InterFi NETWORK

@INTERFINETWORK

SMART CONTRACT SECURITY AUDIT OF

CURE TOKEN STAKING

SMART CONTRACT AUDIT | TEAM KYC | PROJECT EVALUATION

RELENTLESSLY SECURING THE PUBLIC BLOCKCHAIN | MADE IN CANADA

Summary

Auditing Firm	InterFi Network
Architecture	InterFi "Echelon" Auditing Standard
Smart Contract Audit Approved By	Chris | Blockchain Specialist at InterFi Network
Platform	Solidity
Audit Check (Mandatory)	Static, Software, Auto Intelligent & Manual Analysis
Consultation Request Date	November 28, 2021
Report Date	November 29, 2021 (24h fast-tracked)

Audit Summary

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

- **Cure Token's staking contract source code has LOW RISK SEVERITY.**
- **Cure Token's staking contract has successfully PASSED the smart contract audit.**
- **This audit is for Cure Token's staking contract. If you're looking for token audit, please visit**

https://github.com/interfinetwork/smart-contract-audits/blob/main/CureToken_AuditReport_InterFi.pdf

For the detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit. The staking contract isn't deployed on the blockchain at the time of the audit.

Table Of Contents

Project Overview

InterFi was consulted by Cure Token on November 28, 2021 to conduct a smart contract security audit of their staking contract code.

Cure Token: A Community Powered Charitable Crypto Token

CURE Token is the first deflationary crypto token built around doing good. By design, this community powered token has exponential price growth benefits for both holders and charities. Cure Token's mission is to spread awareness for underfunded childhood cancers, fund breakthroughs in cancer research and support families afflicted by childhood cancers.

Project	Cure Token Staking
Blockchain	Binance Smart Chain
Contract	Not deployed
Dashboard	https://staking.curetoken.net/
Website	https://www.curetoken.net/
Twitter	https://twitter.com/cure_token
Telegram	https://telegram.me/CureTokenV2
Medium	https://medium.com/@curetoken
Discord	https://discord.com/invite/DuzQCwRfPd

Public logo

Solidity Source Code On GitHub

//Private source code//

Solidity Files under scope

- **CureStaking.sol**

Audited at hash #4c8ef8a9780702c717c4cd1b8f7c19aed49c4b56

- **FundDistributor.sol**

Audited at hash #0ef28067697a21e5cb335e1e64f0db79ed96f0eb

- **IFundDistributor.sol**

Audited at hash #6e3d66b8343e046a4b36707d673381452fc8f3d5

- **Migrations.sol**

Audited at hash #f9c1e69305822e3ba7c4a1933ef144352faac6e7

- **TokenMock.sol**

Audit Scope & Methodology

The scope of this report is to audit the smart contract source code of Cure Token's staking source code.

InterFi has scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category	
Smart Contract Vulnerabilities	❖ Re-entrancy (RE) ❖ Unhandled Exceptions (UE) ❖ Transaction Order Dependency (TO) ❖ Integer Overflow (IO) ❖ Unrestricted Action (UA)
Source Code Review	❖ Ownership Takeover ❖ Gas Limit and Loops ❖ Deployment Consistency ❖ Repository Consistency ❖ Data Consistency ❖ Token Supply Manipulation
Functional Assessment	❖ Access Control and Authorization ❖ Operations Trail and Event Generation ❖ Assets Manipulation ❖ Liquidity Access

InterFi's Echelon Audit Standard

The aim of InterFi's "Echelon" standard is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by ECHELON-1 to assess the smart contract:

1. Solidity smart contract source code reviewal:
   - ❖ Review of the specifications, sources, and instructions provided to InterFi to make sure we understand the size, scope, and functionality of the smart contract.
   - ❖ Manual review of code, which is the process of reading source code line-byline to identify potential vulnerabilities.
2. Static, Manual, and Automated AI analysis:
   - ❖ Test coverage analysis, which is the process of determining whether the test cases are covering the code and how much code is exercised when we run those test cases.
   - ❖ Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts

Automated 3P frameworks used to assess the smart contract vulnerabilities

- ❖ Slither
- ❖ Consensys MythX
- ❖ Consensys Surya
- ❖ Open Zeppelin Code Analyzer
- ❖ Solidity Code Complier

InterFi's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in ETH/BNB. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. Below are the typical risk levels of a smart contract:

Vulnerable: A contract is vulnerable if it has been flagged by a static analysis tool as such. As we will see later, this means that some contracts may be vulnerable because of a false-positive.

