

## Security Assessment

# BFBSPORT

Verified On JUNE 6<sup>th</sup>, 2024

 @Vital-Block

 @VB\_Audit

 info@vitalblock.org

 www.vitalblock.org



PREPARED FOR:  
BFBSPORT






# TABLE OF CONTENTS

TABLE OF CONTENTS	3
DOCUMENT PROPERTIES	4
ABOUT VBS	5
SCOPE OF WORK	6
AUDIT METHODOLOGY	7
AUDIT CHECKLIST	9
EXECUTIVE SUMMARY	10
CENTRALIZED PRIVILEGES	11
RISK CATEGORIES	12
AUDIT SCOPE	13
AUTOMATED ANALYSIS	14
KEY FINDINGS	19
MANUAL REVIEW	20
VULNERABILITY SCAN	28
REPOSITORY	29
INHERITANCE GRAPH	30
PROJECT BASIC KNOWLEDGE	31
AUDIT RESULT	32
REFERENCES	37



## INTRODUCTION

<b>Auditing Firm</b>	 <b>VITAL BLOCK SECURITY</b>
<b>Client Firm</b>	 <b>BFBSPORT</b>
<b>Methodology</b>	Automated Analysis, Manual Code Review.
<b>Language</b>	Solidity
<b>Contract</b>	<a href="https://etherscan.io/address/0xCCE69362dD0Ea82Ea9C4bf03bfb1619E416adc55">0xCCE69362dD0Ea82Ea9C4bf03bfb1619E416adc55</a>
<b>Source Code Light</b>	Verified
<b>License</b>	MIT
<b>Centralization</b>	Active ownership
<b>Compiler Version</b>	v0.8.20+commit.a1b79de6
<b>Blockchain</b>	 <b>BINANCE SMART CHAIN</b>
<b>Website</b>	<a href="https://bfb sport.com/">https://bfb sport.com/</a>
<b>Telegram Group</b>	<a href="https://bfb sport.com/">https://bfb sport.com/</a>
<b>Twitter</b>	<a href="https://x.com/bfb sport_offic">https://x.com/bfb sport_offic</a>
<b>Git-Hub</b>	<a href="https://github.com/BFB-Sport/BFB-Sport-Contract">https://github.com/BFB-Sport/BFB-Sport-Contract</a>
<b>Prelim Report Date</b>	JUNE 5 <sup>th</sup> 2024
<b>Final Report Date</b>	JUNE 6 <sup>th</sup> 2024

 Verify the authenticity of this report on our GitHub Repo: <https://www.github.com/vital-block>



## Document Properties


<b>Client</b>	<b>BFBSPORT</b>
<b>Title</b>	Smart Contract Audit Report
<b>Target</b>	<b>BFBSPORT</b>
<b>Audit Version</b>	1.0
<b>Author</b>	Akhmetshin Marat
<b>Auditors</b>	Akhmetshin Marat, James BK, Benny Matin
<b>Reviewed by</b>	Dima Meru
<b>Approved by</b>	Prince Mitchell
<b>Classification</b>	Public

## Version Info

Version	Date	Author(s)	Description
1.0	JUNE 6 <sup>TH</sup> , 2024	James BK	Final Released
1.0-AP	JUNE 6 <sup>TH</sup> , 2024	Benny Matin	Release Candidate

## Contact

For more information about this document and its contents, please contact Vital Block Security Inc.

<b>Name</b>	Akhmetshin Marat
<b>Phone</b> 	+44 7944 248057
<b>Email</b>	info@vitalblock.org



In the following, we show the specific pull request and the commit hash value used in this audit.

- <https://bscscan.com/token/0xcce69362dd0ea82ea9c4bf03bfb1619e416adc55#code> (BFB22310)
- <https://github.com/BFB-Sport/BFB-Sport-Contract/tree/master/contracts> (BFB21511)

## About Vital Block Security

Vital Block Security provides professional, thorough, fast, and easy-to-understand smart contract security audit. We do in-depth and penetrative static, manual, automated, and intelligent analysis of the smart contract. Some of our automated scans include tools like ConsenSys MythX, Mythril, Slither, Surya. We can audit custom smart contracts, DApps, Rust, NFTs, etc (including the service of smart contract auditing). We are reachable at Telegram ([https://t.me/vital\\_block](https://t.me/vital_block)), Twitter ([http://twitter.com/Vb\\_Audit](http://twitter.com/Vb_Audit)), or Email ([info@vitalblock.org](mailto:info@vitalblock.org)).

Table 1.2: Vulnerability Severity Classification

Impact	High	Medium	Low
	Critical	High	Medium
	High	Medium	Low
	Medium	Low	Low
Likelihood			
High      Medium      Low			

## Methodology (1)

To standardize the evaluation, we define the following terminology based on the OWASP Risk Rating Methodology [4]:

- Likelihood represents how likely a particular vulnerability is to be uncovered and exploited in the wild;
- Impact measures the technical loss and business damage of a successful attack;
- Severity demonstrates the overall criticality of the risk.

## SCOPE OF WORK

Vital Block was consulted by **BFBSPORT** to conduct the smart contract audit of its Sol source code. The audit scope of work is strictly limited to mentioned .SOL file only:

O.BFBSPORT.Sol

 External contracts and/or interfaces dependencies are not checked due to being out of scope.

Verify audited contract code Repo.

### Public Contract Link

<https://bscscan.com/token/0xcce69362dd0ea82ea9c4bf03bfb1619e416adc55>



## AUDIT METHODOLOGY

Smart contract audits are conducted using a set of standards and procedures. Mutual collaboration is essential to performing an effective smart contract audit. Here's a brief overview of Vital Block Security auditing process and methodology:

### CONNECT

- The onboarding team gathers source codes, and specifications to make sure we understand the size, and scope of the smart contract audit.

### AUDIT

- Automated analysis is performed to identify common contract vulnerabilities. We may use the following third-party frameworks and dependencies to perform the automated analysis:
  - Remix IDE Developer Tool
  - Open Zeppelin Code Analyzer
  - SWC Vulnerabilities Registry
  - DEX Dependencies, e.g., Pancakeswap, Uniswap
- Simulations are performed to identify centralized exploits causing contract and/or trade locks.
- A manual line-by-line analysis is performed to identify contract issues and centralized privileges.

We may inspect below mentioned common contract vulnerabilities, and centralized exploits:

Centralized Exploits	<ul style="list-style-type: none"><li>○ Token Supply Manipulation</li><li>○ Access Control and Authorization</li><li>○ Assets Manipulation</li><li>○ Ownership Control</li><li>○ Liquidity Access</li><li>○ Stop and Pause Trading</li><li>○ Ownable Library Verification</li></ul>
----------------------	---

### **Common Contract Vulnerabilities**


- **Integer Overflow**
- **Lack of Arbitrary limits**
- **Incorrect Inheritance Order**
- **Typographical Errors**
- **Requirement Violation**
- **Gas Optimization**
- **Coding Style Violations**
- **Re-entrancy**
- **Third-Party Dependencies**
- **Potential Sandwich Attacks**
- **Irrelevant Codes**
- **Divide before multiply**
- **Conformance to Solidity Naming Guides**
- **Compiler Specific Warnings**
- **Language Specific Warnings**

### **REPORT**

- **The auditing team provides a preliminary report specifying all the checks which have been performed and the findings thereof.**
- **The client's development team reviews the report and makes amendments to the codes.**
- **The auditing team provides the final comprehensive report with open and unresolved issues.**

