InterFi Network

@interfinetwork

SMART CONTRACT SECURITY AUDIT

Slush Finance

AUDITED ON AUGUST 31, 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 28, 2021
Report Date	August 31, 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, **Slush 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.

Table of contents

Project overview

InterFi was consulted by Slush Finance on August 28, 2021 to conduct a smart contract security audit. The contract under scope is:

https://bscscan.com/token/0x777e0cefc197edb88c986e5328088fe52d1c4a55

Public information

Slush.Finance ($SLUSH) is an autonomous Bitcoin Yield Farming protocol on the Binance Smart Chain that allows holders of Slush Puppie token to farm BTCB without staking on any 3rd Party Platform. Holders will receive hourly airdrops of BTCB direct to their wallet just for holding SLUSH as 8% tax is collected from every BUY and SELL and split proportionately between holders. As the volume increases on SLUSH so will the rewards meaning the longer you hold SLUSH the more Bitcoin you will passively earn.

Information	Slush Finance
Blockchain	Binance Smart Chain
Language	Solidity
Contract	0x777e0cefc197edb88c986e5328088fe52d1c4a55
Website	https://slush.finance/en/
Twitter	https://twitter.com/SlushFinance
Telegram	https://t.me/SlushFinance
GitHub	https://github.com/FinanceSlush

Public logo

Deployed smart contract

https://bscscan.com/token/0x777e0cefc197edb88c986e5328088fe52d1c4a55

Slush Finance contract source code is verified: Solidity v0.6.12+commit.27d51765

Audit scope and methodology

The scope of this report is to audit the smart contract source code of Slush Finance. The source code can be viewed in its entirety on https://bscscan.com/address/0x777e0cefc197edb88c986e5328088fe52d1c4a55#code#F1#L1

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

InterFi methodology

The aim of this report is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by InterFi 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 vulnerabilities

- ❖ Slither
- ❖ MythX
- ❖ Uniswap V2
- ❖ Open Zeppelin
- ❖ Solidity Code Complier

General risk factors

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 Ethereum's 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

Audit overview

Functions used in the smart contract

Query	Result
allowance	True
_marketingWalletAddress	0x8bb5f40a7e2a938e953dc40f93b04adf0133fa5d
dividendTracker	0x2fc1abf2f5d738e9de7c229dd24e06000549b7f9
gasForProcessing	300000
getClaimWait	3600
getLastProcessedIndex	389
decimals	18
getNumberOfDividendTokenHolders	401
getTotalDividendsDistributed	944451126686726420
liquidityFee	2
name	SLUSH PUPPIE
marketingFee	5
owner	0xc8bf5b215c73538c8bfb3e79005bc6f60716bdcc
swapTokensAmount	200000000000000000000000
BTCB	0x7130d2a12b9bcbfae4f2634d864a1ee1ce3ead9c
symbol	SLUSH
BTCBRewardsFee	8
totalFees	15
totalSupply	100000000000000000000000000000
uniswapV2Pair	0x169cf1a911e69800238230e908d4d489eff11ab4
uniswapV2Router	0x10ed43c718714eb63d5aa57b78b54704e256024e

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	**NA**	**Passed**
Transfer	executes transfers of a specified number of tokens to a specified address	**NA**	**Passed**
TransferFrom	executes transfers of a specified number of tokens from a specified address	**NA**	**Passed**
Approve	allow a spender to withdraw a set number of tokens from a specified account	**NA**	**Passed**
Allowance	returns a set number of tokens from a spender to the owner	**Yes**	**Passed**

Verified

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

Note

- Active Owner: 0xc8bf5b215c73538c8bfb3e79005bc6f60716bdcc
- Owner can change transaction tax, allowances, etc.
- Owner can modify BTCB dividend functions

Points To Note

1. The smart contract utilizes the SafeMath to prevent Integer Overflow.

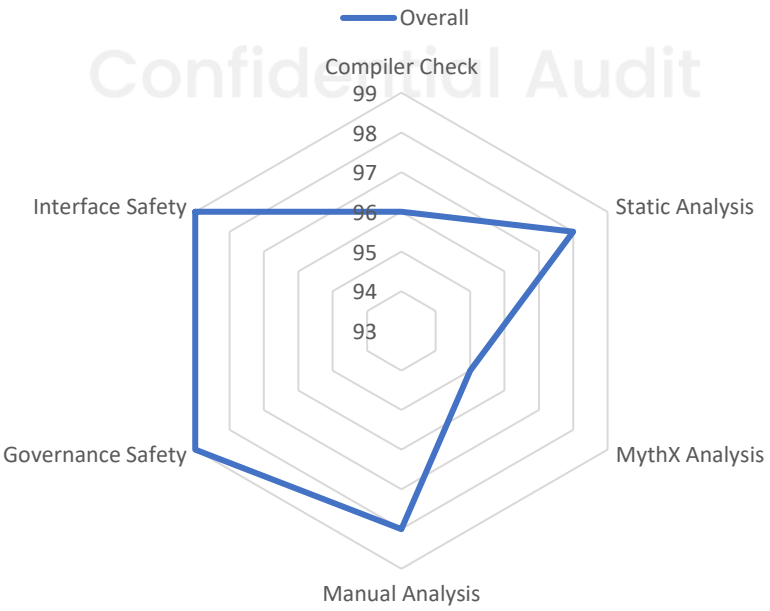
```
ftrace | funcSig
function add(uint256 a1, uint256 b1) internal pure returns (uint256) {
    uint256 c = a1 + b1;
    require(c >= a1, "SafeMath: addition overflow");

    return c;
}
```

2. The source code has some medium-risk severity issues. No major impactful event identified.
3. The source code has a low-risk severity. No major impactful event identified.

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)	**Passed**
Typographical Errors	**Passed**
Signature Malleability	**Passed**
Floating Pragma	**Passed**
Scoping and declarations	**Passed**

Risk Severity	Status
! Critical	None critical severity issues identified
! High	None high severity issues identified
! Medium	Medium severity issues identified KeccakCaching, EmptyByteArrayCopy, DynamicArrayCleanup ***(Low Impact)***
! Low	Low severity issues identified *Contract used outdated solidity compiler at the time of deployment* ***(No Impact)***

Assessment



Conclusion

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.

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

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

Auditor's Footnote:

- Liquidity pair contract's security is not checked due to out of scope. Liquidity locking details NOT provided by the team.
- 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. 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