Exploitable: A contract is exploitable if it is vulnerable and the vulnerability could be exploited by an external attacker. For example, if the "vulnerability" flagged by a tool is in a function which requires to own the contract, it would be vulnerable but not exploitable.

Exploited: A contract is exploited if it received a transaction on the main network which triggered one of its vulnerabilities. Therefore, a contract can be vulnerable or even exploitable without having been exploited.

Risk severity	Meaning
! Critical	This level vulnerabilities could be exploited easily, and can lead to asset loss, data loss, asset manipulation, or data manipulation. They should be fixed right away.
! High	This level vulnerabilities are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to critical risk severity
! Medium	This level vulnerabilities are 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	This level vulnerabilities can be ignored. They are code style violations, and informational statements in the code. They may not affect the smart contract execution

Smart Contract – Static Analysis

Symbol	Meaning
🛑	Function can be modified
💵	Function is payable
🔒	Function is locked
🔐	Function can be accessed
!	Important functionality

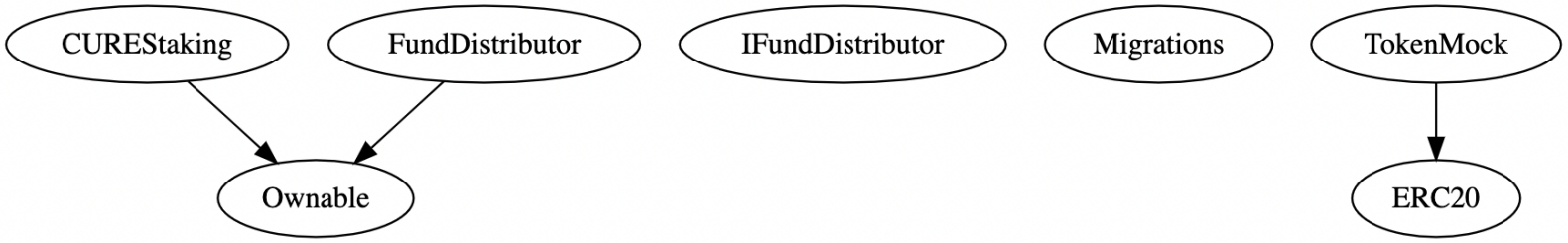
```
| **FundDistributor** | Implementation | Ownable |||
| └ | <Constructor> | Public ! | 🛑 |NO! |
| └ | distributeTo | External ! | 🛑 | onlyRequester |
| └ | setRewardToken | External ! | 🛑 | onlyOwner |
| └ | addRequester | External ! | 🛑 | onlyOwner |
| └ | removeRequester | External ! | 🛑 | onlyOwner |
||||||
| **CUREStaking** | Implementation | Ownable |||
| └ | <Constructor> | Public ! | 🛑 |NO! |
| └ | pendingRewards | Public ! |  |NO! |
| └ | updatePool | Public ! | 🛑 |NO! |
| └ | deposit | External ! | 🛑 |NO! |
| └ | withdraw | Public ! | 🛑 |NO! |
| └ | withdrawAll | External ! | 🛑 |NO! |
| └ | harvest | Public ! | 🛑 |NO! |
| └ | harvestAll | External ! | 🛑 |NO! |
| └ | emergencyWithdraw | External ! | 🛑 |NO! |
| └ | addPool | External ! | 🛑 | onlyOwner |
| └ | getPoolsLength | External ! |  |NO! |
| └ | setFundDistributor | External ! | 🛑 | onlyOwner |
| └ | getLockExpirationTime | Public ! |  |NO! |
| └ | setPoolEndTime | External ! | 🛑 | onlyOwner |
| └ | setMaxStakePerWallet | External ! | 🛑 | onlyOwner |
| └ | setPoolAPYSettings | External ! | 🛑 | onlyOwner |
||||||
| **IFundDistributor** | Interface |  |||
| └ | distributeTo | External ! | 🛑 |NO! |
```