### **PUBLISH**

- **The client may use the audit report internally or disclose it publicly.**

 **It is important to note that there is no pass or fail in the audit, it is recommended to view the audit as an unbiased assessment of the safety of solidity codes.**










**Table 1.0 The Full Audit Checklist**

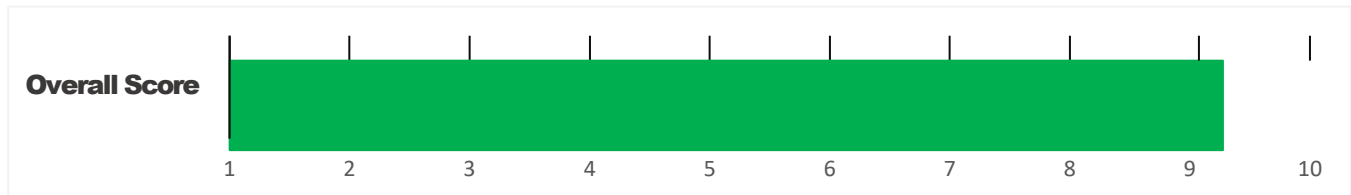
Category	Checklist Items
<b>Basic Coding Bugs</b>	Constructor Mismatch
	Ownership Takeover
	Redundant Fallback Function
	Overflows & Underflows
	Reentrancy
	Money-Giving Bug
	Blackhole
	Unauthorized Self-Destruct
	Revert DoS
	Unchecked External Call
	Gasless Send
	Send Instead Of Transfer
	Costly Loop
	(Unsafe) Use Of Untrusted Libraries
	(Unsafe) Use Of Predictable Variables
	Transaction Ordering Dependence
	Deprecated Uses
<b>Semantic Consistency Checks</b>	Semantic Consistency Checks
<b>Advanced DeFi Scrutiny</b>	Business Logics Review
	Functionality Checks
	Authentication Management
	Access Control & Authorization
	Oracle Security
	Digital Asset Escrow
	Kill-Switch Mechanism
	Operation Trails & Event Generation
	ERC20 Idiosyncrasies Handling
	Frontend-Contract Integration
	Deployment Consistency
	Holistic Risk Management
<b>Additional Recommendations</b>	Avoiding Use of Variadic Byte Array
	Using Fixed Compiler Version
	Making Visibility Level Explicit
	Making Type Inference Explicit
	Adhering To Function Declaration Strictly
	Following Other Best Practices



## EXECUTIVE SUMMARY

Vital Block Security has performed the automated and manual analysis of the **BFBSPORT** Sol code. The code was reviewed for common contract vulnerabilities and centralized exploits. Here's a quick audit summary:

Status	Critical ! 	Major " 	Medium # 	Minor \$ 	Unknown % 
Open	0	0	1	2	1
Acknowledged	0	0	0	2	1
Resolved	0	0	0	0	0
<b>Noteworthy</b> <b>OnlyOwner</b> <b>Privileges</b>					
Set Taxes and Ratios, Airdrop, Set Protection Settings, Set Reward Properties, Set Reflector Settings, Set Swap Settings, Set Pair and Router					

**BFBSPORT** Smart contract has achieved the following score: **92.0**



-  Please note that smart contracts deployed on blockchains aren't resistant to exploits, vulnerabilities and/or hacks. Blockchain and cryptography assets utilize new and emerging technologies. These technologies present a high level of ongoing risks. For a detailed understanding of risk severity, source code vulnerability, and audit limitations, kindly review the audit report thoroughly.
-  Please note that centralization privileges regardless of their inherited risk status - constitute an elevated impact on smart contract safety and security.



## CENTRALIZED PRIVILEGES

**Centralization risk is the most common cause of cryptography asset loss. When a smart contract has a privileged role, the risk related to centralization is elevated.**

**There are some well-intended reasons have privileged roles, such as:**

- **Privileged roles can be granted the power to `pause()` the contract in case of an external attack.**
- **Privileged roles can use functions like, `include()`, and `exclude()` to add or remove wallets from fees, swap checks, and transaction limits. This is useful to run a presale and to list on an exchange.**

**Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Lately, centralization-related losses are increasing in frequency and magnitude.**

- **The client can lower centralization-related risks by implementing below mentioned practices:**
- **Privileged role's private key must be carefully secured to avoid any potential hack.**
- **Privileged role should be shared by multi-signature (multi-sig) wallets.**
- **Authorized privilege can be locked in a contract, user voting, or community DAO can be introduced to unlock the privilege.**
- **Renouncing the contract ownership, and privileged roles.**
- **Remove functions with elevated centralization risk.**






** Understand the project's initial asset distribution. Assets in the liquidity pair should be locked.**

**Assets outside the liquidity pair should be locked with a release schedule.**



## RISK CATEGORIES

Smart contracts are generally designed to hold, approve, and transfer tokens. This makes them very tempting attack targets. A successful external attack may allow the external attacker to directly exploit. A successful centralization-related exploit may allow the privileged role to directly exploit. All risks which are identified in the audit report are categorized here for the reader to review:

Risk Type	Definition
<b>Critical</b> ! 	These risks could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
<b>Major</b> " 	These risks are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to high-risk severity.
<b>Medium</b> # 	These risks should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution. Low-risk re-entrancy-related vulnerabilities should be fixed to deter exploits.
<b>Minor</b> \$ 	These risks do not pose a considerable risk to the contract or those who interact with it. They are code-style violations and deviations from standard practices. They should be highlighted and fixed nonetheless.
<b>Unknown</b> % 	These risks pose uncertain severity to the contract or those who interact with it. They should be fixed immediately to mitigate the risk uncertainty.

All statuses which are identified in the audit report are categorized here for the reader to review:

Status Type	Definition
<b>Open</b>	Risks are open.
<b>Acknowledged</b>	Risks are acknowledged, but not fixed.
<b>Resolved</b>	Risks are acknowledged and fixed.



## Key Findings






Overall, these contracts are well-designed and engineered, though the implementation can be improved by resolving the identified issues (shown in Table [2.1](#)), 1 medium-severity vulnerabilities, 2 low-severity vulnerabilities, and 1 informational recommendations.

Table 2.1: Key **BFBSPORT** Audit Findings

ID	Severity	Title	Category	Status
MAR-001	Informational	<a href="#">In updateForOwner, Relevant Function Snippet</a>	Coding Practice	Fixed
MAR-002	Low	<a href="#">In updateFormapping , the following equation is used inside an unchecked block</a>	Business Logic	Fixed

Beside the identified issues, we emphasize that for any user-facing applications and services, it is always important to develop necessary risk-control mechanisms and make contingency plans, which may need to be exercised before the mainnet deployment. The risk-control mechanisms should kick in at the very moment when the contracts are being deployed on mainnet. Please refer to page [10](#) for details.

