Code Security Assessment

FST SWAP

Jan 16th, 2022

Table of Contents

Summary

This report has been prepared for FST SWAP to discover issues and vulnerabilities in the source code of the FST SWAP project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding 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, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	FST SWAP
Platform	BSC
Language	Solidity
Codebase	https://bscscan.com/address/0xC9882dEF23bc42D53895b8361D0b1EDC7570Bc6A https://bscscan.com/address/0x9A272d734c5a0d7d84E0a892e891a553e8066dce https://bscscan.com/address/0x1B6C9c20693afDE803B27F8782156c0f892ABC2d
Commit	

Audit Summary

Delivery Date	Jan 16, 2022
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	2	0	0	1	0	1
Medium	0	0	0	0	0	0
Minor	2	0	0	2	0	0
Informational	5	0	0	5	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
FSC	FistStandard.sol	92154da3d4bfbecd487725ca34491bed3556e210b8686e9a2b3fd19ee817817d
FFC	FstswapFactory.sol	c5788a11c532fc919cc935795b358172d615ccff63a6db48be62d7a2ff4032c7
FRC	FstswapRouter02.sol	a3b1db17fbf769d76efeeadf9e3abe67503920408dbf8f56e1a9ccf5c7854710

Understandings

Overview

FSTSwap is an automated liquidity protocol powered by a constant product formula. Each smart contract, or pair, manages a liquidity pool made up of reserves of two ERC-20 tokens. Anyone can become a liquidity provider (LP) for a pool by depositing an equivalent value of each underlying token in return for pool tokens. These tokens track pro-rata LP shares of the total reserves, and can be redeemed for the underlying assets at any time.

Pairs act as automated market makers, standing ready to accept one token for the other as long as the `constant product` formula is preserved. This formula, most simply expressed as x * y = k(In practice, contract applies a 0.30% fee to trades), states that trades must not change the product (k) of a pair’ s reserve balances (x and y).When adding or removing liquidity and if `fee` is on, mint liquidity equivalent to 1/4th of the growth in sqrt(k).

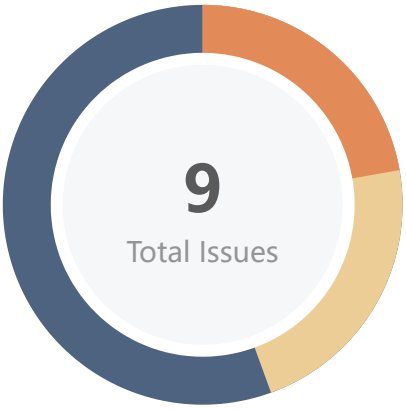
Privileged Functions

The contract contains the following privileged functions that are used to modify the contract configurations and address attributes. We list these functions below:

Contract `factory`:

- setFeeTo(address _feeTo)
- setFeeToSetter(address _feeToSetter)

Findings



Critical	**0** (0.00%)
Major	**2** (22.22%)
Medium	**0** (0.00%)
Minor	**2** (22.22%)
Informational	**5** (55.56%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
FFC-01	Divide by Zero	Logical Issue	Minor	Acknowledged
FFC-02	Missing Input Validation	Logical Issue	Informational	Acknowledged
FFC-03	Centralization Related Risks	**Centralization / Privilege**	**Major**	Resolved
FFC-04	Unnecessary Array as Counter	Gas Optimization	Informational	Acknowledged
FFC-05	Proper Usage Of `require()` And `assert()`	Coding Style	Informational	Acknowledged
FRC-01	Incompatibility With Deflationary Tokens	Logical Issue	Minor	Acknowledged
FSC-01	Too Many Digits	Coding Style	Informational	Acknowledged
FSC-02	Function Visibility Optimization	Gas Optimization	Informational	Acknowledged
FSC-03	Token Minted To Centralized Address	**Centralization / Privilege**	**Major**	Acknowledged

FFC-01 | Divide by Zero

Category	Severity	Location	Status
Logical Issue	Minor	FstswapFactory.sol: 547	Acknowledged

Description

The call to `burn()` function will fail if the value of `totalSupply` is 0.

Recommendation

We advise the client to add the following validation in the function `burn()`:

```
547  require(totalSupply != 0, "The value of totalSupply must not be 0");
```

Alleviation

The client has already acknowledged.

