BlockSAFU

ADVANCE MANUAL
SMART CONTRACT AUDIT

Project: FUTURE BSC CARD

Website: future-bsc.com

PASSED

BlockSAFU Score: 82

Contract Address:

0x5b74485cb4E09EC49DE4f20e6849241B574fB95E

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 this report's date. 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 FUTUREBSCCARD 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	FUTURE BSC CARD
Token Symbol	FBC
Token Decimal	18
Total Supply	100,000,000 FBC
Contract Address	0x5b74485cb4E09EC49DE4f20e6849241B574fB95E
Deployer Address	0x98d91d49C0a7402Ab03bd92F8899C26DF641C75f
Owner Address	0x94f80ad66c7fffa1b056d47Fe0aDfC0af20b7B71
Tax Fees Buy	8%
Tax Fees Sell	10%
Gas Used for Buy	*will be updated after the DEX listing*
Gas Used for Sell	*will be updated after the DEX listing*
Contract Created	Jul-20-2022 01:54:32 AM +UTC
Initial Liquidity	*will be updated after the DEX listing*
Liquidity Status	Locked
Unlocked Date	*will be updated after the DEX listing*
Verified CA	Yes
Compiler	v0.8.7+commit.e28d00a7
Optimization	No with 200 runs
Sol License	MIT License
Top 5 Holders	*will be updated after the DEX listing*
Other	default evmVersion

TAX

BUY	8%	**SELL**	10%
Burn Fee	1%	Burn Fee	1%
Liquidity Fee	2%	Liquidity Fee	4%
NFT Prize	5%	NFT Prize	5%

TOP HOLDER

Rank	Address	Quantity	Percentage	Analytics
1	0x98d91d49c0a7402ab03bd92f8699c26df641c75f	100,000,000	100.0000%	

[Download CSV Export]

Team Review

The FUTURE BSC CARD team has a nice website, their website is professionally built and the Smart contract is well developed, their social media is growing with over 2 people in their telegram group (count in audit date).

OFFICIAL WEBSITE AND SOCIAL MEDIA

Website: https://www.future-bsc.com/

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

Twitter: https://twitter.com/FutureCard

MANUAL CODE REVIEW

🟢 Minor-risk

1 minor-risk code issue found

Could be fixed, and will not bring problems.

```
function handleTax(address from, address to, uint256 amount) private returns (uint256)
{
      ...
payable(taxWallets["nftPrize"]).transfer(nftPrizeETH);
payable(taxWallets["liquidity"]).transfer(liquidityETH);
...
   }
```

Add liquidity, lock or burn CAKE-LP manually by owner.

🟡 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 problem.

🔴 Critical-Risk

0 critical-risk code issues found

Must be fixed, and will bring problem.

EXTRA NOTES SMART CONTRACT

1. IERC20

```
interface IERC20 {
    /**
     * @dev Returns the number of tokens in existence.
     */
    function totalSupply() external view returns (uint256);
...
    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);

    /**
     * @dev Emitted when `value` tokens are moved from one account (`from`) to
     * another (`to`).
     *
     * Note that `value` may be zero.
     */
    event Transfer(address indexed from, address indexed to, uint256 value);
...
}
```

IERC20 Normal Base Template

2. SafeMath Contract

```
library SafeMath {
...
   function add(uint256 a, uint256 b) internal pure returns (uint256) {
      uint256 c = a + b;
      require(c >= a, "SafeMath: addition overflow");
      return c;
   }
...
   function sub(uint256 a, uint256 b, string memory errorMessage) internal
pure returns (uint256) {
      require(b <= a, errorMessage);
      uint256 c = a - b;

      return c;
   }
   /**
    * @dev Returns the multiplication of two unsigned integers, reverting on
    * overflow.
    *
    * Counterpart to Solidity's `*` operator.
    *
    * Requirements:
    *
    * - Multiplication cannot overflow.
    */
 ...
   function mod(
      uint256 a,
      uint256 b,
      string memory errorMessage
   ) internal pure returns (uint256) {
      unchecked {
         require(b > 0, errorMessage);
         return a % b;
      }
   }
}
```

