SECURICHAIN

SMART CONTRACT AUDIT

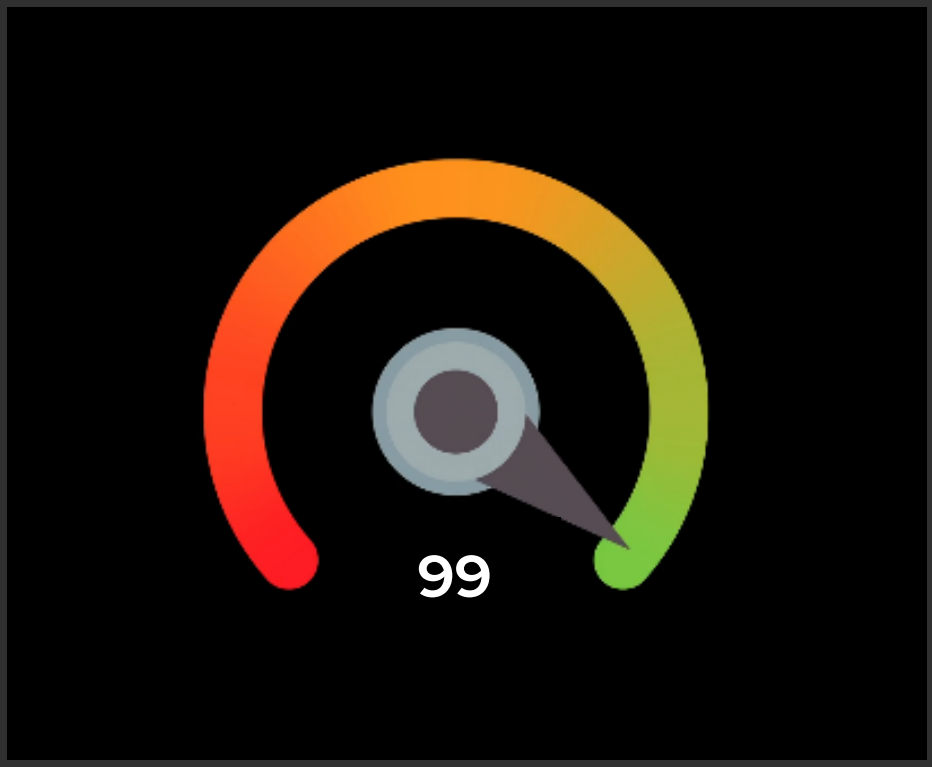
SECURITY ANALYSIS REPORT FOR

beFITTER

August 16th , 2022

SECURICHAIN

Security Rating



The rating is based on the number, severity and latest status of detected issues

beFITTER

SECURICHAIN

Disclaimer

This report containing confidential information which can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

SecuriChain does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed.

The report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

beFITTER

SECURICHAIN

TABLE OF CONTENTS

beFITTER

SECURICHAIN

VULNERABILITY ASSESSMENT OVERVIEW

1.1. ASSIGNING RISK LEVELS

The Auditor categorizes each of the detected vulnerabilities into 4 levels (High, Medium, Low, and Info) according to the degree of the risks it may cause in the Customer's operations. For details of the rating standards, please refer to "Appendix 2 Risk Rating." Please also note that the assessment of the findings is based on Auditor's own perspective and may contain speculations in some cases.

SECURICHAIN

1.2. SCOPE OF WORK

Project Name	**BeFitter Contract**
Platform	**Ethereum**
Languages	**Solidity**
Methods	**Automation scan, architecture review, functional testing, manual code review**
Scope	**https://github.com/hungnd26/beFITTER Commit: de3cbeb00e5ecd6b7e53e35e330fa11256ec9cb2**
Documents	
Timelines	**July 26th 2022 – August 5th 2022**

SECURICHAIN

1.3. CHECKSUM FILE

SCOPE

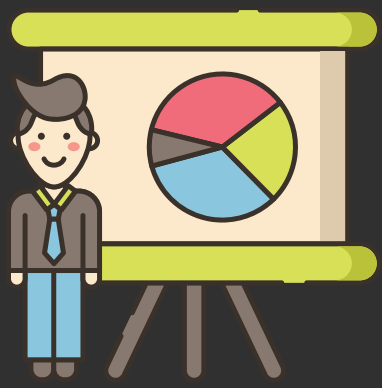
No.	Contract Address	Name
1	fc85bea0ea04b67ab71510d58ca1259513daa7f09cbe111304d64382473d08ec	beFITTER.sol
2	761e6dcee45e47c1e7862a778006d1e22e71b3156cddbb29e47fd46fd8386bb0	BF721.sol
3	076ae86697b1645eb87c1bfc9fc421b147efd0b0cef72925c25f0d1a35c8c08b	BFBox.sol
4	73a1d7e0ffbb53a535aaff066ad7f880a66e2b3fd1587db73c36e7cd2d9a08ba	BFHealth.sol
5	3c0282db38bd5cc7797f7662892f798482f5198279c11439378ef66e33450888	BFOperator.sol
6	0004ee9c15c8d08b16e6fb417c4f7199b40357550b40c3a95caa2368160d80f0	BFPass.sol
7	f67382e361a988749a5781f07fec882502e315584c61eca968097ab6d794f11c	BFShoes.sol
8	6cbec82409cce3ce54d3820e2b0015b509d2f8b6329c1d7b5764e2b5b01e2a1d	BFWallet.sol
9	80dabc8319430c558a6319233f72ca9451ed8231ae945b6631fc6c4f14af6004	FitterPassMinter.sol