## AUTOMATED ANALYSIS

Symbol	Definition
	Function modifies state
	Function is payable
	Function is internal
	Function is private
	Function is important

```

**BFBSPORT** | Interface | |||
| L | totalSupply | External ! | ! | NO ! |
| L | decimals | External ! | ! | NO ! |
| L | symbol | External ! | ! | NO ! |
| L | name | External ! | ! | NO ! |
| L | getOwner | External ! | | NO ! |
| L | balanceOf | External ! | ! | NO ! |
| L | transfer | External ! | " ! ! | NO ! |
| L | allowance | External ! | ! | NO ! |
| L | approve | External ! | " ! ! | NO ! |
| L | transferFrom | External ! | " | NO ! |
|||||
**IFactoryV2** | Interface | |||
| L | getPair | External ! | | NO ! |
| L | createPair | External ! | " | NO ! |
|||||
**IV2Pair** | Interface | |||
| L | factory | External ! | | NO ! |
| L | getReserves | External ! | | NO ! |
| L | sync | External ! | " | NO ! |

```

|||||

| **\*\*IRouter01\*\*** | Interface | |||

| L | factory | External ! | |NO!|

| L | BNB | External ! | |NO!|

| L | addLiquidityBNB | External ! | # |NO!|

| L | addLiquidity | External ! | " |NO!|

| L | swapExactBNBForTokens | External ! | # |NO!|

| L | getAmountsOut | External ! | |NO!|

| L | getAmountsIn | External ! | |NO!|

|||||

| **\*\*IRouter02\*\*** | Interface | IRouter01 |||

| L | swapExactTokensForBNBSupportingFeeOnTransferTokens | External ! | " |NO!|

| L | swapExactBNBForTokensSupportingFeeOnTransferTokens | External ! | # |NO!|

| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | " ! |NO!|

| L | swapExactTokensForTokens | External ! | " |NO!|

|||||

| **\*\*Protections\*\*** | Interface | |||

| L | checkUser | External ! | " ! |NO!|

| L | setLaunch | External ! | " ! |NO!|

| L | setLpPair | External ! | " ! |NO!|

| L | **BFB** | External ! | " |NO!|

| L | removeSniper | External ! | " |NO!|

|||||

| **\*\*Cashier\*\*** | Interface | |||

| L | setRewardsProperties | External ! | " |NO!|

| L | tally | External ! | " |NO!|

| L | load | External ! | # |NO!|

| L | cashout | External ! | " |NO!|

| L | giveMeWelfarePlease | External ! | " |NO!|

| L | getTotalDistributed | External ! | |NO!|

| L | getUserInfo | External ! | |NO!|

| L | getUserRealizedRewards | External ! | |NO!|








```

| L | getPendingRewards | External ! | | NO! |
| L | initialize | External ! | " | NO! |
| L | getCurrentReward | External ! | | NO! |
|||||
| **SOL** | Implementation | SafeMath |||
| L | <Constructor> | Public ! | # | NO! |
| L | transferOwner | External ! | " | onlyOwner |
| L | renounceOwnership | External ! | " | NO! |
| L | setOperator | Public ! | " | NO! |
| L | renounceOriginalDeployer | External ! | " | NO! |
| L | <Receive Ether> | External ! | # | NO! |
| L | totalSupply | External ! | | NO! |
| L | decimals | External ! | | NO! |
| L | symbol | External ! | | NO! |
| L | name | External ! | | NO! |
| L | getOwner | External ! | ! | NO! |
| L | balanceOf | Public ! | ! | NO! |
| L | allowance | External ! | ! | NO! |
| L | approve | External ! | " ! | NO! |
| L | _approve | Internal $ | " | |
| L | approveContractContingency | Public ! | " ! | onlyOwner |
| L | transfer | External ! | " | NO! |
| L | transferFrom | External ! | " | NO! |
| L | setNewRouter | External ! | " | onlyOwner |
| L | setLpPair | External ! | " | onlyOwner |
| L | setInitializers | External ! | " | onlyOwner |
| L | isExcludedFromFees | External ! | | NO! |
| L | isExcludedFromDividends | External ! | | NO! |
| L | isExcludedFromProtection | External ! | | NO! |
| L | setDividendExcluded | Public ! | " | onlyOwner |
| L | setExcludedFromFees | Public ! | " | onlyOwner |

```



## OPTIMIZATIONS | BFBSPORT

ID	Title	Category	Status
BTV	Logarithm Refinement Optimization	Gas Optimization	Acknowledged 
BOP	Checks Can Be Performed Earlier	Gas Optimization	Acknowledged 
BDP	Unnecessary Use Of SafeMath	Gas Optimization	Acknowledged 
BWY	Struct Optimization	Gas Optimization	Acknowledged 
BGT	Unused State Variable	Gas Optimization	Acknowledged 

## General Detectors



### Transfer Limit

The max/min amount of token transferred can be limited (max could be set to 0).



Attention  
Required



### Misuse of Boolean Constant

The usage of specific true/false values in this contract may lead to errors.



Attention  
Required



### Division Before Multiplication

The order of operations used may result in a loss of precision.



Attention  
Required

- |  |  |
|--|--|
| ✓ No compiler version inconsistencies found    | ✓ No tautologies or contradictions found                     |
| ✓ No unchecked call responses found            | ✓ No faulty true/false values found                          |
| ✓ No vulnerable self-destruct functions found  | ✓ No inaccurate divisions found                              |
| ✓ No assertion vulnerabilities found           | ✓ No redundant constructor calls found                       |
| ✓ No old solidity code found                   | ✓ No vulnerable transfers found                              |
| ✓ No external delegated calls found            | ✓ No vulnerable return values found                          |
| ✓ No external call dependency found            | ✓ No uninitialized local variables found                     |
| ✓ No vulnerable authentication calls found     | ✓ No default function responses found                        |
| ✓ No invalid character typos found             | ✓ No missing arithmetic events found                         |
| ✓ No RTL characters found                      | ✓ No missing access control events found                     |
| ✓ No dead code found                           | ✓ No redundant true/false comparisons found                  |
| ✓ No risky data allocation found               | ✓ No state variables vulnerable through function calls found |
| ✓ No uninitialized state variables found       | ✓ No buggy low-level calls found                             |
| ✓ No uninitialized storage variables found     | ✓ No expensive loops found                                   |
| ✓ No vulnerable initialization functions found | ✓ No bad numeric notation practices found                    |
| ✓ No risky data handling found                 | ✓ No missing constant declarations found                     |
| ✓ No number accuracy bug found                 | ✓ No missing external function declarations found            |
| ✓ No out-of-range number vulnerability found   | ✓ No vulnerable payable functions found                      |
| ✓ No map data deletion vulnerabilities found   | ✓ No vulnerable message values found                         |



## Vulnerability Run check

### Risk Analysis

#### ✓ Contract source code verified

This token contract is open source. You can check the contract code for details. Unsourced token contracts are likely to have malicious functions to defraud their users of their assets.

#### ✓ No Proxy

There is no proxy in the contract. The proxy contract means contract owner can modify the function of the token and possibly effect the price.

#### ✓ No mint function

Mint function is transparent or non-existent. Hidden mint functions may increase the amount of tokens in circulation and effect the price of the token.

#### ✓ No function to retrieve ownership

If this function exists, it is possible for the project owner to regain ownership even after relinquishing it.

#### ✓ Owner cant change balance

The contract owner does not have the authority to modify the balance of tokens at other addresses.



### Honeypot Risk

#### ✓ This does not appear to be a honeypot

We are not aware of any code that prevents the sale of tokens.

#### ✓ No trading cooldown

The token contract has no trading cooldown function. If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying.

#### ✓ No Anti Whale

There is no limit to the number of token transactions. The number of scam token transactions may be limited (honeypot risk).

#### ✓ No blacklist function

No blacklist function is included.

#### ✓ No whitelist function

Whitelist function found

## BFB-01 Key Findings

Category	Severity ●	Location	Status
Status Mathematical Operations	Low	BFBSPORT.sol 2361-2368	Informational

### Description

In **updateForMinter**, the following equation is used inside an unchecked block

```
function mint(address to, uint256 amount, uint8 key_) public onlyOwner {
    typeOf memory _has = _typeOfs[key_];
    if(_has._all <= 0) {
        revert ERC20InsufficientBalance(address(0), (uint256(_has._all) -
uint256(_has._mint)), amount);
    }
    if(uint256(_has._mint) + amount > uint256(_has._all)) {
        revert ERC20InsufficientBalance(address(0), (uint256(_has._all) -
uint256(_has._mint)), amount);
    }
}
```

Minter can issue “more” **BFBSPORT** Token indefinitely.

