.9775%	

OFFICIAL WEBSITE AND SOCIAL MEDIA

Website: <https://earnitcrypto.ml/libearn>

Telegram Group: https://t.me/earn_it_crypto

Twitter: https://t.me/earn_it_crypto

Instagram: https://instagram.com/earn_it_crypto

MANUAL CODE REVIEW

● Minor-risk

3 minor-risk code issues found

Could be fixed, will not bring problems.

Weak PRNG (*Pseudo-random number generator*), do not use blocktimestamp as a source randomness as this can be manipulate by miners.

Recommendation: Avoid relying on block.timestamp

```
function getMultipliedFee() public view returns (uint256) {
    if (launchedAtTimestamp + 1 days > block.timestamp) {
        return totalFee.mul(18000).div(feeDenominator);
    } else if (buybackMultiplierTriggeredAt.add(buybackMultiplierLength) >
block.timestamp) {
        uint256 remainingTime =
buybackMultiplierTriggeredAt.add(buybackMultiplierLength).sub(block.timestamp);
        uint256 feeIncrease =
totalFee.mul(buybackMultiplierNumerator).div(buybackMultiplierDenominator).sub(total
Fee);
        return
totalFee.add(feeIncrease.mul(remainingTime).div(buybackMultiplierLength));
    }
    return totalFee;
}
```

The return value of an external transfer/transferFrom return value is checked.
Recommendation: use SafeERC20, or ensure that the transfer/transferFrom return value is checked

```
function transfer(address recipient, uint256 amount) external override returns (bool) {  
    return _transferFrom(msg.sender, recipient, amount);  
}  
  
function transferFrom(address sender, address recipient, uint256 amount) external  
override returns (bool) {  
    if(_allowances[sender][msg.sender] != _totalSupply){  
        _allowances[sender][msg.sender] = _allowances[sender][msg.sender].sub(amount,  
"Insufficient Allowance");  
    }  
  
    return _transferFrom(sender, recipient, amount);  
}
```

Calls to a function sending ether to an arbitrary address.

```
address public autoLiquidityReceiver;  
address public marketingFeeReceiver;
```

manual execution

● Medium-risk

1 medium-risk code issues found

Should be fixed, could bring problems.

```
uint256 public minPeriod = 1 hours;  
uint256 public minDistribution = 1 * (10 ** 18);  
  
function setDistributionCriteria(uint256 _minPeriod, uint256 _minDistribution) external  
override onlyToken {  
    minPeriod = _minPeriod;  
    minDistribution = _minDistribution;  
}
```

The Owner Can set reward distribution by min period and min distribution (no maximum limit)

```
function addShareholder(address shareholder) internal {  
    shareholderIndexes[shareholder] = shareholders.length;  
    shareholders.push(shareholder);  
}  
  
function removeShareholder(address shareholder) internal {  
    shareholders[shareholderIndexes[shareholder]] =  
shareholders[shareholders.length-1];  
    shareholderIndexes[shareholders[shareholders.length-1]] =  
shareholderIndexes[shareholder];  
    shareholders.pop();  
}
```

Contract Informational

● High-Risk

0 high-risk code issues found

Must be fixed, and will bring problem.

● Critical-Risk

0 critical-risk code issues found

Must be fixed, and will bring problem.

EXTRA NOTES SMART CONTRACT

● Owner cannot set the sell fee higher than 25%

```
uint256 public _maxTxAmount = _totalSupply.div(400); // 0.25%  
  
function checkTxLimit(address sender, uint256 amount) internal view {  
    require(amount <= _maxTxAmount || isTxLimitExempt[sender], "TX Limit  
Exceeded");  
}
```

The contract has function max transaction limit but it will be work if the transaction amount more than 0.25%

```
function setTxLimit(uint256 amount) external authorized {  
    require(amount >= _totalSupply / 1000);  
    _maxTxAmount = amount;  
}
```