SECURICHAIN

1.4. ASSESSMENT RESULTS

According to the assessment, the Customer's smart contracts have a security rating of 99/100

RATE	DESCRIPTION
96-100	**No vulnerabilities** were found or all detected ones have been resolved
70-95	Unresolved **Low-level** vulnerabilities exist
40-69	Unresolved **Medium-level** vulnerabilities exist
0-39	Unresolved **High-level** vulnerabilities exist



For more information on criteria for risk rating, refer to Appendix.2

SECURICHAIN

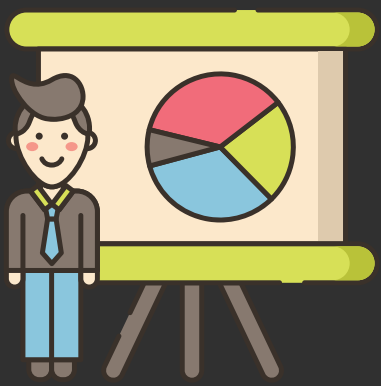
FINDINGS

2.1 List of Vulnerabilities

The detected vulnerabilities are listed below. Please refer to "Appendix.2 Risk Rating" for the risk assessment method.

Vulnerabilities distributed in the smart contract

ID	Risk Level	Name	Amount	Status
SC1	Low	Public function that could be declared external	7	Acknowledged



For rating each vulnerability, refer to Appendix 2.

SECURICHAIN

[1] A public function that could be declared external

Low: 7

Overview

Public functions that are never called by the contract should be declared external to save gas.

```
7  abstract contract FitterPassMinter is BeFitterOperator {
8
9      event MintFitterPass(address indexed to, uint256 tokenId, uint256 amount)
10
11     function mint(address to, uint256 tokenId, uint256 amount)
12         public virtual onlyOperators {}
13
14 }
15
```

Recommendation

Use the external attribute for functions never called from the contract.

Location

- BFPass.mint(address, uint256, uint256) (BFPass.sol#61-69)
- FitterPassMinter.mint(address, uint256, uint256) (FitterPassMinter.sol#11-12)
- BF721.getOwnedTokens(address) (BF721.sol#39-45)
- BFBox.setBaseTokenURI(string memory) (BFBox.sol#30-35)
- BFBox.mint(address, string memory) (BFBox.sol#41-51)
- BFBox.transferWithMessage(address, uint256, string memory) (BFBox.sol#63-70)
- BFBox.getBoxType (uint256) (BFBox.sol#72-76)

SECURICHAIN

CONCLUSION

This document, and its appendices, represents the results of several days of our intensive work.

Smart contracts within the scope were analyzed with static analysis tools and manually reviewed.

Please feel free to direct any questions on this assessment to: audit@securichain.io.

beFITTER

SECURICHAIN

APPENDIX 1: ASSESSMENT LIST

	CHECKLIST	
	Integer Overflow/Underflow	Integer Overflow/Underflow
Arithmetic operations	Integer Truncation	Integer Sign
	Wrong Operator	
Re-entrancy		
Bad Randomness	Timestamp Dependence	Blockhash
Front running		
DDos	DOS By Complex Fallback Function	DOS By Gaslimit
	DOS By Non-existent Address Or Malicious Contract	
Gas usage	Invariants in Loop	Invariants State Variables Are Not Declared Constant
Unsafe external calls		
Business Logics Review		
Access Control & Authorization	Replay Attack	Use tx.origin For Authentication
Logic Vulnerability		

SECURICHAIN

APPENDIX 2: LIST RATING

Risk Level	Explain	Example Types
High	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.	Re-entrancy Front running DDos Bad Randomness Logic Vulnerability Arithmetic operations
Medium	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.	Access Control Unsafe external calls Business Logics Review Logic Vulnerability
Low	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.	Gas Usage
Info	The issue does not pose an immediate risk, but is relevant to security best practices or Defense in Depth.	Do not specify a specific version of Solidity