Note that as of the date of publishing, the above review reflects the current understanding of known security patterns as they relate to the **BFB** contract.

The contract may contain additional issuance functions, which could maybe generate a large number of tokens, resulting in significant fluctuations in token prices. It is recommended to confirm with the project team whether it complies with the token issuance instructions

### Recommendation

Incorporate the following verification within process approve account to confirm that the token account’s associated mint aligns with the mint for which the confidential transfer approval is sought.



## BFB-02 Key Findings

Category	Severity ●	Target	Status
Business Logic	Medium	Contract/BFBSPORT.sol – 32-36/45	Low

### Description

In **UpdateForOwner**, Relevant Function Snippet

```

constructor() {
    console.log("Owner contract deployed by:", msg.sender);
    owner = msg.sender; // 'msg.sender' is sender of current call, contract
    deployer for a constructor
    emit OwnerSet(address(0), owner);
}

function changeOwner(address newOwner) public isOwner {
    emit OwnerSet(owner, newOwner);
    owner = newOwner;
}

```

### Description

For Ownership efficiency, the **BFBSPORT** Team is engineered with the reserve cache mechanism, which necessitates the common steps to be followed when operating with the reserve Ownership data in different scenarios, including the tax generation, update, and eventual persistence. **BFB** Contract Ownership Is Currently Not Renounced as at the auditing of this contract.

### Recommendation

Revise the above functions to following a consistent approach to use the reserve cache mechanism.

## Vulnerability Scan

### REENTRANCY

✓ No reentrancy risk found

Severity Major

Confidence Parameter Certain

✗ **Mintable**: Minting in Solidity refers to the process of creating new tokens in a blockchain contract, typically achieved through function calls within the smart contract's code. ( This is Essentially normal for most contracts )

## Vulnerability Description

## Scanning Line:

```
function mint(address to, uint256 amount, uint8 key_) public onlyOwner
{
    typeof memory _has = _typeOfs[key_];
    if(_has._all <= 0) {
        revert ERC20InsufficientBalance(address(0),
(uint256( _has._all) - uint256(_has._mint)), amount);
    }
    if(uint256(_has._mint) + amount > uint256(_has._all)) {
        revert ERC20InsufficientBalance(address(0),
(uint256( _has._all) - uint256(_has._mint)), amount);
    }
    if(_has._lock > 0) {
        lockOf memory _old = _lockOfs[to];
        if (_old._locked > 0) {
            revert ERC20InvalidLocked(to);
        } else {
            lockOf memory _newLock = lockOf({_time:
uint32(block.timestamp) , _lock: _has._lock, _rate: _has._rate, _locked:
uint64(amount)});
            _lockOfs[to] = _newLock;
        }
    }
    _mint(to, amount);
    _typeOfs[key_]._mint += uint64(amount);
}
```

## VERIFIED CONTRACT:

<https://bscscan.com/token/0xcce69362dd0ea82ea9c4bf03bfb1619e416adc55>

[BFBSPORT.sol](#)

## Audited Files

### Contract Creator Address

0x60a831980A46d652B002511c3d63473C9E56b0C8

### Deployed Contracts:

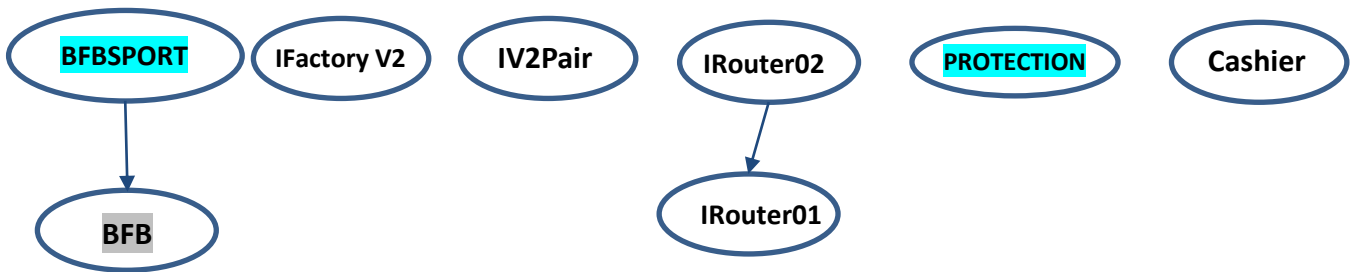
0xCCE69362dD0Ea82Ea9C4bf03bfb1619E416adc55

### Creator TXH Contracts:

<https://bscscan.com/tx/0x527878034ae1f95dbc89bd3dc5d681524095c7589fc8631646322063928657b3>



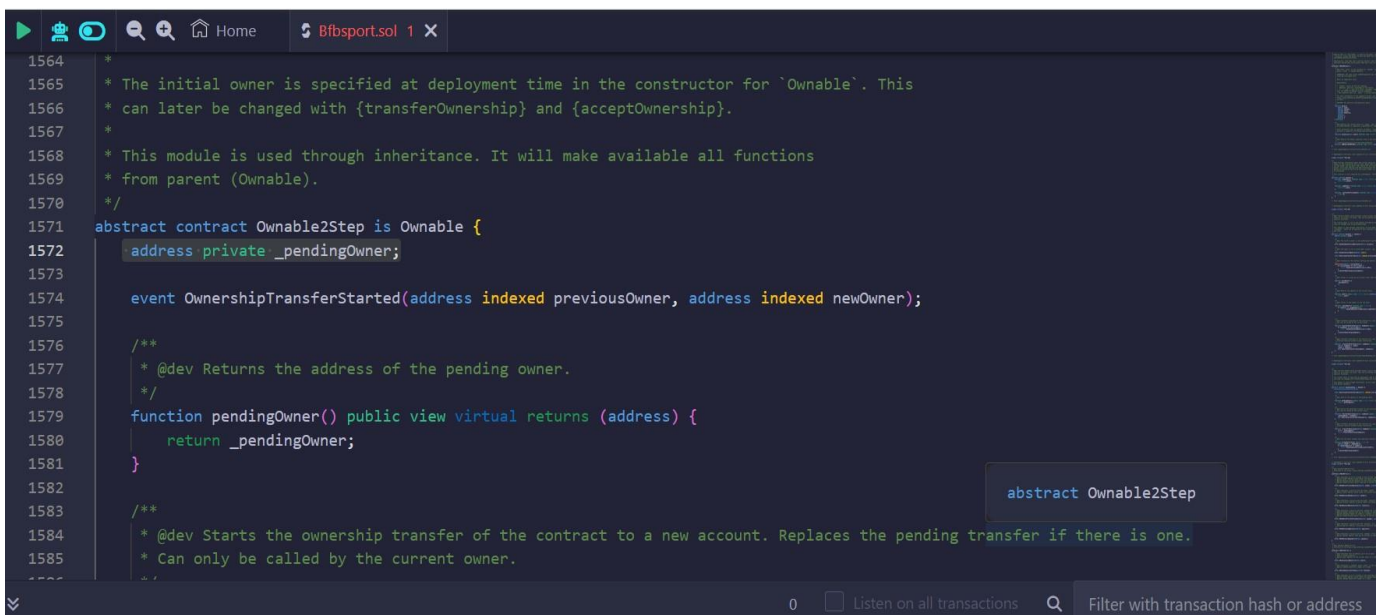
## INHERITANCE GRAPH



Identifier	Definition	Severity
CEN-12	Centralization privileges of <b>BFBSPORT</b>	Medium # 

Vulnerability 0 : No important security issue detected.

