CERTIK

Security Assessment

Swell II

Jun 6th, 2022

Table of Contents

Summary

This report has been prepared for Swell Ⅱ to discover issues and vulnerabilities in the source code of the Swell Ⅱ 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	Swell II
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/SwellNetwork/vault/commits/main/contracts
Commit	db212449126d46da68c8b49d03980382b6d86b36

Audit Summary

Delivery Date	Jun 06, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

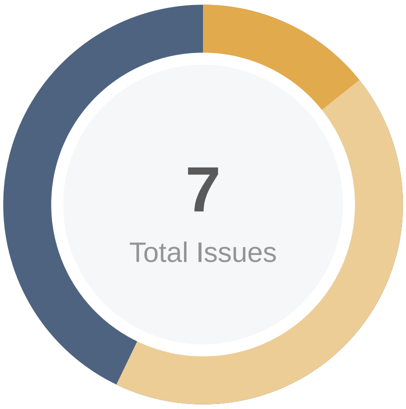
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
Critical	0	0	0	0	0	0	0
Major	0	0	0	0	0	0	0
Medium	1	0	0	0	0	0	1
Minor	3	0	0	3	0	0	0
Informational	3	0	0	1	0	0	2
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
SBV	implementations/SwellBalancerVault.sol	e3b3bad26968e1a01e2d2faf67e17ba4043223cd6fd1d3be8ff3fa36fe8c39b1

Findings



Critical	**0** (0.00%)
Major	**0** (0.00%)
Medium	**1** (14.29%)
Minor	**3** (42.86%)
Informational	**3** (42.86%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
SBV-01	Incorrect Array Length	Logical Issue	Medium	Resolved
SBV-02	Unknown Implementations	Volatile Code	Minor	Acknowledged
SBV-03	Potential Incorrect Index Of The Asset Token	Logical Issue	Minor	Acknowledged
SBV-04	Different Interfaces With `rari-capital`	Logical Issue	Minor	Acknowledged
SBV-05	Unlocked Compiler Version	Language Specific	Informational	Resolved
SBV-06	Missing Zero Address Validation	Coding Style	Informational	Resolved
SBV-07	Mathematical Verification	Logical Issue	Informational	Acknowledged

SBV-01 | Incorrect Array Length

Category	Severity	Location	Status
Logical Issue	Medium	implementations/SwellBalancerVault.sol: 121~122	Resolved

Description

The lengths of the array `tokens` and `tokenAmounts` are incorrect, they should be `tokensFromPool.length`.

```
IAsset[] memory tokens = new IAsset[](2);
uint256[] memory tokenAmounts = new uint256[](2);
```

Recommendation

We recommend reviewing the logic again and fix the length of the array.

Alleviation

The team heeded our advice and resolved this issue in commit `06c89f2c2ed18b93b6f0eb4cf0f183166af8877a`.

SBV-02 | Unknown Implementations

Category	Severity	Location	Status
Volatile Code	Minor	implementations/SwellBalancerVault.sol	Acknowledged

Description

The project is using the underlying entities `IVault`, `IWeightedPool`, and inherits the abstract contracts `ERC4626` and `WeightedMath` and uses util lib `FixedPointMathLib`.

However, the real implementations of these interfaces and abstract contracts and libs are out of the scope of this audit.

The scope of the audit treats these entities as black boxes and assumes their functional correctness. The related codes are not in the scope of this audit.

Recommendation

We understand that the business logic of this protocol requires interaction with `IVault`, `IWeightedPool`, `ERC4626`, `FixedPointMathLib`, and `WeightedMath`. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

The team acknowledged this issue and they will leave it as it is for now.

SBV-03 | Potential Incorrect Index Of The Asset Token

Category	Severity	Location	Status
Logical Issue	Minor	implementations/SwellBalancerVault.sol: 53~59, 89~96	Acknowledged

Description

Even if the condition `if (address(tokens[i]) == address(asset))` is not satisfied, the variable `assetIndex`/`exitTokenIndex` still can get the default value of 0 index for the asset token.

Recommendation

We recommend reviewing the logic again and ensuring it is intended.

Alleviation

The team acknowledged this issue and they stated it is impossible to deposit tokens that do not belong to a pool in the balancer protocol.

SBV-04 | Different Interfaces With `rari-capital`

Category	Severity	Location	Status
Logical Issue	Minor	implementations/SwellBalancerVault.sol	Acknowledged

Description

The definition of functions `beforeWithdraw()` and `afterDeposit()` are different from interfaces of `rari-capital` with an additional parameter in commit `06c89f2c2ed18b93b6f0eb4cf0f183166af8877a`, and the parameter is not used.

Recommendation

We would like to confirm with the client if the current implementation aligns with the original project design.

We recommend reviewing the functions again and ensuring it is intended.

Alleviation

The team acknowledged this issue and they stated the additional function parameters are there because the UniV3 vault uses them.

SBV-05 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	implementations/SwellBalancerVault.sol: 2	Resolved

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.8.0` the contract should contain the following line:

```
pragma solidity 0.8.0;
```

Alleviation

The team heeded our advice and resolved this issue in commit `06c89f2c2ed18b93b6f0eb4cf0f183166af8877a`.

SBV-06 | Missing Zero Address Validation

Category	Severity	Location	Status
Coding Style	Informational	implementations/SwellBalancerVault.sol: 25	Resolved

Description

Address should be checked before assignment to make sure it is not zero addresses.

Recommendation

Consider adding a zero check.

Alleviation

The team heeded our advice and resolved this issue in commit `06c89f2c2ed18b93b6f0eb4cf0f183166af8877a`.

SBV-07 | Mathematical Verification

Category	Severity	Location	Status
Logical Issue	Informational	implementations/SwellBalancerVault.sol: 61~67	Acknowledged

Description

The protocol is using some algorithms, including the logic of the function `_calcTokenOutGivenExactBptIn()`. The mathematical verification of these algorithms is not in the scope of this audit. The function logic will be checked based on the requirement documents.

Recommendation

Alleviation

The team acknowledged this issue and they will leave it as it is for now.

Appendix

Finding Categories

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.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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.