



Security Assessment

Venus - Private Conversions

CertiK Assessed on Nov 27th, 2023





Certik Assessed on Nov 27th, 2023

Venus - Private Conversions

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

Executive Summary

TYPES

DeFi

ECOSYSTEM

Binance Smart Chain
(BSC)

METHODS

Manual Review, Static Analysis

LANGUAGE

Solidity

TIMELINE

Delivered on 11/27/2023

KEY COMPONENTS

N/A

CODEBASE

<https://github.com/VenusProtocol/protocol-reserve>

View All in Codebase Page

COMMITTS

base: [23d4e99719b57b939b75a731f48387b875722b5e](#)update: [72e1b37676587dbcb0e8ea1502081c9646d52f3b](#)update: [c8588a01c5cef91e887aaedcbefef1405a58f3e6](#)

View All in Codebase Page

Highlighted Centralization Risks

Contract upgradeability

Vulnerability Summary



11

Total Findings

8

Resolved

2

Mitigated

0

Partially Resolved

1

Acknowledged

0

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.

2 Major

2 Mitigated



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.

1 Medium

1 Resolved



Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.

3 Minor

3 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.

■ 5 Informational

4 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 - PRIVATE CONVERSIONS

I **Summary**

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

I **Summary**

I **Third Party Dependencies**

I **Findings**

[TCP-01 : Centralization Related Risks](#)

[VPB-01 : Centralized Control of Contract Upgrade](#)

[ATT-01 : Discussion on `setConversionConfig\(\)` Check](#)

[ATT-02 : Missing Input Validation](#)

[TCP-03 : Issue with deflationary tokens](#)

[TCP-06 : Future Consideration of Reentrancy](#)

[CNC-04 : No Upper Bound on `_loopsLimit`](#)

[CNC-05 : Discussion On Amount Of Converters With Configuration For Fixed `_tokenAddressIn` and `_tokenAddressOut`](#)

[IAC-01 : Recorded Information Meaning Will Change at the time of Upgrade](#)

[TCP-04 : Typos An Inconsistencies](#)

[TCP-05 : Missing And Incomplete NatSpec Comments](#)

I **Optimizations**

[CNC-02 : Unneeded Check](#)

I **Appendix**

I **Disclaimer**

CODEBASE | VENUS - PRIVATE CONVERSIONS

Repository

<https://github.com/VenusProtocol/protocol-reserve>

Commit








base: [23d4e99719b57b939b75a731f48387b875722b5e](#)

update: [72e1b37676587dbcb0e8ea1502081c9646d52f3b](#)

update: [c8588a01c5cef91e887aaedcbefef1405a58f3e6](#)

AUDIT SCOPE | VENUS - PRIVATE CONVERSIONS

7 files audited ● 1 file with Acknowledged findings ● 1 file with Mitigated findings ● 3 files with Resolved findings
● 2 files without findings

ID	Repo	File	SHA256 Checksum
● CNC	VenusProtocol/protocol-reserve	 TokenConverter/ConverterNetwork.sol	2e58331a9c7e76d2d245e7cc7a9b32837d8ee41380555cfb77e8368e00a6ef84
● ATT	VenusProtocol/protocol-reserve	 TokenConverter/AbstractTokenConverter.sol	a155fca1d55a7aa42dff1688c0056c45e22668562db3336163f42ddea93bb101
● IAC	VenusProtocol/protocol-reserve	 TokenConverter/IAbstractTokenConverter.sol	26638da25f59ee61951098cf705cfb49a16330b41864e9cf79d3baa5e91dbdf9
● RFT	VenusProtocol/protocol-reserve	 TokenConverter/RiskFundConverter.sol	3517c3f84eb66ba61de065c612775f856a9b75019679b4e2fe60196c5d1fcd4
● STT	VenusProtocol/protocol-reserve	 TokenConverter/SingleTokenConverter.sol	d76e0e6f4cb56e64d8daf15767a21666a3ed7b78a1a9946ffdecadc0c33c4e5a
● ICI	VenusProtocol/protocol-reserve	 Interfaces/IConverterNetwork.sol	1f184dd8aa04a14ecca3623397fbfd13637ac6a82168e57c9588560f3b80aaf1
● AHV	VenusProtocol/protocol-reserve	 Utils/ArrayHelpers.sol	251177de68eaf963064439689b0c0d1310ab9334165e4b42dd31e1ff326c9ee7

APPROACH & METHODS | VENUS - PRIVATE CONVERSIONS

This report has been prepared for Venus to discover issues and vulnerabilities in the source code of the Venus - Private Conversions 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.

