InterFi
Network

@interfinetwork

SMART CONTRACT SECURITY AUDIT

Scorpion Finance

AUDITED ON SEPTEMBER 02, 2021

USING INTERFI AUDITING ARCHITECTURE

Summary

Audit:

Auditing Firm	InterFi Network
Architecture	InterFi Auditing Architecture
Smart Contract Audit Approved By	Chris | Blockchain Specialist at InterFi
Project Overview Approved BY	Albert | Project Specialist at InterFi
Platform	Solidity
Audit Check (Mandatory)	Vulnerability Check, Source Code Review, Functional Test
Project Check (Optional)	Website Review, Socials Review, Token Review (Not Applicable)
Consultation Request Date	August 30, 2021
Report Date	September 02, 2021

Risk profile:

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, **Scorpion Finance's smart contract source code has Low Risk Severity.**

For the detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit. At the time of the audit, the contract is not deployed on any blockchain. Note, the owner/developer can change/modify contract before blockchain deployment. Please proceed with caution.

Table of contents

Project Overview

InterFi was consulted by Scorpion Finance on August 30, 2021 to conduct a smart contract security audit of their solidity source code.

Public information

Scorpion finance is one of the newest and forward-looking innovation projects in cryptocurrency platforms. Scorpion finance aims to interconnect Blockchain services like Defi, NFT, Gaming, Payment and Marketplace under one ecosystem. Longevity is one of the driving forces of Scorpion Finance. Come on board and earn a long-term passive income through our innovations. Like dual rewards up to 10% BNB and 5% in scorpfin token.

Information	Scorpion Finance
Blockchain	No deployment info at the time of audit
Language	Solidity
Contract	https://github.com/interfinetwork/audited-codes/blob/main/ScorpFin.sol
Website	https://www.scorpion-finance.com/
Twitter	https://twitter.com/ScorpionFinance
Telegram	https://t.me/ScorpFin
Reddit	https://www.reddit.com/user/ScorpionFinance
Facebook	https://www.facebook.com/ScorpFin
Instagram	https://www.instagram.com/ScorpionFinance

Public logo

Solidity Source Code

https://github.com/interfinetwork/audited-codes/blob/main/ScorpFin.sol

GitHub Commits

Solidity source code committed at: 4c54d7ca9a2982ffdd9c5d27c76c1082d3c20e6c

Audit Scope & Methodology

The scope of this report is to audit the smart contract source code of Scorpion Finance. The source code can be viewed in its entirety on

https://github.com/interfinetwork/audited-codes/blob/main/ScorpFin.sol

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 ❖ Code Typo Error ❖ Token Supply Manipulation
Functional Assessment	❖ Access Control and Authorization ❖ Operations Trail and Event Generation ❖ Assets Manipulation ❖ Liquidity Access

ECHELON-1 Analysis

The aim of "InterFi's ECHELON-1 Analysis" 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. Code review that includes the following
   - 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. Testing and automated analysis that includes the following
   - 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
- MythX
- Consensys Mythril
- Open Zeppelin
- Solidity Code Complier

InterFi's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in Ether. 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 Overview

Knick-knacks in the smart contract

Query	Result
EcosystemandCommunityWallet	0x4fC5B79C759eB5D6E4FA92eD3d749efc27FFD77A
MarketingandDevelopmentWallet	0x6a66368a59618A658aa90B36ff35E6605a96378B
RandDWallet	0xe70BcFc64da9a64401462D8fA6682a1F745cF89C
FoundersandTeamWallet	0x2B0C1f305e7363A95229f0890EA95e4C6c0E137D
StakingandFarmingWallet	0x7d622428EB8604abC18dF0eDDAf953eD041493B4
maxFeeRate	25
totalFees	15
owner	0xf6EB7252A388b5f1FC304a103faf8dD35D09fB2B
uniswapV2Router	0x10ED43C718714eb63d5aA57B78B54704E256024E
swapAndLiquifyEnabled	True
symbol	scorpfin
Name	Scorpion Finance
totalSupply	100000000000

Verifying token functions

Function	Description	Tested	Verdict
TotalSupply	provides information about the total token supply	**Yes**	**Passed**
BalanceOf	provides account balance of the owner's account	**Yes**	**Passed**
Transfer	executes transfers of a specified number of tokens to a specified address	**Yes**	**Passed**
TransferFrom	executes transfers of a specified number of tokens from a specified address	**Yes**	**Passed**
Approve	allow a spender to withdraw a set number of tokens from a specified account	**Yes**	**Passed**
Allowance	returns a set number of tokens from a spender to the owner	**Yes**	**Passed**