Smart Contract – Software Analysis

Function Signatures

```
d18df53c  =>  pendingRewards(uint256,address)
51eb05a6  =>  updatePool(uint256)
e2bbb158  =>  deposit(uint256,uint256)
441a3e70  =>  withdraw(uint256,uint256)
958e2d31  =>  withdrawAll(uint256)
ddc63262  =>  harvest(uint256)
8ed955b9  =>  harvestAll()
5312ea8e  =>  emergencyWithdraw(uint256)
5d019661  =>  addPool(uint256,uint256,uint256,uint256,IERC20)
ce883cdb  =>  getPoolsLength()
3350dc3c  =>  setFundDistributor(address)
1776ed97  =>  getLockExpirationTime(uint256,address)
5ca73f2b  =>  setPoolEndTime(uint256,uint256)
687e372c  =>  setMaxStakePerWallet(uint256)
5df01008  =>  setPoolAPYSettings(uint256,uint256,uint256)
```


Inheritance Graph



Smart Contract – Manual Analysis

- Cure Token's staking smart contract has a low severity issue which may or may not create any functional vulnerability.

{

"resource": " /Migrations.sol",

"owner": "_generated_diagnostic_collection_name_#0",

"severity": 8, (! Low Severity)

" Wrong argument count for function call: 2 arguments given but expected 1",

"source": "solc",

}

- When the smart contract has an active owner address, some of the smart contract functions can be edited, modified or altered.
- Cure Token's staking contract code is not deployed on any blockchain at the time of the audit. The contract code can be modified or altered after the audit is completed.
- Cure Token's staking contract does not utilize "ReentrancyGuard" to prevent reentrant calls to a function. Reentrancy Guard is a contract module that helps prevent reentrant calls to a function. Inheriting from Reentrancy Guard will make the nonReentrant modifier available, which can be applied to functions to make sure there are no nested (reentrant) calls to them.

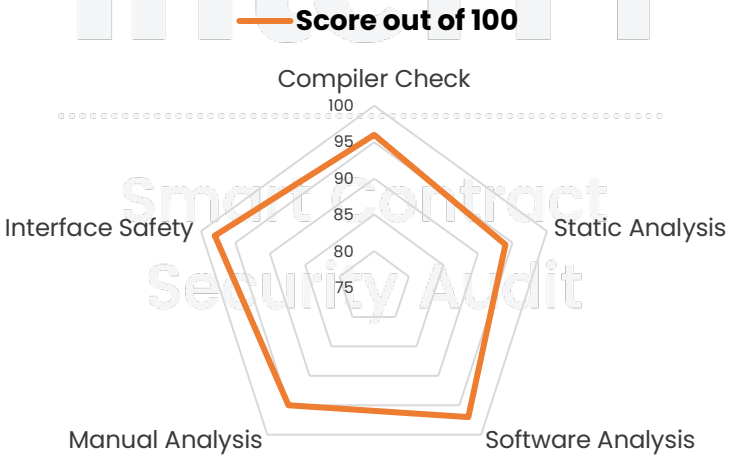
Smart Contract – SWC Attacks

SWC ID	Description	Verdict
SWC-101	Integer Overflow and Underflow	**Passed**
SWC-102	Outdated Compiler Version	**Passed**
SWC-103	Floating Pragma	**Passed**
SWC-104	Unchecked Call Return Value	**Passed**
SWC-105	Unprotected Ether Withdrawal	**Passed**
SWC-106	Unprotected SELFDESTRUCT Instruction	**Passed**
SWC-107	Re-entrancy	**Passed**
SWC-108	State Variable Default Visibility	**Passed**
SWC-109	Uninitialized Storage Pointer	**Passed**
SWC-110	Assert Violation	**Passed**
SWC-111	Use of Deprecated Solidity Functions	**Passed**
SWC-112	Delegate Call to Untrusted Callee	**Passed**
SWC-113	DoS with Failed Call	**Passed**
SWC-114	Transaction Order Dependence	**Passed**
SWC-115	Authorization through tx.origin	**Passed**
SWC-116	Block values as a proxy for time	**Passed**
SWC-117	Signature Malleability	**Passed**
SWC-118	Incorrect Constructor Name	**Passed**