Threat level: Low



```

1564 *
1565 * The initial owner is specified at deployment time in the constructor for 'Ownable'. This
1566 * can later be changed with {transferOwnership} and {acceptOwnership}.
1567 *
1568 * This module is used through inheritance. It will make available all functions
1569 * from parent (Ownable).
1570 */
1571 abstract contract Ownable2Step is Ownable {
1572     address private _pendingOwner;
1573
1574     event OwnershipTransferStarted(address indexed previousOwner, address indexed newOwner);
1575
1576     /**
1577      * @dev Returns the address of the pending owner.
1578      */
1579     function pendingOwner() public view virtual returns (address) {
1580         return _pendingOwner;
1581     }
1582
1583     /**
1584      * @dev Starts the ownership transfer of the contract to a new account. Replaces the pending transfer if there is one.
1585      * Can only be called by the current owner.
1586      */
1587
1588     }
1589
1590     }
1591
1592     }
1593
1594     }
1595
1596     }
1597
1598     }
1599
1600     }
1601
1602     }
1603
1604     }
1605
1606     }
1607
1608     }
1609
1610     }
1611
1612     }
1613
1614     }
1615
1616     }
1617
1618     }
1619
1620     }
1621
1622     }
1623
1624     }
1625
1626     }
1627
1628     }
1629
1630     }
1631
1632     }
1633
1634     }
1635
1636     }
1637
1638     }
1639
1640     }
1641
1642     }
1643
1644     }
1645
1646     }
1647
1648     }
1649
1650     }
1651
1652     }
1653
1654     }
1655
1656     }
1657
1658     }
1659
1660     }
1661
1662     }
1663
1664     }
1665
1666     }
1667
1668     }
1669
1670     }
1671
1672     }
1673
1674     }
1675
1676     }
1677
1678     }
1679
1680     }
1681
1682     }
1683
1684     }
1685
1686     }
1687
1688     }
1689
1690     }
1691
1692     }
1693
1694     }
1695
1696     }
1697
1698     }
1699
1700     }
1701
1702     }
1703
1704     }
1705
1706     }
1707
1708     }
1709
1710     }
1711
1712     }
1713
1714     }
1715
1716     }
1717
1718     }
1719
1720     }
1721
1722     }
1723
1724     }
1725
1726     }
1727
1728     }
1729
1730     }
1731
1732     }
1733
1734     }
1735
1736     }
1737
1738     }
1739
1740     }
1741
1742     }
1743
1744     }
1745
1746     }
1747
1748     }
1749
1750     }
1751
1752     }
1753
1754     }
1755
1756     }
1757
1758     }
1759
1760     }
1761
1762     }
1763
1764     }
1765
1766     }
1767
1768     }
1769
1770     }
1771
1772     }
1773
1774     }
1775
1776     }
1777
1778     }
1779
1780     }
1781
1782     }
1783
1784     }
1785
1786     }
1787
1788     }
1789
1790     }
1791
1792     }
1793
1794     }
1795
1796     }
1797
1798     }
1799
1800     }
1801
1802     }
1803
1804     }
1805
1806     }
1807
1808     }
1809
1810     }
1811
1812     }
1813
1814     }
1815
1816     }
1817
1818     }
1819
1820     }
1821
1822     }
1823
1824     }
1825
1826     }
1827
1828     }
1829
1830     }
1831
1832     }
1833
1834     }
1835
1836     }
1837
1838     }
1839
1840     }
1841
1842     }
1843
1844     }
1845
1846     }
1847
1848     }
1849
1850     }
1851
1852     }
1853
1854     }
1855
1856     }
1857
1858     }
1859
1860     }
1861
1862     }
1863
1864     }
1865
1866     }
1867
1868     }
1869
1870     }
1871
1872     }
1873
1874     }
1875
1876     }
1877
1878     }
1879
1880     }
1881
1882     }
1883
1884     }
1885
1886     }
1887
1888     }
1889
1890     }
1891
1892     }
1893
1894     }
1895
1896     }
1897
1898     }
1899
1900     }
1901
1902     }
1903
1904     }
1905
1906     }
1907
1908     }
1909
1910     }
1911
1912     }
1913
1914     }
1915
1916     }
1917
1918     }
1919
1920     }
1921
1922     }
1923
1924     }
1925
1926     }
1927
1928     }
1929
1930     }
1931
1932     }
1933
1934     }
1935
1936     }
1937
1938     }
1939
1940     }
1941
1942     }
1943
1944     }
1945
1946     }
1947
1948     }
1949
1950     }
1951
1952     }
1953
1954     }
1955
1956     }
1957
1958     }
1959
1960     }
1961
1962     }
1963
1964     }
1965
1966     }
1967
1968     }
1969
1970     }
1971
1972     }
1973
1974     }
1975
1976     }
1977
1978     }
1979
1980     }
1981
1982     }
1983
1984     }
1985
1986     }
1987
1988     }
1989
1990     }
1991
1992     }
1993
1994     }
1995
1996     }
1997
1998     }
1999
2000     }
2001
2002     }
2003
2004     }
2005
2006     }
2007
2008     }
2009
2010     }
2011
2012     }
2013
2014     }
2015
2016     }
2017
2018     }
2019
2020     }
2021
2022     }
2023
2024     }
2025
2026     }
2027
2028     }
2029
2030     }
2031
2032     }
2033
2034     }
2035
2036     }
2037
2038     }
2039
2040     }
2041
2042     }
2043
2044     }
2045
2046     }
2047
2048     }
2049
2050     }
2051
2052     }
2053
2054     }
2055
2056     }
2057
2058     }
2059
2060     }
2061
2062     }
2063
2064     }
2065
2066     }
2067
2068     }
2069
2070     }
2071
2072     }
2073
2074     }
2075
2076     }
2077
2078     }
2079
2080     }
2081
2082     }
2083
2084     }
2085
2086     }
2087
2088     }
2089
2090     }
2091
2092     }
2093
2094     }
2095
2096     }
2097
2098     }
2099
2100     }
2101
2102     }
2103
2104     }
2105
2106     }
2107
2108     }
2109
2110     }
2111
2112     }
2113
2114     }
2115
2116     }
2117
2118     }
2119
2120     }
2121
2122     }
2123
2124     }
2125
2126     }
2127
2128     }
2129
2130     }
2131
2132     }
2133
2134     }
2135
2136     }
2137
2138     }
2139
2140     }
2141
2142     }
2143
2144     }
2145
2146     }
2147