Verified

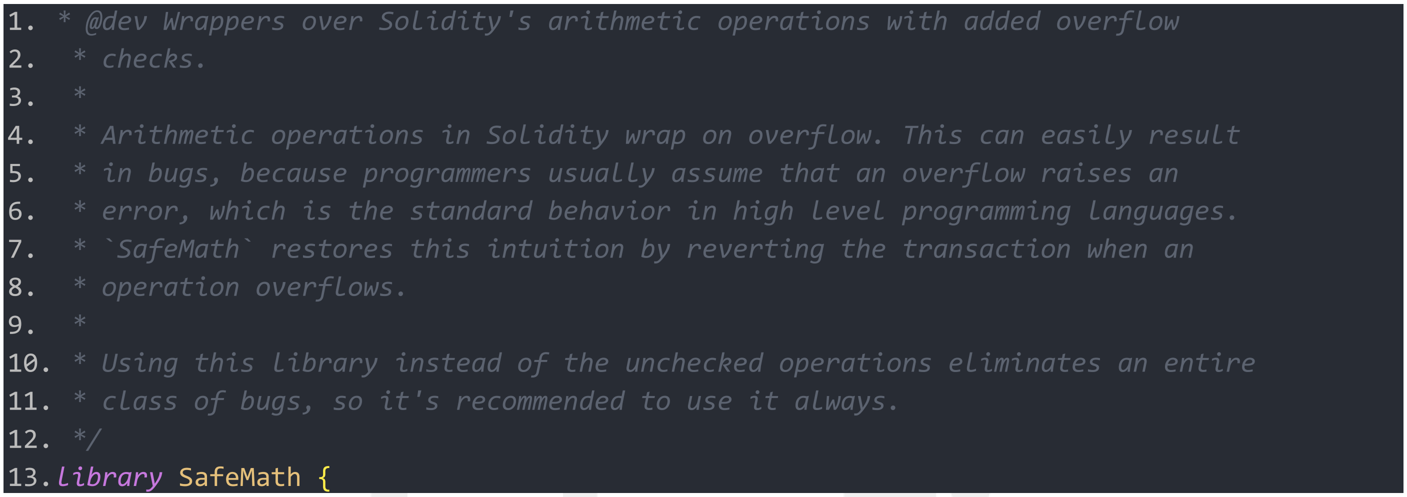
- Owner can mint tokens one-time
- Owner can not burn/lock users' assets
- Owner can not pause the contract

Note

- Active Owner: 0xf6EB7252A388b5f1FC304a103faf8dD35D09fB2B
- Owner can change transaction tax, allowances, etc.
- At the time of the audit, the contract is not deployed on any blockchain. Note, the owner/developer can change/modify contract before blockchain deployment.

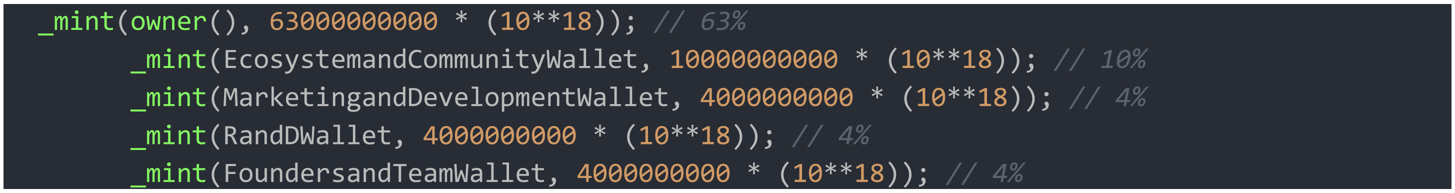
Points To Note

The smart contract utilizes the “SafeMath” to prevent Integer Overflow.



```
 * @dev Wrappers over Solidity's arithmetic operations with added overflow
 * checks.
 *
 * Arithmetic operations in Solidity wrap on overflow. This can easily result
 * in bugs, because programmers usually assume that an overflow raises an
 * error, which is the standard behavior in high level programming languages.
 * `SafeMath` restores this intuition by reverting the transaction when an
 * operation overflows.
 *
 * Using this library instead of the unchecked operations eliminates an entire
 * class of bugs, so it's recommended to use it always.
 */
library SafeMath {
```