SWC-119	Shadowing State Variables	**Passed**
SWC-120	Weak Sources of Randomness from Chain Attributes	**Passed**
SWC-121	Missing Protection against Signature Replay Attacks	**Passed**
SWC-122	Lack of Proper Signature Verification	**Passed**
SWC-123	Requirement Violation	**Passed**
SWC-124	Write to Arbitrary Storage Location	**Passed**
SWC-125	Incorrect Inheritance Order	**Passed**
SWC-126	Insufficient Gas Griefing	**Passed**
SWC-127	Arbitrary Jump with Function Type Variable	**Passed**
SWC-128	DoS With Block Gas Limit	**Passed**
SWC-129	Typographical Error	**Passed**
SWC-130	Right-To-Left-Override control character (U+202E)	**Passed**
SWC-131	Presence of unused variables	**Passed**
SWC-132	Unexpected Ether balance	**Passed**
SWC-133	Hash Collisions With Multiple Variable Length Arguments	**Passed**
SWC-134	Message call with hardcoded gas amount	**Passed**
SWC-135	Code With No Effects (Irrelevant/Dead Code)	**Passed**
SWC-136	Unencrypted Private Data On-Chain	**Passed**

Smart Contract - Risk Status & Radar Chart

Risk Severity	Status
! Critical	None critical severity issues identified
! High	None high severity issues identified
! Medium	None medium severity issues identified
! Low	1 low severity issues identified
Passed	27 functions and instances verified and passed



Auditor's Verdict

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks.

Cure Token's staking contract source code has LOW RISK SEVERITY.

Cure Token's staking contract has successfully PASSED the smart contract audit.

InterFi

Smart Contract
Security Audit

Important Disclaimer

InterFi Network provides contract auditing and project verification services for blockchain projects. The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project. **This report should not be transmitted, disclosed, referred to, or relied upon by any person for any purposes without InterFi's prior written consent.**

InterFi provides the easy-to-understand assessment of the project, and the smart contract (otherwise known as the source code). The audit makes no statements or warranties on the security of the code. It also cannot be considered as an enough assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have used all the data at our disposal to provide the transparent analysis, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. **Be aware that smart contracts deployed on a blockchain aren't resistant from external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, InterFi does not guarantee the explicit security of the audited smart contract.**

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

This report should not be considered as an endorsement or disapproval of any project or team. The information provided on 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. Do conduct your own due diligence and consult your financial advisor before making any investment decisions.

About InterFi Network

InterFi Network provides intelligent blockchain solutions. InterFi is developing an ecosystem that is seamless and responsive. Some of our services: Blockchain Security, Token Launchpad, NFT Marketplace, etc. **InterFi's mission is to interconnect multiple services like Blockchain Security, DeFi, Gaming, and Marketplace under one ecosystem that is seamless, multi-chain compatible, scalable, secure, fast, responsive, and easy-to-use.**

InterFi is built by a decentralized team of UI experts, contributors, engineers, and enthusiasts from all over the world. Our team currently consists of 6+ core team members, and 10+ casual contributors. **InterFi provides manual, static, and automatic smart contract analysis, to ensure that project is checked against known attacks and potential vulnerabilities.**

To learn more, visit https://interfi.network

To view our audit portfolio, visit https://github.com/interfinetwork

To book an audit, message https://t.me/interfiaudits

@INTERFINETWORK

RELENTLESSLY SECURING THE PUBLIC BLOCKCHAIN | MADE IN CANADA