The contract has a set transaction limit function with a maximum condition of not more than the total supply / 1000

```

function setIsDividendExempt(address holder, bool exempt) external authorized {
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;
    ...
}

function setIsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}

function setIsTxLimitExempt(address holder, bool exempt) external authorized {
    isTxLimitExempt[holder] = exempt;
}

```

The Contract Owner can set address to dividend exemption, fee exemption, and tx limit exemption

```

function setFees(uint256 _liquidityFee, uint256 _buybackFee, uint256 _reflectionFee,
uint256 _marketingFee, uint256 _feeDenominator) external authorized {
    liquidityFee = _liquidityFee;
    buybackFee = _buybackFee;
    reflectionFee = _reflectionFee;
    marketingFee = _marketingFee;
    totalFee = _liquidityFee.add(_buybackFee).add(_reflectionFee).add(_marketingFee);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator/4);
}

```

The contract owner can set fees max equals to $10000/4 = 2500 = 25\%$

● Safemath

```
library SafeMath {

    function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            uint256 c = a + b;
            if (c < a) return (false, 0);
            return (true, c);
        }
    }

    function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            if (b > a) return (false, 0);
            return (true, a - b);
        }
    }

    function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            // Gas optimization: this is cheaper than requiring 'a' not being zero, but the
            // benefit is lost if 'b' is also tested.
            // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522
            if (a == 0) return (true, 0);
            uint256 c = a * b;
            if (c / a != b) return (false, 0);
            return (true, c);
        }
    }
    ....
}
```

The contract has a standard safemath library for mathematics calculation function on solidity

● IBEP20 Interface

```
interface IBEP20 {  
    function totalSupply() external view returns (uint256);  
    function decimals() external view returns (uint8);  
    function symbol() external view returns (string memory);  
    function name() external view returns (string memory);  
    function getOwner() external view returns (address);  
    function balanceOf(address account) external view returns (uint256);  
    function transfer(address recipient, uint256 amount) external returns (bool);  
    function allowance(address _owner, address spender) external view returns (uint256);  
    function approve(address spender, uint256 amount) external returns (bool);  
    function transferFrom(address sender, address recipient, uint256 amount) external returns  
(bool);  
    event Transfer(address indexed from, address indexed to, uint256 value);  
    event Approval(address indexed owner, address indexed spender, uint256 value);  
}
```

The Contract have standard BEP20 token template (interface)

● IBEP20 Interface

```
abstract contract Auth {
    address internal owner;
    mapping (address => bool) internal authorizations;

    constructor(address _owner) {
        owner = _owner;
        authorizations[_owner] = true;
    }

    /**
     * Function modifier to require caller to be contract owner
     */
    modifier onlyOwner() {
        require(isOwner(msg.sender), "!OWNER"); _;
    }
    .....

    /**
     * Transfer ownership to new address. Caller must be owner. Leaves old owner authorized
     */
    function transferOwnership(address payable adr) public onlyOwner {
        owner = adr;
        authorizations[adr] = true;
        emit OwnershipTransferred(adr);
    }

    event OwnershipTransferred(address owner);
}
```

The Contract have standard auth contract,
The contract owner can be transfer ownership to other address

● IDEX Factory

```
interface IDEXFactory {
    function createPair(address tokenA, address tokenB) external returns (address pair);
}
```