Standard Safemath contract

3. FUTUREBSCCARD Contract

```
contract FBCToken is ERC20, Ownable {
    using SafeMath for uint256;
    uint256 private constant initialSupply = 100_000_000_000 * 10**18;
    uint256 public constant denominator = 10000;
    uint256 public constant swapThreshold = 0.0000005 ether; // The contract will only swap to ETH, once the fee tokens reach the specified threshold

    mapping (address => bool) public blacklist;
    mapping (address => bool) public excludeList;

    mapping (string => uint256) public buyTaxes;
    mapping (string => uint256) public sellTaxes;
    mapping (string => address) public taxWallets;
    ...
        // Buy Taxes
        buyTaxes["nftPrize"] = 500;
        buyTaxes["burn"] = 100;
        buyTaxes["liquidity"] = 200;

        // Sell Taxes
        sellTaxes["nftPrize"] = 500;
        sellTaxes["burn"] = 100;
        sellTaxes["liquidity"] = 400;

        // External wallets
        taxWallets["nftPrize"] = msg.sender;
        taxWallets["liquidity"] = msg.sender;

        _mint(owner(), initialSupply);
    }

    uint256 private nftPrizeTokens;
    uint256 private liquidityTokens;
    uint256 private burnTokens;
    ...

    /**
     * @dev Sets tax for buys.
     */
```

```
    function setBuyTax(uint256 _nft, uint256 _burnAmount, uint256 _liquidity) external
onlyOwner {
        require(_nft <= 500,"FBC: Maximum 5%");
        require(_burnAmount <= 500,"FBC: Maximum 5%");
        require(_liquidity <= 500,"FBC: Maximum 5%");

        buyTaxes["nftPrize"] = _nft;
        buyTaxes["burn"] = _burnAmount;
        buyTaxes["liquidity"] = _liquidity;
    }

    /**
     * @dev Sets tax for sells.
     */
    function setSellTax(uint256 _nft, uint256 _burnAmount, uint256 _liquidity) external
onlyOwner {
        require(_nft <= 500,"FBC: Maximum 5%");
        require(_burnAmount <= 500,"FBC: Maximum 5%");
        require(_liquidity <= 500,"FBC: Maximum 5%");

        sellTaxes["nftPrize"] = _nft;
        sellTaxes["burn"] = _burnAmount;
        sellTaxes["liquidity"] = _liquidity;
    }


    /**
     * @dev Sets wallets for taxes.
     */
    function setTaxWallets(address _nft, address _liquidity) external onlyOwner {
        taxWallets["nftPrize"] = _nft;
        taxWallets["liquidity"] = _liquidity;
    }

    /**
     * @dev Enables tax globally.
     */
    function enableTax() external onlyOwner {
        require(!taxStatus, "FBC: Tax is already enabled");
        taxStatus = true;
```

```
}

/**
 * @dev Disables tax globally.
 */
function disableTax() external onlyOwner {
    require(taxStatus, "FBC: Tax is already disabled");
    taxStatus = false;
}

/**
 * @dev Returns true if the account is blacklisted, and false otherwise.
 */
function isBlacklisted(address account) public view returns (bool) {
    return blacklist[account];
}

/**
 * @dev Returns true if the account is excluded, and false otherwise.
 */
function isExcluded(address account) public view returns (bool) {
    return excludeList[account];
}

function isContract(address account) internal view returns (bool) {
    return account.code.length > 0;
}

function percentBuyNft() external view returns(uint256){
    return buyTaxes["nftPrize"];
}

function percentBuyBurn() external view returns(uint256){
    return buyTaxes["burn"];
}

function percentBuyLiquidity() external view returns(uint256){
    return buyTaxes["liquidity"];
}
```

```
    function percentSellNft() external view returns(uint256){
        return sellTaxes["nftPrize"];
    }

    function percentSellBurn() external view returns(uint256){
        return sellTaxes["burn"];
    }

    function percentSellLiquidity() external view returns(uint256){
        return sellTaxes["liquidity"];
    }

    receive() external payable {}
}
```