The smart contract uses the “Mint” function



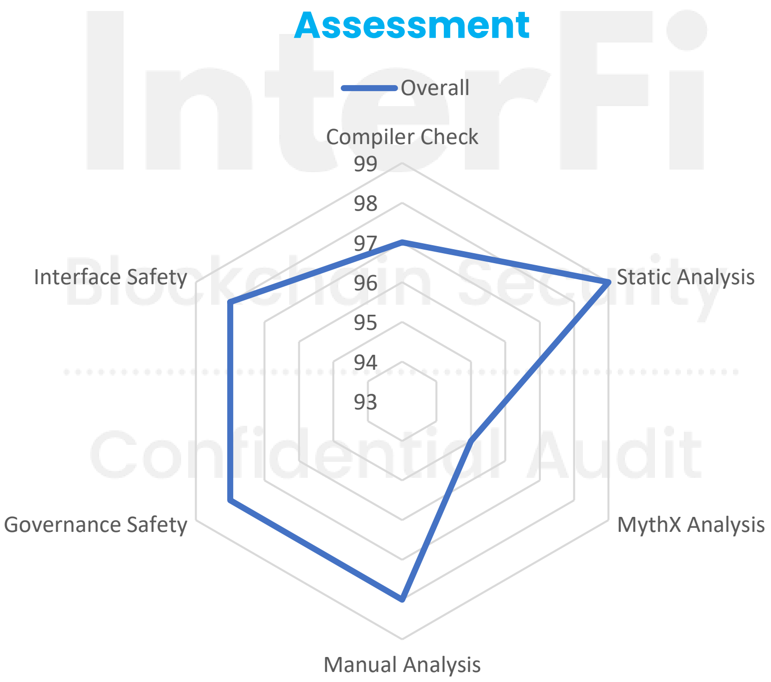
```
  _mint(owner(), 63000000000 * (10**18)); // 63%
        _mint(EcosystemandCommunityWallet, 10000000000 * (10**18)); // 10%
        _mint(MarketingandDevelopmentWallet, 4000000000 * (10**18)); // 4%
        _mint(RandDWallet, 4000000000 * (10**18)); // 4%
        _mint(FoundersandTeamWallet, 4000000000 * (10**18)); // 4%
```

Vulnerability	Status
Compiler errors	**! Medium**
Re-entrancy. Race conditions and cross function race conditions (RE)	**Passed**
Possible delays in data delivery	**Passed**
Gas optimization	**Passed**
Integer Underflow and overflow	**Passed**
Oracle Calls	**Passed**
Call stack depth attack	**Passed**
Parity Multisig Bug	**Passed**
Tx ordering dependency (TO)	**Passed**
DOS with revert and block gas limit	**Passed**
Private user data leaks	**Passed**
Malicious event log	**Passed**
Safe open zeppelin contract implementation and usage	**Passed**
The impact of exchange rate on the logic	**Passed**
Functions that are not used (dead-code)	**! Low**
Typographical Errors	**! Low**
Signature Malleability	**Passed**
Floating Pragma	**Passed**
Scoping and declarations	**Passed**

Smart Contract Risk Assessment

SWC Errors	Issue	Severity
SWC-110	**Out of bounds array access** The index access expression can cause an exception in case of use of invalid array index value.	! Low
SWC-115	**Use of "tx.origin" as a part of authorization control.** The tx.origin environment variable has been found to influence a control flow decision. Note that using "tx.origin" as a security control might cause a situation where a user inadvertently authorizes a smart contract to perform an action on their behalf. It is recommended to use "msg.sender" instead.	! Low
SWC-120	**Potential use of "block.number" as source of randonmness.** The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.	! Low

Risk Severity	Status
! Critical	None critical severity issues identified
! High	None high severity issues identified
! Medium	None medium severity issues identified
! Low	3 Low severity issues identified **(! Low Impact)**



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.

Scorpion Finance's smart contract source code has LOW RISK SEVERITY.

Scorpion Finance has PASSED the InterFi's ECHELON-1 standard smart contract audit.

Auditor's Footnote:

- At the time of the audit, the contract is not deployed on any blockchain. Note, the owner/developer can change/modify contract before deployment. Please proceed with caution.
- Project website is not checked due to out of scope. The website hasn't been reviewed for SSL and lighthouse report.
- Project team, and the project's social channels are not checked due to out of scope.

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.**

For more information, visit https://interfi.network

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