BlockSAFU

ADVANCE MANUAL
SMART CONTRACT AUDIT

Project: Crypto Velox

Website: cryptovelox.com

BlockSAFU Score: 74%

Contract Address:

0x452f8Bd3D7Fc9f21DfB18A91f4562e705d054Fe9

DISCLAMER

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 of the date of this report. To understand the full scope of our analysis, it is vital for you to at the date of this report. To understand the full scope of our analysis, you need 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 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 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 that reviews Smart Contracts and Token information to Identify Rug Pull and Honey Pot scamming activity. BlockSAFUs Development Team consists of several Smart Contract creators, Auditors Developers, and Blockchain experts. BlockSAFU provides solutions, prevents, and hunts down scammers. BSAFU is a utility token with features Audit, KYC, Token Generators, and Bounty Scammers. It will enrich the crypto ecosystem.

OVERVIEW

BlockSAFU was commissioned by Crypto Velox 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/)

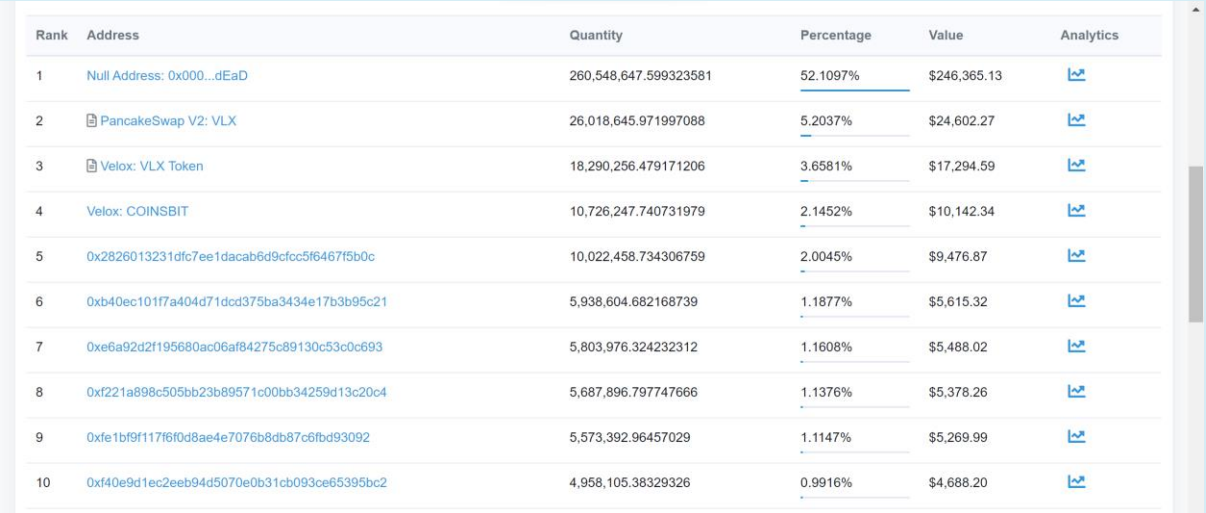
SMART CONTRACT REVIEW

Token Name	VELOX
Token Symbol	VLX
Token Decimal	9
Total Supply	1,000,000,000 LibEarn
Contract Address	0x11BE9D518F03a4d309a1340fddAb944CCC5B3b15
Deployer Address	0x5654c6ae69F2e3a4728e5F9C39a0410578bE4716
Owner Address	0x5654c6ae69F2e3a4728e5F9C39a0410578bE4716
Tax Fees Buy	3%
Tax Fees Sell	3%
Gas Used for Buy	233,700
Gas Used for Sell	170,670
Contract Created	Aug-04-2021 07:46:22 AM +UTC
Initial Liquidity	0.08 BNB
Liquidity Status	Burn
Unlocked Date	N/A
Verified CA	Yes
Compiler	v0.8.4+commit.c7e474f2
Optimization	Enable with 200 runs
Sol License	None License
Top 5 Holders	6.6% Held
Other	default evmVersion

TOKENOMICS

BUY	
Holders	1%
Liquidity Locked Forever	1%
Rewards, Marketing, and Maintenance	1%
SELL	
Holders	1%
Liquidity Locked Forever	1%
Rewards, Marketing, and Maintenance	1%

TOP 10 HOLDER



Rank	Address	Quantity	Percentage	Value	Analytics
1	Null Address: 0x000...dEaD	260,548,647.599323581	52.1097%	$246,365.13	
2	PancakeSwap V2: VLX	26,018,645.971997088	5.2037%	$24,602.27	
3	Velox: VLX Token	18,290,256.479171206	3.6581%	$17,294.59	
4	Velox: COINSBIT	10,726,247.740731979	2.1452%	$10,142.34	
5	0x2826013231dfc7ee1dacab6d9cfcc5f6467f5b0c	10,022,458.734306759	2.0045%	$9,476.87	
6	0xb40ec101f7a404d71dcd375ba3434e17b3b95c21	5,938,604.682168739	1.1877%	$5,615.32	
7	0xe6a92d2f195680ac06af84275c89130c53c0c693	5,803,976.324232312	1.1608%	$5,488.02	
8	0xf221a898c505bb23b89571c00bb34259d13c20c4	5,687,896.797747666	1.1376%	$5,378.26	
9	0xfe1bf9f117f6f0d8ae4e7076b8db87c6fbd93092	5,573,392.96457029	1.1147%	$5,269.99	
10	0xf40e9d1ec2eeb94d5070e0b31cb093ce65395bc2	4,958,105.38329326	0.9916%	$4,688.20	