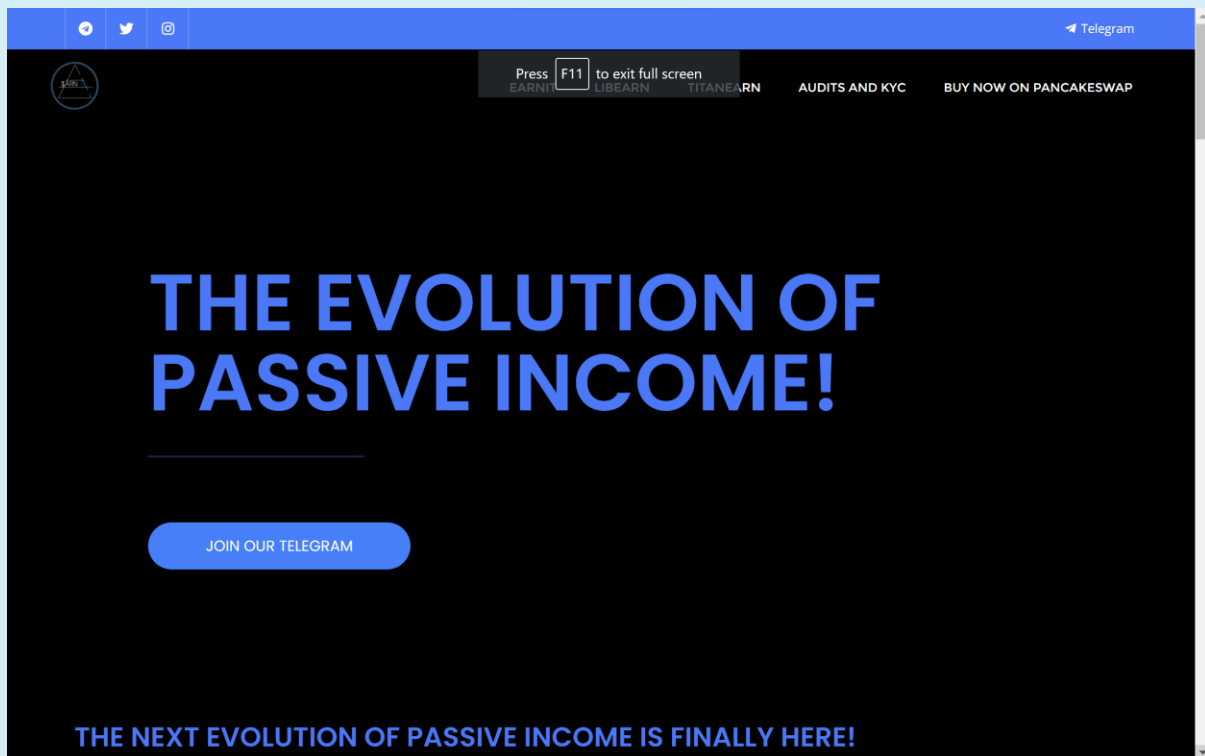
The Contract have factory function for create Pair

● IDEX Router

```
interface IDEXRouter {  
    function factory() external pure returns (address);  
    function WETH() external pure returns (address);  
  
    function addLiquidity(  
        address tokenA,  
        address tokenB,  
        ....  
    ) external returns (uint amountA, uint amountB, uint liquidity);  
  
    function addLiquidityETH(  
        address token,  
        uint amountTokenDesired,  
        ....  
    ) external payable returns (uint amountToken, uint amountETH, uint liquidity);  
  
    function swapExactTokensForTokensSupportingFeeOnTransferTokens(  
        uint amountIn,  
        ....  
    ) external;  
  
    function swapExactETHForTokensSupportingFeeOnTransferTokens(  
        uint amountOutMin,  
        ....  
    ) external payable;  
  
    function swapExactTokensForETHSupportingFeeOnTransferTokens(  
        uint amountIn,  
        ....  
    ) external;  
}
```

The Contract have router function for swap reward from dividend to token reward

WEBSITE REVIEW



- **Mobile Friendly**
- **Contains no code error**
- **SSL Secured (By Zero SSL) but not trusted in all web browser**

Web Tec stack: Nginx, WordPress

DOM Content Loaded	15.80 s
Load	31.15 s
Performance	67%
Accessibility	91%
Best Practices	83%
SEO	83%

RUG-PULL REVIEW

Base on the available information analysed by us, we come to the following conclusions:

- Locked Liquidity

(Unlocked Date: July-05-2022 06:05:00 AM +UTC)

- TOP 5 Holder: 18.1% Held

- Team KYC by BlockSAFU

HONEYPOT REVIEW

- Ability to sell

- Owner is not able to pause the contract

- The Owner Can set reward distribution by min period and min distribution (no maximum limit)

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.