FUTUREBSCCARD Contract

```
function setBuyTax(uint256 _nft, uint256 _burnAmount, uint256 _liquidity) external
onlyOwner {
        require(_nft <= 500,"FBC: Maximum 5%");
        require(_burnAmount <= 500,"FBC: Maximum 5%");
        require(_liquidity <= 500,"FBC: Maximum 5%");
        buyTaxes["nftPrize"] = _nft;
        buyTaxes["burn"] = _burnAmount;
        buyTaxes["liquidity"] = _liquidity;
    }
```

The owner cannot set Buy tax over 5% for NFT, burn, liquidity fee, and can’t set fee over 15% for total

```
function setSellTax(uint256 _nft, uint256 _burnAmount, uint256 _liquidity) external
onlyOwner {
        require(_nft <= 500,"FBC: Maximum 5%");
        require(_burnAmount <= 500,"FBC: Maximum 5%");
        require(_liquidity <= 500,"FBC: Maximum 5%");

        buyTaxes["nftPrize"] = _nft;
        buyTaxes["burn"] = _burnAmount;
        buyTaxes["liquidity"] = _liquidity;
    }
```

The owner cannot set Sell tax over 5% for NFT, burn, liquidity fee, and can’t set fee over 15% for total

BlockSAFU Token Scanner

Token Information

Indicator	Value
Token Name	FUTURE BSC CARD
Token Symbol	FBC
Total Supply	100,000,000
Already Listed On Dex	Already Listed
Dex Listed	PancakeV2
Open Source	Open Source
Price	$0.00000000
Volume 24H	$0.00
Liquidity	$0 (0.00 BNB)
Tx Count 24H	0
Marketcap	$0

Security Information

Indicator	Value
Honeypot	Liquidity Not Available
Buy Fees	8%
Sell Fees	10%
Buy Gas	185,179**Gwei (0.000926 BNB / $0.26)**
Sell Gas	214,933 **Gwei (0.001075BNB / $0.30)**
Holder Count	1 **Holders**

Honeypot Safety

Indicator	Value
Can Take Back Ownership	Not detected
Owner Change Balance	Not detected
Blacklist	Detected
Modify Fees	Detected
Proxy	Not detected
Whitelisted	Not detected
Anti Whale	Not detected
Trading Cooldown	Not detected
Transfer Pausable	Not detected
Cannot Sell All	Not detected

Rug Pull Safety

Indicator	Value
Hidden Owner	Not detected
Creator Address	0x98d91d49...75f
Creator Balance	100,000,000 FBC
Creator Percent	100%
Owner Address	0x94f80ad6...b71
Owner Balance	0 FBC
Owner Percent	0%
Lp Holder Count	0
Lp Total Supply	NaN
Mint	Not detected

WEBSITE REVIEW

- **Mobile Friendly**
- **Contains no code error**
- **SSL Secured (By Sectigo SSL)**

Web-Tech stack: React, Bootstrap, Sectigo

Domain .com - Tracked by whois

First Contentful Paint:	245ms
Fully Loaded Time	5.2s
Performance	90%
Accessibility	96%
Best Practices	92%
SEO	92%

RUG-PULL REVIEW

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

- Locked Liquidity (Locked by pinksale)

  (*Will be updated after DEX listing*)

- Add liquidity, lock or burn CAKE-LP manually by owner.
- TOP 5 Holder.

  (*Will be updated after DEX listing*)

- **The team is No KYC.**

  (*Will be updated after KYC*)

HONEYPOT REVIEW

- Ability to sell.
- The owner is not able to pause the contract.
- The owner cannot set a fee over 5% for NFT, burn, and liquidity fee, and cannot set a fee over 15% for total.

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.