OFFICIAL WEBSITE AND SOCIAL MEDIA

Website: https://cryptovelox.com/

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

Twitter: https://twitter.com/cryptovelox/

Medium: https://cryptovelox.medium.com/

Reddit: https://www.reddit.com/user/E-Velox?utm_source=share&utm_medium=ios_app&utm_name=iossmf

MANUAL CODE REVIEW

Minor-risk

4 minor-risk code issues found

Weak PRNG (***Pseudo-random number generator***), do not use blocktimestamp as source randomness as this can be manipulated by miners.
Recommendation: Avoid relying on block.timestamp

```
function unlock() public virtual {
      require(_previousOwner == msg.sender, "You don't have permission to unlock.");
      require(block.timestamp > _lockTime , "Contract is locked.");
      emit OwnershipTransferred(_owner, _previousOwner);
      _owner = _previousOwner;
    }
```

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

```
function transferFrom(address sender, address recipient, uint256 amount) public override
returns (bool) {
      _transfer(sender, recipient, amount);
      _approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount,
"ERC20: transfer amount exceeds allowance"));
      return true;
    }
```

Calls to a function sending ether to an arbitrary address.

```
function setDevWalletAddress(address _addr) public onlyOwner {
      _devWalletAddress = _addr;
    }
```

All fees transferred to one address, it will be mixed up and will make it difficult to trace the use of funds

```
function _getTValues(uint256 tAmount) private view returns (uint256, uint256, uint256,
uint256) {
        uint256 tFee = calculateTaxFee(tAmount);
        uint256 tLiquidity = calculateLiquidityFee(tAmount);
        uint256 tDev = calculateDevFee(tAmount);
        uint256 tTransferAmount = tAmount.sub(tFee).sub(tLiquidity).sub(tDev);
        return (tTransferAmount, tFee, tLiquidity, tDev);
    }
```

Medium-risk

0 medium-risk code issues found

Should be fixed, could bring problems.

High-Risk

1 high-risk code issue found

Must be fixed, and will bring problem.

The owner can set the selling fee up to 100%

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
  _taxFee = taxFee;
}

function setDevFeePercent(uint256 devFee) external onlyOwner() {
  _devFee = devFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
  _liquidityFee = liquidityFee;
}

function setMaxTxPercent(uint256 maxTxPercent) public onlyOwner {
  _maxTxAmount = maxTxPercent  * 10 ** _decimals;
}
```

The owner contract can be modified Tax Fee, Dev Fee, Liquidity, or MaxTxPercent with the maximum amount

Critical-Risk

0 critical-risk code issues found

Must be fixed, and will bring honeypot.

EXTRA NOTES SMART CONTRACT

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

Contract Ownable

```
abstract contract Ownable is Context {
    address public _owner;
    address private _previousOwner;
    uint256 public _lockTime;


    ....
    function renounceOwnership() public virtual onlyOwner {
        emit OwnershipTransferred(_owner, address(0));
        _owner = address(0);
```

```
    }


    function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        emit OwnershipTransferred(_owner, newOwner);
        _owner = newOwner;
    }

        //Locks the contract for owner for the amount of time provided
    function lock(uint256 time) public virtual onlyOwner {
        _previousOwner = _owner;
        _owner = address(0);
        _lockTime = time;
        emit OwnershipTransferred(_owner, address(0));
    }

    //Unlocks the contract for owner when _lockTime is exceeds
    function unlock() public virtual {
        require(_previousOwner == msg.sender, "You don't have permission to unlock.");
        require(block.timestamp > _lockTime , "Contract is locked.");
        emit OwnershipTransferred(_owner, _previousOwner);
        _owner = _previousOwner;
    }
}
```

The Contract has a lock and unlocks function for disabling and enabling transfer ownership, The contract owner can be transfer ownership to another address

IUniswapV2Factory

```
interface IUniswapV2Factory {
    event PairCreated(address indexed token0, address indexed token1, address pair, uint);
    function feeTo() external view returns (address);
    function feeToSetter() external view returns (address);
    function getPair(address tokenA, address tokenB) external view returns (address pair);
    function allPairs(uint) external view returns (address pair);
    function allPairsLength() external view returns (uint);
    function createPair(address tokenA, address tokenB) external returns (address pair);
    function setFeeTo(address) external;
    function setFeeToSetter(address) external;
}
```

The Contract has a factory function for PairCreated

Contract Coin Token

```
function isExcludedFromReward(address account) public view returns (bool) {
    return _isExcluded[account];
  }

  function totalFees() public view returns (uint256) {
    return _tFeeTotal;
  }
```

The contract owner can set the address to exclude from the reward