FFC-02 | Missing Input Validation

Category	Severity	Location	Status
Logical Issue	Informational	FstswapFactory.sol: 762	Acknowledged

Description

The given input is missing the check for the non-zero value.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error as below:

```
762    function uqdiv(uint224 x, uint112 y) internal pure returns (uint224 z) {
763          require(y !=0, "y can not be 0!");
764          z = x / uint224(y);
765      }
```

Alleviation

The client has already acknowledged.

FFC-03 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**	FstswapFactory.sol: 699, 704	Resolved

Description

In the contract `FstswapFactory`, the role `feeToSetter` has authority over the following functions:

- function setFeeTo()
- function setFeeToSetter()

Any compromise to the `feeToSetter` account may allow a hacker to take advantage of this authority.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination mitigate by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, mitigate by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered fully resolved.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

The client has renounced the ownership of `FistStandard` contract, the ownership of `FistStandard` contract is address(0).

FFC-04 | Unnecessary Array as Counter

Category	Severity	Location	Status
Gas Optimization	Informational	FstswapFactory.sol: 660	Acknowledged

Description

The usage of `allPairs` array is as a counter to maintain the number of created pairs.

Recommendation

We advise the client to replace the `allPairs` with a simple uint type counter to store the number of pairs created.

Alleviation

The client has already acknowledged.

FFC-05 | Proper Usage Of `require()` And `assert()`

Category	Severity	Location	Status
Coding Style	Informational	FstswapRouter02.sol: 411, 452, 526, 821, 910, 992	Acknowledged

Description

The `assert()` function should only be used to test for internal errors, and to check invariants. The `require()` function should be used to ensure valid conditions, such as inputs, or contract state variables are met, or to validate return values from calls to external contracts.

Recommendation

We advise the client using the `require()` function, along with a custom error message when the condition fails, instead of the `assert()` function

Alleviation

The client has already acknowledged.

FRC-01 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Logical Issue	Minor	FstswapRouter02.sol: 491, 492, 551, 583	Acknowledged

Description

When users add or remove LP tokens into the router, and the `mint` and `burn` operations are performed. When transferring standard ERC20 deflationary tokens, the input amount may not be equal to the received amount due to the charged transaction fee. As a result, the amount inconsistency will occur and the transaction may fail due to the validation checks.

Recommendation

We advise the client to regulate the set of LP tokens supported and add necessary mitigation mechanisms to keep track of accurate balances if there is a need to support deflationary tokens.

Alleviation

The client response:
FistSwap will not be specially compatible with deflationary tokens.

FSC-01 | Too Many Digits

Category	Severity	Location	Status
Coding Style	Informational	FistStandard.sol: 359	Acknowledged

Description

Literals with many digits are difficult to read and review.

Recommendation

We advise the client to use the scientific notation to improve readability. For example:

```
359    _totalSupply = 2 * 10**8 * 10**6;
```

Alleviation

The client has already acknowledged.

FSC-02 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	Informational	FistStandard.sol: 469, 488	Acknowledged

Description

The following functions are declared as `public`, contain array function arguments, and are not invoked in any of the contracts contained within the project's scope. The functions that are never called internally within the contract should have external visibility.

Recommendation

We advise that the functions' visibility specifiers are set to `external` and the array-based arguments change their data location from `memory` to `calldata`, optimizing the gas cost of the function.

Alleviation

The client has already acknowledged.

FSC-03 | Token Minted To Centralized Address

Category	Severity	Location	Status
Centralization / Privilege	**Major**	FistStandard.sol: 360	Acknowledged

Description

The amount of `_totalSupply` tokens that are minted to the centralized address `msg.sender` who is `owner`, may raise the community's concerns about the centralization issue.

Recommendation

We advise the client to carefully manage the `owner` account's private key and avoid any potential risks of being hacked. We also advise the client to adopt Multisig, Timelock, and/or DAO in the project to manage this specific account in this case.

Alleviation

The client has already acknowledged.

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 mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

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 specified commit.

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

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under 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 copies be delivered to any other person other than the Company, without CertiK's prior written consent in 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 CertiK 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, business model 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 should be 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and 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 CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but 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 tokens are 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 PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES 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, CERTIK 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 HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK 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 STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK' S AGENTS 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. CERTIK 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, RESULTING FROM 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 THE THIRD-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 PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK' S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

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

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts 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 our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.