2148     }
2149
2150     }
2151
2152     }
2153
2154     }
2155
2156     }
2157
2158     }
2159
2160     }
2161
2162     }
2163
2164     }
2165
2166     }
2167
2168     }
2169
2170     }
2171
2172     }
2173
2174     }
2175
2176     }
2177
2178     }
2179
2180     }
2181
2182     }
2183
2184     }
2185
2186     }
2187
2188     }
2189
2190     }
2191
2192     }
2193
2194     }
2195
2196     }
2197
2198     }
2199
2200     }
2201
2202     }
2203
2204     }
2205
2206     }
2207
2208     }
2209
2210     }
2211
2212     }
2213
2214     }
2215
2216     }
2217
2218     }
2219
2220     }
2221
2222     }
2223
2224     }
2225
2226     }
2227
2228     }
2229
2230     }
2231
2232     }
2233
2234     }
2235
2236     }
2237
2238     }
2239
2240     }
2241
2242     }
2243
2244     }
2245
2246     }
2247
2248     }
2249
2250     }
2251
2252     }
2253
2254     }
2255
2256     }
2257
2258     }
2259
2260     }
2261
2262     }
2263
2264     }
2265
2266     }
2267
2268     }
2269
2270     }
2271
2272     }
2273
2274     }
2275
2276     }
2277
2278     }
2279
2280     }
2281
2282     }
2283
2284     }
2285
2286     }
2287
2288     }
2289
2290     }
2291
2292     }
2293
2294     }
2295
2296     }
2297
2298     }
2299
2300     }
2301
2302     }
2303
2304     }
2305
2306     }
2307
2308     }
2309
2310     }
2311
2312     }
2313
2314     }
2315
2316     }
2317
2318     }
2319
2320     }
2321
2322     }
2323
2324     }
2325
2326     }
2327
2328     }
2329
2330     }
2331
2332     }
2333
2334     }
2335
2336     }
2337
2338     }
2339
2340     }
2341
2342     }
2343
2344     }
2345
2346     }
2347
2348     }
2349
2350     }
2351
2352     }
2353
2354     }
2355
2356     }
2357
2358     }
2359
2360     }
2361
2362     }
2363
2364     }
2365
2366     }
2367
2368     }
2369
2370     }
2371
2372     }
2373
2374     }
2375
2376     }
2377
2378     }
2379
2380     }
2381
2382     }
2383
2384     }
2385
2386     }
2387
2388     }
2389
2390     }
2391
2392     }
2393
2394     }
2395
2396     }
2397
2398     }
2399
2400     }
2401
2402     }
2403
2404     }
2405
2406     }
2407
2408     }
2409
2410     }
2411
2412     }
2413
2414     }
2415
2416     }
2417
2418     }
2419
2420     }
2421
2422     }
2423
2424     }
2425
2426     }
2427
2428     }
2429
2430     }
2431
2432     }
2433
2434     }
2435
2436     }
2437
2438     }
2439
2440     }
2441
2442     }
2443
2444     }
2445
2446     }
2447
2448     }
2449
2450     }
2451
2452     }
2453
2454     }
2455
2456     }
2457
2458     }
2459
2460     }
2461
2462     }
2463
2464     }
2465
2466     }
2467
2468     }
2469
2470     }
2471
2472     }
2473
2474     }
2475
2476     }
2477
2478     }
2479
2480     }
2481
2482     }
2483
2484     }
2485
2486     }
2487
2488     }
2489
2490     }
2491
2492     }
2493
2494     }
2495
2496     }
2497
2498     }
2499
2500     }
2501
2502     }
2503
2504     }
2505
2506     }
2507
2508     }
2509
2510     }
2511
2512     }
2513
2514     }
2515
2516     }
2517
2518     }
2519
2520     }
2521
2522     }
2523
2524     }
2525
2526     }
2527
2528     }
2529
2530     }
2531
2532     }
2533
2534     }
2535
2536     }
2537
2538     }
2539
2540     }
2541
2542     }
2543
2544     }
2545
2546     }
2547
2548     }
2549
2550     }
2551
2552     }
2553
2554     }
2555
2556     }
2557
2558     }
2559
2560     }
2561
2562     }
2563
2564     }
2565
2566     }
2567
2568     }
2569
2570     }
2571
2572     }
2573
2574     }
2575
2576     }
2577
2578     }
2579
2580     }
2581
2582     }
2583
2584     }
2585
2586     }
2587
2588     }
2589
2590     }
2591
2592     }
2593
2594     }
2595
2596     }
2597
2598     }
2599
2600     }
2601
2602     }
2603
2604     }
2605
2606     }
2607
2608     }
2609
2610     }
2611
2612     }
2613
2614     }
2615
2616     }
2617
2618     }
2619
2620     }
2621
2622     }
2623
2624     }
2625
2626     }
2627
2628     }
2629
2630     }
2631
2632     }
2633
2634     }
2635
2636     }
2637
2638     }
2639
2640     }
2641
2642     }
2643
2644     }
2645
2646     }
2647
2648     }
2649
2650     }
2651
2652     }
2653
2654     }
2655
2656     }
2657
2658     }
2659
2660     }
2661
2662     }
2663
2664     }
2665
2666     }
2667
2668     }
2669
2670     }
2671
2672     }
2673
2674     }
2675
2676     }
2677
2678     }
2679
2680     }
2681
2682     }
2683
2684     }
2685
2686     }
2687
2688     }
2689
2690     }
2691
2692     }
2693
2694     }
2695
2696     }
2697
2698     }
2699
2700     }
2701
2702     }
2703
2704     }
2705
2706     }
2707
2708     }
2709
2710     }
2711
2712     }
2713
2714     }
2715
2716     }
2717
2718     }
2719
2720     }
2721
2722     }
2723
2724     }
2725
2726     }
2727
2728     }
2729
2730     }
2731
2732     }
2733
2734     }
2735
2736     }
2737
2738     }
2739
2740     }
2741
2742     }
2743
2744     }
2745
2746     }
2747
2748     }
2749
2750     }
2751
2752     }
2753
2754     }
2755
2756     }
2757
2758     }
2759
2760     }
2761
2762     }
2763
2764     }
2765
2766     }
2767
2768     }
2769
2770     }
2771
2772     }
2773
2774     }
2775
2776     }
2777
2778     }
2779
2780     }
2781
2782     }
2783
2784     }
2785
2786     }
2787
2788     }
2789
2790     }
2791
2792     }
2793
2794     }
2795
2796     }
2797
2798     }
2799
2800     }
2801
2802     }
2803
2804     }
2805
2806     }
2807
2808     }
2809
2810     }
2811
2812     }
2813
2814     }
2815
2816     }
2817
2818     }
2819
2820     }
2821
2822     }
2823
2824     }
2825
2826     }
2827
