CERTIK

Security Assessment

Venus - VAI Controller Upgrade

CertiK Assessed on Apr 26th, 2024

CertiK Assessed on Apr 26th, 2024

Venus - VAI Controller Upgrade

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Binance Smart Chain (BSC)	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Solidity	Delivered on 04/26/2024	N/A

CODEBASE
https://github.com/VenusProtocol/venus-protocol/
View All in Codebase Page

COMMITS
base: a5569976c6b88c2fb82f9a9c5343817144b558b4
update 1: 8cb3def9cf4d44f9956f5f2ea98add98bcedf925
View All in Codebase Page

Vulnerability Summary

6	5	0	0	1	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined

0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
0	Major		Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
2	Minor	2 Resolved	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
4	Informational	3 Resolved, 1 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | VENUS - VAI CONTROLLER UPGRADE

CODEBASE | VENUS - VAI CONTROLLER UPGRADE

Repository

https://github.com/VenusProtocol/venus-protocol/

Commit

base: a5569976c6b88c2fb82f9a9c5343817144b558b4
update 1: 8cb3def9cf4d44f9956f5f2ea98add98bcedf925

AUDIT SCOPE | VENUS - VAI CONTROLLER UPGRADE

1 file audited • 1 file without findings

ID	Repo	File	SHA256 Checksum
VAI	VenusProtocol/venus-protocol	VAIController.sol	30b54e98ed6805b840db922135caec6bf4a5ffa86b4d56929ea8e19d1c81e325

APPROACH & METHODS | VENUS - VAI CONTROLLER UPGRADE

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

- Testing the smart contracts against both common and uncommon attack vectors;
- 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.

REVIEW NOTES | VENUS - VAI CONTROLLER UPGRADE

The scope of this audit is changes within PR-467 until commit a5569976c6b88c2fb82f9a9c5343817144b558b4 within the in-scope files.

The changes within the scope of the audit represent an upgrade made to VAI Unitroller contract 0x004065D34C6b18cE4370ced1CeBDE94865DbFAFE. The current implementation before upgrade can be found at 0x9817823d5c4023efb6173099928f17bb77cd1d69.

The in-scope portions of this PR make the following changes:

- Add the ability for users to repay the VAI debt of another user. This is done by implementing a function `repayVAIBehalf()`, which utilizes the same logic as `repayVAI()` except the borrower is specified as an input rather than assumed to be the `msg.sender`.
- Change the return value of `repayVAIFresh()` to be the total amount repaid as opposed to the amount that is burned. This change is implemented to fix a bug in the liquidation logic. In particular, the function `liquidateVAIFresh()` uses the return value of `repayVAIFresh()` to determine the amount of collateral the liquidator receives. However, this previously only returned the burned amount of VAI and did not account for the amount of VAI the liquidator pays for interest. As a result, the value of the collateral the liquidator received can be less than the amount they repay, making the liquidation unprofitable.
- Cosmetic Changes
    - Use `uint256` as opposed to `uint`
    - Utilize `add_`, `sub_`, `div_`, `mult_` as opposed to `addUint`, `subUint`, `divUint`, `multUint` in certain functions. In particular, this reverts with a generic overflow/underflow error as opposed to returning an error code.
    - Remove `MintLocalVars` struct.
    - Update `mintVai()` and `repayVAIFresh()` updated to revert in certain cases as opposed to returning an error.
    - Add private functions `_ensureNotPaused()` and `_ensureNonzeroAmount()` to check if the protocol is paused or the amount is nonzero.

Note that this audit only considered the changes above and did not take into consideration any pre-existing code or Centralization Risks. We recommend all users carefully review any centralization risks.

FINDINGS | VENUS - VAI CONTROLLER UPGRADE

6	0	0	0	2	4
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Venus - VAI Controller Upgrade. Through this audit, we have uncovered 6 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
VAI-01	Return Value Of `repayVAI()` Is Changed	Logical Issue	Minor	Resolved
VAI-02	Missing Zero Address Check	Volatile Code	Minor	Resolved
VAI-03	Inconsistent Use Of `addUint` , `subUint` , `mulUint` , And `divUint`	Inconsistency	Informational	Acknowledged
VAI-04	`ensureNotPaused()` Is Not Used Consistently	Inconsistency	Informational	Resolved
VAI-05	Breaking Change In Error Handling Of Some Functions	Coding Style	Informational	Resolved
VAI-06	Suggested Changes To Natspec Comments	Coding Style	Informational	Resolved

VAI-01 | RETURN VALUE OF `repayVAI()` IS CHANGED

Category	Severity	Location	Status
Logical Issue	Minor	VAIController.sol (Base): 162~165	Resolved

Description

The function, `repayVAI()`, returns `repayVAIFresh()` which has been changed to return the `repaidAmount` as opposed to the `burn` value. While this change was implemented to fix an error in the logic of `liquidateVAIFresh()`, it changes the return value of an external-facing function. Consequently, a protocol that utilizes the return value of this function may have their logic broken by this change.

