FusionTech

BrcQuest

Security Assessment

www.brcquest.com/

TABLE OF CONTENTS

Summary

This report has been prepared to discover issues and vulnerabilities in the source code of the project as well as any contract dependencies that were not part of an officially recognized library.A comprehensive examination has been performed, utilizing Static Analysisand Manual Reviewtechniques. The auditingprocess pays specialattention to the following considerations:

Testing the smart contracts against both commonand uncommon attackvectors. Assessing the codebase to ensure compliancewith current best practices and industry standards.Ensuring contract logic meets the specifications and intentions of the client.
Cross referencing contract structure and implementation againstsimilar smart contracts produced by industryleaders.
Thorough line-by-line manual review of the entirecodebase by industryexperts.

The security assessment resulted in findings that ranged from critical to informational. We recommendaddressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could betterserve the projectfrom the securityperspective:

Enhance generalcoding practices for better structures of source codes;Add enough unit tests to cover the possible use cases;
Provide more comments per each function for readability, especiallycontracts that are verified in public;
Provide more transparency on privileged activities once the protocolis live.

Project Summary

Project Name	BrcQuest - (https://brcquest.com/)
Platform	Binance Smart Chain
Language	Solidity
Codebase	https://www.bscscan.com/address/0xdfe967119c3447af6272662db4e7f871ddc80d8b#code
Commit	805013253b3de4b9c8dd72ae7f27b6b520917f6a16a62fafbe83dd6e30073f93

Audit Summary

Delivery Date	JULY 31, 2023
Audit Methodology	Static Analysis, Manual Review
Key Components	BSCToken

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informational	0	0	0	0	0	0
Discussion	0	0	0	0	0	0

Overview

External Dependencies

Thecontract serves as the underlying entity to interactwith third-party protocols (token- wapping). The scopeof the audit treats third-party entities as blackboxes and assumestheir functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolenassets.

Privileged Functions

The contract contains the followingprivileged functions that are restricted by role with the modifier. They are used to modify the contractconfigurations and addressattributes. We grouped these functions below.

- allowance()
- balanceof()
- decimals()
- name()
- owner()
- symbol()
- totalsupply()
- tradingenabled()

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plan to invoke the aforementioned functions should be also considered to move to the execution queue of the Timelock contract.

Audit Scope

ID	File	SHA256 Checksum
CKP	contract.sol	f79198f1e334d2889b0de0d9507c2bf3e16e6299f37d30102d9496b69c383809

01 | Centralization Risk in Function

Description

The addLiquidity()_hasLiqBeenAdded() function calls the uniswapV2Router.addLiquidityETH function with the to() address specified as owner() for acquiring the generated LP tokens from the corresponding pool. As a result, over time the _owner address will accumulate a significant portionof LP tokens. If _owner the is an EOA (Externally Owned Account),mishandling of its private key can have devastating consequences to the projectas a whole.

Recommendation

We advise to() the address of the uniswapV2Router.addLiquidityETH() function call to be replaced by the contract() itself, i.e. address(this) , and to restrict the managementof the LP tokens within the scope of the contract's businesslogic. This will also protectthe LP tokens from being stolen if the _owner() account is compromised. In general, we strongly recommend centralized privileges or roles in the protocolto be improved via a decentralized mechanism or via smart-contract based accounts with enhanced securitypractices, f.e.Multisignature wallets().

Indicatively, here are some feasible solutionsthat would also mitigate the potential risk:

- Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to preventsingle point of failure due to the private key;
- Introduction of a DAO / governance / voting moduleto increase transparency and user involvement

02 | Contract Gains Non-withdrawable BSC via the _owner Function

Function

Category	Severity	Location	Status
Logical Issue	Medium	projects/contract.sol (98ba012): 817	Acknowledged

Description

The swapAndLiquify() function converts half of the contractTokenBalance() tokens to BSC. The other half of the tokens and partof the converted BSC are deposited into the corresponding pool on pancakeswap as liquidity. For every swap&liquify() function call, a small amountof BSC leftover in the contract. This is due to the price of drops after swapping the first half of tokens into BSCs, and the other half of tokens require less than the converted BSC to be paired with it when adding liquidity. The contract doesn't appear to provide a way to withdraw those BSC, and they will be locked in the contract forever.

Recommendation

It'snot ideal that more and more BSCare lockedinto the contract over time. The simplest solution is to add a withdraw() function in the contract to withdrawBSC. Other approaches that benefit the token holders add a can be:

- Distribute BSC to token holders proportional to the amount of token they hold.
- Use leftover BSC to buy back tokens from the market to increase the token price.

03 | Initial Token Distribution

Category	Severity	Location	Status
Logical Issue	Minor	projects/contract.sol (98ba012): 497	Acknowledged

Description

All of the tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute those tokens withoutobtaining the consensusof the community.

Recommendation

We recommend the team to be transparent regarding the initialtoken distribution process.

04 | Lack of Return Value Handling

Category	Severity	Location	Status
Volatile Code	Minor	projects/contract.sol (98ba012): 843	Acknowledged

Description

The return valuesof function addLiquidityETH() are properly handled.

Recommendation

We recommend using variables to receive the return value of the functions mentionedabove and handleboth success and failure cases if neededby the business logic.