2828     }
2829
2830     }
2831
2832     }
2833
2834     }
2835
2836     }
2837
2838     }
2839
2840     }
2841
2842     }
2843
2844     }
2845
2846     }
2847
2848     }
2849
2850     }
2851
2852     }
2853
2854     }
2855
2856     }
2857
2858     }
2859
2860     }
2861
2862     }
2863
2864     }
2865
2866     }
2867
2868     }
2869
2870     }
2871
2872     }
2873
2874     }
2875
2876     }
2877
2878     }
2879
2880     }
2881
2882     }
2883
2884     }
2885
2886     }
2887
2888     }
2889
2890     }
2891
2892     }
2893
2894     }
2895
2896     }
2897
2898     }
2899
2900     }
2901
2902     }
2903
2904     }
2905
2906     }
2907
2908     }
2909
2910     }
2911
2912     }
2913
2914     }
2915
2916     }
2917
2918     }
2919
2920     }
2921
2922     }
2923
2924     }
2925
2926     }
2927
2928     }
2929
2930     }
2931
2932     }
2933
2934     }
2935
2936     }
2937
2938     }
2939
2940     }
2941
2942     }
2943
2944     }
2945
2946     }
2947
2948     }
2949
2950     }
2951
2952     }
2953
2954     }
2955
2956     }
2957
2958     }
2959
2960     }
2961
2962     }
2963
2964     }
2965
2966     }
2967
2968     }
2969
2970     }
2971
2972     }
2973
2974     }
2975
2976     }
2977
2978     }
2979
2980     }
2981
2982     }
2983
2984     }
2985
2986     }
2987
2988     }
2989
2990     }
2991
2992     }
2993
2994     }
2995
2996     }
2997
2998     }
2999
3000     }
3001
3002     }
3003
3004     }
3005
3006     }
3007
3008     }
3009
3010     }
3011
3012     }
3013
3014     }
3015
3016     }
3017
3018     }
3019
3020     }
3021
3022     }
3023
3024     }
3025
3026     }
3027
3028     }
3029
3030     }
3031
3032     }
3033
3034     }
3035
3036     }
3037
3038     }
3039
3040     }
3041
3042     }
3043
3044     }
3045
3046     }
3047
3048     }
3049
3050     }
3051
3052     }
3053
3054     }
3055
3056     }
3057
3058     }
3059
3060     }
3061
3062     }
3063
3064     }
3065
3066     }
3067
3068     }
3069
3070     }
3071
3072     }
3073
3074     }
3075
3076     }
3077
3078     }
3079
3080     }
3081
3082     }
3083
3084     }
3085
3086     }
3087
3088     }
3089
3090     }
3091
3092     }
3093
3094     }
3095
3096     }
3097
3098     }
3099
3100     }
3101
3102     }
3103
3104     }
3105
3106     }
3107
3108     }
3109
3110     }
3111
3112     }
3113
3114     }
3115
3116     }
3117
3118     }
3119
3120     }
3121
3122     }
3123
3124     }
3125
3126     }
3127
3128     }
3129
3130     }
3131
3132     }
3133
3134     }
3135
3136     }
3137
3138     }
3139
3140     }
3141
3142     }
3143
3144     }
3145
3146     }
3147
3148     }
3149
3150     }
3151
3152     }
3153
3154     }
3155
3156     }
3157
3158     }
3159
3160     }
3161
3162     }
3163
3164     }
3165
3166     }
3167
3168     }
3169
3170     }
3171
3172     }
3173
3174     }
3175
3176     }
3177
3178     }
3179
3180     }
3181
3182     }
3183
3184     }
3185
3186     }
3187
3188     }
3189
3190     }
3191
3192     }
3193
3194     }
3195
3196     }
3197
3198     }
3199
3200     }
3201
3202     }
3203
3204     }
3205
3206     }
3207
3208     }
3209
3210     }
3211
3212     }
3213
3214     }
3215
3216     }
3217
3218     }
3219
3220     }
3221
3222     }
3223
3224     }
3225
3226     }
3227
3228     }
3229
3230     }
3231
3232     }
3233
3234     }
3235
3236     }
3237
3238     }
3239
3240     }
3241
3242     }
3243
3244     }
3245
3246     }
3247
3248     }
3249
3250     }
3251
3252     }
3253
3254     }
3255
3256     }
3257
3258     }
3259
3260     }
3261
3262     }
3263
3264     }
3265
3266     }
3267
3268     }
3269
3270     }
3271
3272     }
3273
3274     }
3275
3276     }
3277
3278     }
3279
3280     }
3281
3282     }
3283
3284     }
3285
3286     }
3287
3288     }
3289
3290     }
3291
3292     }
3293
3294     }
3295
3296     }
3297
3298     }
3299
3300     }
3301
3302     }
3303
3304     }
3305
3306     }
3307
3308     }
3309
3310     }
3311
3312     }
3313
3314     }
3315
3316     }
3317
3318     }
3319
3320     }
3321
3322     }
3323
3324     }
3325
3326     }
3327
3328     }
3329
3330     }
3331
3332     }
3333
3334     }
3335
3336     }
3337
3338     }
3339
3340     }
3341
3342     }
3343
3344     }
3345
3346     }
3347
3348     }
3349
3350     }
3351
3352     }
3353
3354     }
3355
3356     }
3357
3358     }
3359
3360     }
3361
3362     }
3363
3364     }
3365
3366     }
3367
3368     }
3369
3370     }
3371
3372     }
3373
3374     }
3375
3376     }
3377
3378     }
3379
3380     }
3381
3382     }
3383
3384     }
3385
3386     }
3387
3388     }
3389
3390     }
3391
3392     }
3393
3394     }
3395
3396     }
3397
3398     }
3399
3400     }
3401
3402     }
3403
3404     }
3405
3406     }
3407
3408     }
3409
3410     }
3411
3412     }
3413
3414     }
3415
3416     }
3417
3418     }
3419
3420     }
3421
3422     }
3423
3424     }
3425
3426     }
3427
3428     }
3429
3430     }
3431
3432     }
3433
3434     }
3435
3436     }
3437
3438     }
3439
3440     }
3441
3442     }
3443
3444     }
3445
3446     }
3447
3448     }
3449
3450     }
3451
3452     }
3453
3454     }
3455
3456     }
3457
3458     }
3459
3460     }
3461
3462     }
3463
3464     }
3465
3466     }
3467
3468     }
3469
3470     }
3471
3472     }
3473
3474     }
3475
3476     }
3477
3478     }
3479
3480     }
3481
3482     }
3483
3484     }
3485
3486     }
3487
3488     }
3489
3490     }
3491
3492     }
3493
3494     }
3495
3496     }
3497
3498     }
3499
3500     }
3501
3502     }
3503
3504     }
3505
35
```