Recommendation

We recommend providing a sufficient amount of warning about this breaking change prior to implementint it. Alternatively, or while waiting for a sufficient amount of time to pass, we recommend considering adjusting the logic of `liquidateVAIFresh()` in a way that does not cause a breaking change. A potential solution could be to move the logic of `repayVAIFresh()` to another internal function, that returns both the `burn` and `repaidAmount`. Then `repayVAIFresh()` can call this and use the `burn` amount, while `liquidateVAIFresh()` can also reference the return and instead utilize the `repaidAmount`.

Alleviation

`[Venus, 04/22/2024]`: "We will provide sufficient amount of warning with respect to the change in behavior of this function."

VAI-02 | MISSING ZERO ADDRESS CHECK

Category	Severity	Location	Status
Volatile Code	Minor	VAIController.sol (Base): 176~178	Resolved

Description

Function `repayVAIBehalf()` does not ensure that the input address `borrower` is not `address(0)`.

Recommendation

We recommend including the check above.

Alleviation

`[CertiK, 04/22/2024]`: The client made changes resolving the finding in commit 233ffcc60c7eabc6d199546554c9b2e455780593.

VAI-03 INCONSISTENT USE OF `addUint` , `subUint` , `mulUint` , AND `divUint`

Category	Severity	Location	Status
Inconsistency	Informational	VAIController.sol (Base): 116, 131, 132, 136, 142, 143, 218~219, 225	Acknowledged

Description

In the functions `mintVAI()` and `repayVAIFresh()` , the use of functions `addUint` , `subUint` , `mulUint` , and `divUint` was replaced by functions `add_` , `sub_` , `mul_` and `div_` respectively. The main difference is that the previous set of functions returned an error which was either used to revert the transaction and return a specific string message describing where the calculation failed or emit a `Failure` event. The updated version will revert with a generic overflow, underflow, or divide by zero string error.

This is done only in the functions `mintVAI()` and `repayVAIFresh()` , with the remaining instances of `addUint` , `subUint` , `mulUint` , and `divUint` in the contract unchanged.

Recommendation

We recommend choosing a convention and remaining consistent with it throughout the contract. In addition, changing the calculation error handling can also cause issues if there are protocols designed to interact with those error cases in a specific manner.

Alleviation

`[Venus, 04/22/2024]` : "Issue acknowledged. We will fix the issue in the future, which will not be included in this audit engagement."

VAI-04 | `ensureNotPaused()` IS NOT USED CONSISTENTLY

Category	Severity	Location	Status
Inconsistency	Informational	VAIController.sol (Base): 243	Resolved

Description

The function `liquidateVAI()` makes the following check:

```
require(!comptroller.protocolPaused(), "protocol is paused");
```

as opposed to using `_ensureNotPaused()`.

Recommendation

We recommend using `_ensureNotPaused()` to be consistent.

Alleviation

`[CertiK, 04/22/2024]`: The client made changes resolving the finding in commit 59dc488bf226e2c3120c28fb951d3da4e694e3f0.

VAI-05 BREAKING CHANGE IN ERROR HANDLING OF SOME FUNCTIONS

Category	Severity	Location	Status
Coding Style	Informational	VAIController.sol (Base): 121~122, 138~139, 221~223	Resolved

Description

Functions `mintVAI()` and `repayVAIFresh()` change how certain errors are handled, moving from returning an error to reverting.

Protocols may be designed to interact and handle errors in a specific manner, causing issues with compatibility.

Recommendation

We recommend providing a sufficient amount of warning about this breaking change prior to it being implemented.

Alleviation

`[Venus, 04/22/2024]`: "We will provide sufficient amount of warning with respect to the change in behavior for these functions."

VAI-06 | SUGGESTED CHANGES TO NATSPEC COMMENTS

Category	Severity	Location	Status
Coding Style	Informational	VAIController.sol (Base): 160~161, 161~162, 170~171	Resolved

Description

- The comments above functions `repayVAI()` and `repayVAIBehalf()` contain the following statement: "The repay function transfers VAI into the protocol and burn, reducing the borrower's borrow balance." The statement may read better as "The repay function transfers VAI interest into the protocol and burns the rest, reducing the borrower's borrow balance."
- The following comment above function `repayVAI()` is also relevant to `repayVAIBehalf()`: "Before repaying an asset, users must first approve the VAI to access their VAI balance."

Recommendation

We recommend making changes to the cited comments above.

Alleviation

`[CertiK, 04/22/2024]`: The client made changes resolving the finding in commit 8cb3def9cf4d44f9956f5f2ea98add98bcedf925.

APPENDIX | VENUS - VAI CONTROLLER UPGRADE

Finding Categories

Categories	Description
Coding Style	Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable.
Inconsistency	Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.

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 | CERTIK

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.

CertiK | Securing the Web3 World

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.