SUMMARY | VENUS - PRIVATE CONVERSIONS

This audit concerns the changes made in files outlined in PR: <https://github.com/VenusProtocol/protocol-reserve/pull/35>, with the commit audited being [23d4e99719b57b939b75a731f48387b875722b5e](#).

Note that any centralization risks present in the existing codebase before this PR was not considered in this audit and only those added in this PR are addressed in the audit. We recommend all users to carefully review the centralization risks, much of which can be found in our previous audit *VENUS - TOKEN CONVERTER*, which can be found here: <https://skynet.certik.com/projects/venus>.

THIRD PARTY DEPENDENCIES | VENUS - PRIVATE CONVERSIONS

The protocol is serving as the underlying entity to interact with third party protocols. The third parties that the contracts interact with are:

- ERC20 Tokens
- Oracles

The scope of the audit treats third party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets. Moreover, updates to the state of a project contract that are dependent on the read of the state of external third party contracts may make the project vulnerable to read-only reentrancy. In addition, upgrades of third parties can possibly create severe impacts, such as increasing fees of third parties, migrating to new LP pools, etc.

FINDINGS | VENUS - PRIVATE CONVERSIONS



11

Total Findings

0

Critical

2

Major

1

Medium

3

Minor

5

Informational

This report has been prepared to discover issues and vulnerabilities for Venus - Private Conversions. Through this audit, we have uncovered 11 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
TCP-01	Centralization Related Risks	Centralization	Major	● Mitigated
VPB-01	Centralized Control Of Contract Upgrade	Centralization	Major	● Mitigated
ATT-01	Discussion On <code>setConversionConfig()</code> Check	Logical Issue	Medium	● Resolved
ATT-02	Missing Input Validation	Volatile Code	Minor	● Resolved
TCP-03	Issue With Deflationary Tokens	Logical Issue	Minor	● Resolved
TCP-06	Future Consideration Of Reentrancy	Concurrency	Minor	● Resolved
CNC-04	No Upper Bound On <code>_loopsLimit</code>	Volatile Code	Informational	● Resolved
CNC-05	Discussion On Amount Of Converters With Configuration For Fixed <code>_tokenAddressIn</code> And <code>_tokenAddressOut</code>	Logical Issue	Informational	● Acknowledged
IAC-01	Recorded Information Meaning Will Change At The Time Of Upgrade	Design Issue	Informational	● Resolved
TCP-04	Typos An Inconsistencies	Inconsistency	Informational	● Resolved

ID	Title	Category	Severity	Status
TCP-05	Missing And Incomplete NatSpec Comments	Inconsistency	Informational	● Resolved

TCP-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization	● Major	TokenConverter/AbstractTokenConverter.sol (base): 247, 259-260; TokenConverter/ConverterNetwork.sol (base): 56, 66	● Mitigated

Description

Note that any centralization risks present in the existing codebase before this PR was not considered in this audit and only those added in this PR are addressed in the audit. We recommend all users to carefully review the centralization risks, much of which can be found in our previous audit *VENUS - TOKEN CONVERTER*, which can be found here:

<https://skynet.certik.com/projects/venus>.

In the contract `AbstractTokenConverter` the role `_owner` was given authority over the functions:

- `setConverterNetwork()`
- `setConversionConfig()`

Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and do the following:

- Change the converter network to a malicious contract to be able to convert tokens when only converters should be allowed to or to cause private conversions to fail.
- Set or update conversion configurations and who is allowed to do the conversions.

In the contract `ConverterNetwork` the role `DEFAULT_ADMIN_ROLE` of the `AccessControlManager` can grant addresses the privilege to call the following functions:

- `addTokenConverter()`
- `removeTokenConverter()`

Any compromise to the `DEFAULT_ADMIN_ROLE` or accounts granted this privilege may allow a hacker to take advantage of this authority and add or remove token converters from the network. This can be used to either allow add a malicious converter or to exclude converters to prevent private conversions.

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., multisignature 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 ($\frac{2}{3}$, $\frac{3}{5}$) 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

[Venus, 11/22/2023] : "The owner of the contracts RiskFundConverter and SingleTokenConverter will be 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396, that is the Timelock contract used to execute the normal Venus Improvement Proposals (VIP).

For normal VIPs, the time config is: 24 hours voting + 48 hours delay before the execution.

So, only the Venus Community, via a VIP will be able to execute the mentioned protected functions.

We'll use the AccessControlManager (ACM) deployed at 0x4788629abc6cfca10f9f969efdeaa1cf70c23555.