05 | Potential Sandwich Attacks

Category	Severity	Location	Status
Logical Issue	Minor	projects/contract.sol (98ba012): 832~838, 843~850	Acknowledged

Description

A sandwich attack might happen when an attackerobserves a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (beforethe transaction being attacked) a transaction to purchase one of the assets and make profitsby backrunning (afterthe transaction beingattacked) a transaction to sell the asset.

The following functionsare called withoutsetting restrictions on slippage or minimum outputamount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

Recommendation

We recommend settingreasonable minimum output amounts, insteadof 0, based on token prices when calling the aforementioned functions.

06 | Lack of Error Message

Category	Severity	Location	Status
Coding Style	● Informational	projects/contract.sol (98ba012): 560	ⓘ Acknowledged

Description

The require statement can be used to check for conditions and throw an exception if the condition is not met.It is better to providea string messagecontaining details about the errorthat will be passed back to the caller.

Recommendation

We advise refactoring the linked codes as below:

```
560         _approve(_msgSender(), spender, _allowances[_msgSender()]
[spender].add(addedValue), "increase allowance overflow");
```

07 | Redundant Code

Category	Severity	Location	Status
Logical Issue	● Informational	projects/contract.sol (98ba012): 862	ⓘ Acknowledged

Description

The condition! _isExcluded[sender] & !_isExcluded[recipient]
can be included in else .

Recommendation

The following code can be removed:

```
... else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
    _transferStandard(sender, recipient, amount);
} ...
```

08 | Potential Resource Exhaustion

Category	Severity	Location	Status
Logical Issue	● Informational	projects/contract.sol (98ba012): 614, 709	ⓘ Acknowledged

Description

The farloop() within functions includeInReward(address) and _getCurrentSupply() takes the variable _excluded.length(), as the maximal iterationtimes. If the size of the array is very large, it could exceed the gas limit to execute the functions. In this case, the contract might suffer from DoS (Denial of Service) situation.

Recommendation

We recommend the team reviewthe design and ensure this would not cause loss to the project.

09 | Inconsistency Between Comment and Code

Category	Severity	Location	Status
Inconsistency	● Informational	projects/contract.sol (98ba012): 230~236	ⓘ Acknowledged

Description

According to the commentin L238, the lock() function will lock the contract **for a given time period**. However, the code implementation will lock the contract **until the given timestamp.**

```
238     //Unlocks the contract for owner when _lockTime is exceeds
239     function unlock() public virtual {
240         require(_previousOwner == msg.sender, "You don't have permission to
unlock.");
241         require(block.timestamp > _lockTime , "Contract is locked.");
242         emit OwnershipTransferred(_owner, _previousOwner);
243         _owner = _previousOwner;
244     }
```

Recommendation

We recommend the team reviewthe design and update either comments or code implementation to ensure consistent logic between code and comment.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanismto relocate funds.

Logical Issue

Logical Issue findingsdetail a faultin the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Volatile Code

Volatile Code findingsrefer to segmentsof code that behave unexpectedly on certain edge cases that may resultin a vulnerability.

Coding Style

Coding Style findingsusually do not affect the generated byte-code but rather commenton how to make the codebase more legible and, as a result, easilymaintainable.

Inconsistency

Inconsistency findings referto functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing differentrequire statements on the input variables than a setterfunction.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specifiedcommit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" commandagainst the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimerand limitation of liability) set forth in the ServicesAgreement, or the scope of services, and terms and conditions providedto you ("Customer" or the "Company") in connection with the Agreement. This report providedin connection with the Serviceset forth in the Agreementshall be used by the Company only to the extent permittedunder the terms and conditions set forth in the Agreement.
This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes,nor may copiesbe delivered to any otherperson other than the Company,without FusionTech prior written consentin each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts FusionTech to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, businessmodel or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor shouldbe leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. FusionTech's position is that each companyand individual are responsible for their own due diligenceand continuous security.
FusionTech's goal is to help reduce the attack vectorsand the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by FusionTech is subject to dependencies and under continuing development. You agree that your access and/or use, includingbut not limited to any services, reports,and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokensare emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon,multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTSOR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUTWARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, FusionTech HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, FusionTech SPECIFICALLY DISCLAIMS ALL IMPLIEDWARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, FusionTech MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE,ACCURATE, COMPLETE, FREE OF HARMFULCODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, FusionTech PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDSOR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHERFusionTech NOR ANY OF FusionTech'SAGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. FusionTech WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTINGFROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES,ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED"AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THETHIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSENOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIESBE DELIVERED TO, ANY OTHERPERSON WITHOUT FusionTech'SPRIOR WRITTEN CONSENTIN EACH INSTANCE.

NOTHIRD PARTY OR ANYONE ACTINGON BEHALF OF ANY THEREOF,SHALL BE A THIRD PARTYOR OTHER BENEFICIARY OF SUCH SERVICES,ASSESSMENT REPORT, AND ANY ACCOMPANYING

FORAVOIDANCE OF DOUBT, THE SERVICES,INCLUDING ANY ASSOCIATEDASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2022 by leading academics in the field of Computer Science, FusionTech is going to be a leading blockchain security company that serves to verify the security and correctness of smart contracts KYC and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of ourclients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

FusionTech

https://fusiontech.live

https://twitter.com/fusiontechh

https://t.me/fusiontechofficial

https://github.com/fusiontechofficial