```
_tTotal = _supply * 10 ** _decimals;
...
function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view
returns(uint256) {
    require(tAmount <= _tTotal, "Amount must be less than supply");
    if (!deductTransferFee) {
      (uint256 rAmount,,,,,,) = _getValues(tAmount);
      return rAmount;
    } else {
      (,uint256 rTransferAmount,,,,,) = _getValues(tAmount);
      return rTransferAmount;
    }
  }
```

Function reflection from token amount cannot be bigger than the Total Supply

```
function excludeFromReward(address account) public onlyOwner() {
        require(!_isExcluded[account], "Account is already excluded");
        if(_rOwned[account] > 0) {
            _tOwned[account] = tokenFromReflection(_rOwned[account]);
        }
        _isExcluded[account] = true;
        _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
        require(_isExcluded[account], "Account is already included");
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_excluded[i] == account) {
                _excluded[i] = _excluded[_excluded.length - 1];
                _tOwned[account] = 0;
                _isExcluded[account] = false;
                _excluded.pop();
                break;
            }
        }
 }
```

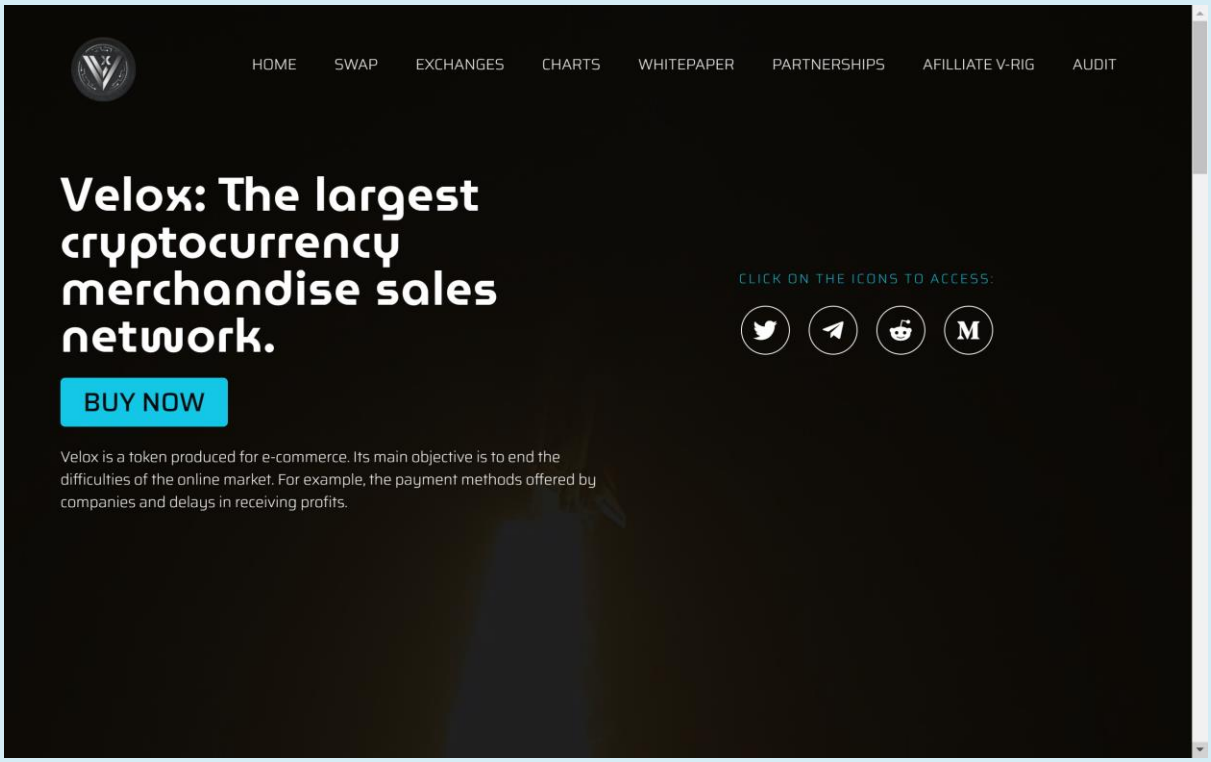
The contract owner can include and exclude the address from the reward

```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

The contract owner can include and exclude the address from the fee

WEBSITE REVIEW



- **Mobile Friendly**
- **Contains no code error**
- **SSL Secured (By Let's Encrypt)**

Web-Tec stack: Elementor, apache, PHP, jQuery, jQuery UI (need update to latest version), Google Fonts, FontAwesome

Largest Contentful Paint	1.7 s
Full Load	5.0 s
Performance	71%
Accessibility	82%
Best Practices	92%
SEO	75%

RUG-PULL REVIEW

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

- Burn Liquidity
- Large unlocked wallets

  TOP 5 Held: 6.6%
- Team KYC by BlockSAFU

HONEYPOT REVIEW

- Ability to sell
- The owner is not able to pause the contract
- The owner can set the selling fee up to 100%

Note: Please check the disclaimer above and note, that the audit makes no statements or warranties on the 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.