In this ACM, only 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396 (Normal Timelock) has the DEFAULT_ADMIN_ROLE. And this contract is a Timelock contract used during the Venus Improvement Proposals.

The idea is to grant 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396 to execute every mentioned function. Moreover, we'll allow [a] (Fast-track) and [b] (Critical) also to execute the following functions:

ConverterNetwork.addTokenConverter() ConverterNetwork.removeTokenConverter()

The current config for the three Timelock contracts are:

normal: 24 hours voting + 48 hours delay fast-track: 24 hours voting + 6 hours delay critical: 6 hours voting + 1 hour delay

[a] 0x555ba73dB1b006F3f2C7dB7126d6e4343aDBce02

[b] 0x213c446ec11e45b15a6E29C1C1b402B8897f606d"

[Certik, 11/22/2023] : Considering these steps we have marked this finding as *mitigated*. While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it. We strongly recommend the team and community to constantly monitor these privileges.

VPB-01 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

Category	Severity	Location	Status
Centralization	● Major	TokenConverter/ConverterNetwork.sol (base): 45; ProtocolReserve/ProtocolShareReserve.sol (update_20231127): 23; ProtocolReserve/RiskFundStorage.sol (update_20231127): 11, 64; ProtocolReserve/XVSVaultTreasury.sol (update_20231127): 16; TokenConverter/AbstractTokenConverter.sol (update_20231127): 98	● Mitigated

Description

The contract `ConverterNetwork` is upgradeable; the corresponding `admin` role in each respective proxy has the authority to update the implementation contract behind each contract.

Any compromise to the `admin` account in each proxy may allow a hacker to take advantage of this authority and change the implementation contract the proxy points to, and therefore execute potential malicious functionality in the implementation contract.

Note that other contracts in scope are also upgradeable, but this functionality was not added in the PR which is in scope of this audit. For more details see our previous audit *VENUS - TOKEN CONVERTER*, which can be found here: <https://skynet.certik.com/projects/venus>.

Recommendation

We recommend that the team make efforts to restrict access to the admin of the proxy contract. A strategy of combining a time-lock and a multi-signature (2/3, 3/5) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to migrate to a new implementation contract.

Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently fully resolve the risk.

Short Term:

A combination of a time-lock and a multi signature (2/3, 3/5) wallet mitigate the risk by delaying the sensitive operation and avoiding a single point of key management failure.

- A time-lock with reasonable latency, such as 48 hours, for awareness of privileged operations;
AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
AND

- A medium/blog link for sharing the time-lock contract and multi-signers addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A combination of a time-lock on the contract upgrade operation and a DAO for controlling the upgrade operation mitigate the contract upgrade risk by applying transparency and decentralization.

- A time-lock with reasonable latency, such as 48 hours, for community awareness of privileged operations;
AND
- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
AND
- A medium/blog link for sharing the time-lock contract, multi-signers addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

Renouncing ownership of the `admin` account or removing the upgrade functionality can *fully* resolve the risk.

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

Alleviation

[Venus, 11/22/2023] : "The admin of the ConverterNetwork will be the ProxyAdmin contract deployed at 0x6beb6D2695B67FEb73ad4f172E8E2975497187e4.

The owner of this ProxyAdmin contract is 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396, the Normal Timelock used to execute the normal Venus Improvement Proposals (VIP).

For normal VIPs, the time configuration is: 24 hours voting + 48 hours delay before the execution.

So, these contracts will be upgraded only via a Normal VIP, involving the Venus Community/Governance in the process."

[Certik, 11/22/2023]: Considering these steps we have marked this finding as *mitigated*. While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it. We strongly recommend the team and community to constantly monitor these privileges.

ATT-01 | DISCUSSION ON `setConversionConfig()` CHECK

Category	Severity	Location	Status
Logical Issue	● Medium	TokenConverter/AbstractTokenConverter.sol (base): 269~272	● Resolved

Description

In the function `setConversionConfig()`, the following check is made:

```
if (
    ((conversionConfigurations[tokenAddressIn]
[tokenAddressOut].conversionAccess ==
    ConversionAccessibility.ONLY_FOR_CONVERTERS) ||
    (conversionConfigurations[tokenAddressIn]
[tokenAddressOut].conversionAccess ==
    ConversionAccessibility.ALL)) && (address(converterNetwork) ==
address(0))
) {
    revert InvalidConverterNetwork();
}
```

This check is performed on what is already recorded within `conversionConfigurations` mapping, rather than the input that is being used for `conversionConfig.conversionAccess`. This means that the function allows an update to this state, but that the configuration cannot be corrected unless the `address(converterNetwork)` is first updated to a nonzero address.