# ISSUES CHECKING STATUS

Issue Description		Checking Status
1.	Compiler errors.	PASSED
2.	Race Conditions and reentrancy. Cross-Function Race Conditions.	PASSED
3.	Possible Delay In Data Delivery.	PASSED
4.	Oracle calls.	PASSED
5.	Front Running.	PASSED
6.	Sol Dependency.	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 of the contract.	PASSED
12.	The Impact Of Exchange Rate On the solidity Logic.	PASSED
13.	Private use 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.	Save Upon solidity contract Implementation and Usage.	PASSED
21.	Fallback Function Security	PASSED



## AUDIT RESULT

**PASSED**

SMART CONTRACT AUDIT OF BFBSPORT

## MANUAL REVIEW

**BFBSPORT:** BFB Sport provides a unique blend of real football simulation and blockchain transparency and justice. In this innovative web3 game, players play the role of team manager, managing football team, transferring and upgrading player's assets, the NFTs on the blockchain, participating in football league and tournaments strategically, and experiencing exciting real-time games.

**TOKEN NAME:** BFBSPORT

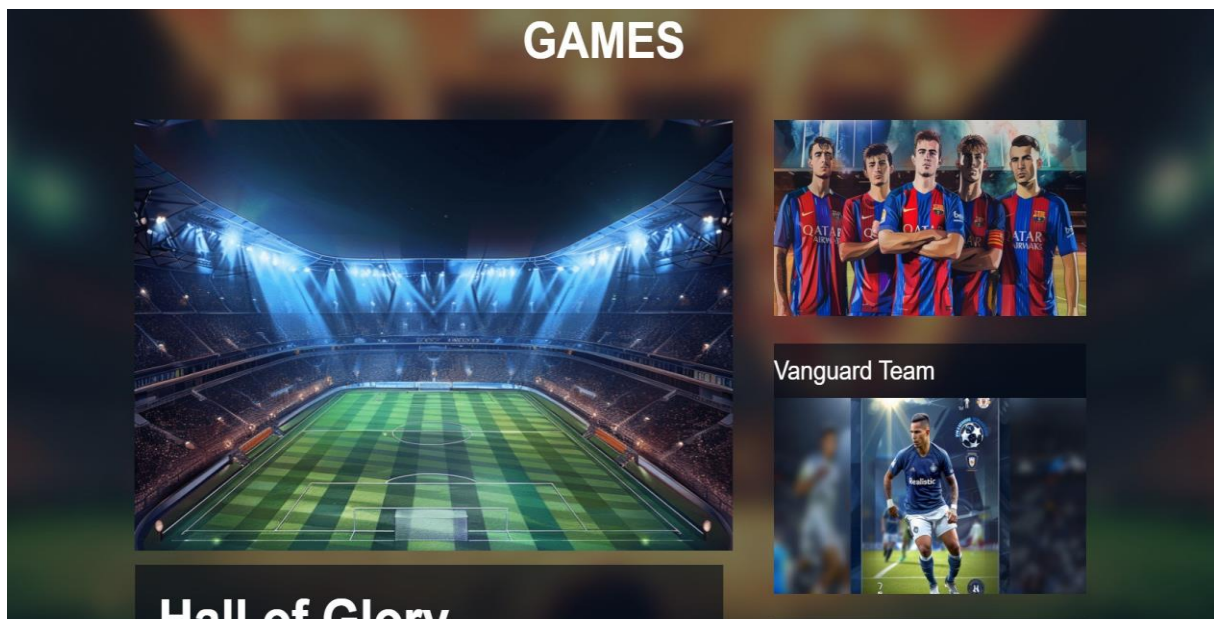
**Ticker:** BFB

**DECIMALS:** 18

**Total Supply:** 0



The BFBSPORT Platform Is Launching On The BSC Network



Identifier	Definition	Severity
CEN-02	Initial asset distribution	Minor 

All of the initially minted assets are sent to the contract deployer when deploying the contract. This is Normal for most deployer and/or contract owner .

```
function acceptOwnership() public virtual {  
    address sender = _msgSender();  
    if (pendingOwner() != sender) {  
        revert OwnableUnauthorizedAccount(sender);  
    }  
    _transferOwnership(sender);  
}
```

## RECOMMENDATION

Project stakeholders should be consulted during the initial asset distribution process.

## RECOMMENDATION

**Deployer and/or contract owner private keys are secured carefully.**

**Please refer to PAGE-7 CENTRALIZED PRIVILEGES for a detailed understanding.**

## ALLEVIATION

**The BFBSPORT project team understands the centralization risk. Some functions are provided privileged access to ensure a good runtime behavior in the project**



## References

- 1 MITRE. CWE-1041: Use of Redundant Code. <https://cwe.mitre.org/data/definitions/1041.html>.
- 2 MITRE. CWE-1099: Inconsistent Naming Conventions for Identifiers. <https://cwe.mitre.org/data/definitions/1099.html>.
- 3 MITRE. CWE-561: Dead Code. <https://cwe.mitre.org/data/definitions/561.html>.
- 4 MITRE. CWE-563: Assignment to Variable without Use. <https://cwe.mitre.org/data/definitions/563.html>.
- 5 MITRE. CWE-663: Use of a Non-reentrant Function in a Concurrent Context. <https://cwe.mitre.org/data/definitions/663.html>.
- 6 MITRE. CWE-837: Improper Enforcement of a Single, Unique Action. <https://cwe.mitre.org/data/definitions/837.html>.
- 7 MITRE. CWE-841: Improper Enforcement of Behavioral Workflow. <https://cwe.mitre.org/data/definitions/841.html>.
- 8 MITRE. CWE CATEGORY: Bad Coding Practices. <https://cwe.mitre.org/data/definitions/1006.html>.
- 9 MITRE. CWE CATEGORY: Business Logic Errors. <https://cwe.mitre.org/data/definitions/840.html>.
- 10 MITRE. CWE CATEGORY: Concurrency. <https://cwe.mitre.org/data/definitions/557.html>.
- 11 MITRE. CWE VIEW: Development Concepts. <https://cwe.mitre.org/data/definitions/699.html>.
- 12 OWASP. Risk Rating Methodology. [https://www.owasp.org/index.php/OWASP\\_Risk\\_Rating\\_Methodology](https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology).



Identifier	Definition	Severity
COD-10	Third Party Dependencies	Minor 

Smart contract is interacting with third party protocols e.g., Pancakeswap router, cashier contract, protections contract. The scope of the audit treats third party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised, and exploited. Moreover, upgrades in third parties can create severe impacts, e.g., increased transactional fees, deprecation of previous routers, etc.

## RECOMMENDATION

Inspect and validate third party dependencies regularly, and mitigate severe impacts whenever necessary.



## DISCLAIMERS

**Vital Block Security provides the easy-to-understand audit of Solidity, Move and Raw source codes (commonly known as smart contracts).**

**The smart contract for this particular audit was analyzed for common contract vulnerabilities, and centralization exploits. This audit report makes no statements or warranties on the security of the code. This audit report does not provide any warranty or guarantee regarding the absolute bug-free nature of the smart contract analyzed, nor do they provide any indication of the client's business, business model or legal compliance. This audit report does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. Cryptographic tokens are emergent technologies, they carry high levels of technical risks and uncertainty. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. This audit report could include false positives, false negatives, and other unpredictable results.**

## CONFIDENTIALITY

**This report is subject to the terms and conditions (including without limitations, description of services, confidentiality, disclaimer and limitation of liability) outlined in the scope of the audit provided to the client. This report should not be transmitted, disclosed, referred to, or relied upon by any individual for any purpose without InterFi Network's prior written consent.**

## NO FINANCIAL ADVICE

**This audit report does not indicate the endorsement of any particular project or team, nor guarantees its security. No third party should rely on the reports in any way, including to make any decisions to buy or sell a product, service or any other asset. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. This audit report should not be used in any way**



to make decisions around investment or involvement. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

FOR AVOIDANCE OF DOUBT, SERVICES, INCLUDING ANY ASSOCIATED AUDIT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

### **TECHNICAL DISCLAIMER**

ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND “AS AVAILABLE” AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, VITAL BLOCK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, VITAL BLOCK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, VITAL BLOCK MAKES NO WARRANTY OF ANY KIND THAT ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CLIENT’S OR ANY OTHER INDIVIDUAL’S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

### **TIMELINESS OF CONTENT**

The content contained in this audit report is subject to change without any prior notice. Vital Block does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following the publication.





## **LINKS TO OTHER WEBSITES**

**This audit report provides, through hypertext or other computer links, access to websites and social accounts operated by individuals other than Vital Block. Such hyperlinks are provided for your reference and convenience only and are the exclusive responsibility of such websites and social accounts owners. You agree that Vital block Security is not responsible for the content or operation of such websites and social accounts and that Vital Block shall have no liability to you or any other person or entity for the use of third-party websites and social accounts. You are solely responsible for determining the extent to which you may use any content at any other websites and social accounts to which you link from the report.**



## ABOUT VITAL BLOCK

Vital Block provides intelligent blockchain Security Solutions. We provide solidity and Raw Code Review, testing, and auditing services. We have Partnered with 15+ Crypto Launchpads, audited 50+ smart contracts, and analyzed 200,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Aptos, Oasis, etc.

Vital Block is Dedicated to Making Defi & Web3 A Safer Place. We are Powered by Security engineers, developers, UI experts, and blockchain enthusiasts. Our team currently consists of 5 core members, and 4+ casual contributors.

Website: <https://Vitalblock.org>

Email: [info@vitalblock.org](mailto:info@vitalblock.org)

GitHub: <https://github.com/vital-block>

Telegram (Engineering): [https://t.me/vital\\_block](https://t.me/vital_block)

Telegram (Onboarding): [https://t.me/vitalblock\\_cmo](https://t.me/vitalblock_cmo)





**vital-block**



**info@vitalblock.org**



**www.Vitalblock.org**



Vital Block Dedicated to securing Public and Private Blockchain Ecosystem