Can you provide more information on the intended purpose of this check? Is it structured in this way with consideration of the change from an `enabled` bool to a `conversionAccess` enum during upgrade, or should it have been an input validation check?

Recommendation

We recommend clarifying the purpose of this check or correcting it if it is meant to serve as input validation.

Alleviation

[Certik, 11/20/2023]: The client made changes resolving the finding in commit [abe7eadf62571ef4cd803f20c208390f6ff646c7](#).

ATT-02 | MISSING INPUT VALIDATION

Category	Severity	Location	Status
Volatile Code	Minor	TokenConverter/AbstractTokenConverter.sol (base): 262~263	Resolved

Description

In function `setConversionConfig()` of `AbstractTokenConverter`, the input `conversionConfig.incentive` can be nonzero even when the `conversionConfig.conversionAccess` is set to `ONLY_FOR_CONVERTERS` or `NONE`. While the incentive value will not be used in that case, it should be made zero to ensure intended updates are made.

- When set to `NONE`, the mapping `conversionConfigurations` corresponding entry can be deleted, instead of updating the value.
- If being set to `ONLY_FOR_CONVERTERS`, then incentive should be required to be 0.

Recommendation

We recommend ensuring updates to `conversionConfigurations` are consistent within function `setConversionConfig()`.

Alleviation

[Certik, 11/27/2023]: The client made changes fully resolving the finding in commit [c8588a01c5cef91e887aaedcbefef1405a58f3e6](#).

TCP-03 | ISSUE WITH DEFLATIONARY TOKENS

Category	Severity	Location	Status
Logical Issue	Minor	TokenConverter/AbstractTokenConverter.sol (base): 843~849; TokenConverter/RiskFundConverter.sol (base): 367	Resolved

Description

In the contract `RiskFundConverter`, the function `_postPrivateConversion()` makes the following call:

```
IRiskFund(destinationAddress).updatePoolState(comptroller, tokenAddressIn, convertedTokenInBalance);
```

However, the `convertedTokenInBalance` may not necessarily be the amount of tokens the `destinationAddress` received if it is a deflationary token. This is because it is only called via the function `AbstractTokenConverter._privateConversion()` which calculates the `convertedTokenInBalance` by incrementing by the `actualAmountOut`, which is the amount of tokens that are sent from the converter, not the amount that has been received by the destination address.

Recommendation

We recommend ensuring the `convertedTokenInBalance` represents the amount of tokens received by the destination address.

Alleviation

[Certik, 11/22/2023]: The client made changes resolving the finding in commit [f4c8d1e179c71a5cc10e9970ee45b4771e95c801](#).

TCP-06 | FUTURE CONSIDERATION OF REENTRANCY

Category	Severity	Location	Status
Concurrency	Minor	TokenConverter/AbstractTokenConverter.sol (base): 654~655, 825~831, 843~849; TokenConverter/ConverterNetwork.sol (base): 172~173, 180~181; TokenConverter/RiskFundConverter.sol (base): 226~227, 370~371	Resolved

Description

Venus has documented that they do not currently support tokens with hooks. This finding is created in consideration of future support for tokens with hooks or callback features.

Function `_findTokenConverters()` in contract `ConverterNetwork` reads the `balanceOf()` value of token converter contracts. In the case of the `RiskFundConverter` contract, this returns the value of `assetsReserves` for the input `tokenAddress`. It is important to note that this value is not updated until after transfers occur, in both `updateAssetsState()` and in any token converting functions it inherits from `AbstractTokenConverter`.

Consequently, this value may not accurately reflect the state of the `RiskFundConverter` contract in the order in which it is read, if tokens with hooks are supported within the protocol. Any logic depending upon this read value within `_findTokenConverters()` may consequently fail or act unexpectedly, preventing conversions between token converter contracts.

Recommendation

We recommend keeping this information in consideration, and, if tokens with hooks are to ever be supported, we recommend following the check-effect-interaction pattern in order to prevent any potential read-only reentrancy in this case.

Alleviation

[Venus, 11/22/2023]: "We don't support tokens with hooks now, but some of the underlying tokens are upgradable, so the behavior of these tokens in the future is unpredictable. We prefer to try to mitigate this risk."

We have made two changes:

1. Update `assetReserves` first, with a hook, following check-effect-interaction pattern
2. Avoid reentrancy in the `getPoolAssetReserve` view, because `poolsAssetsReserves` will not be updated when the token hook would be executed"

[Certik, 11/22/2023]: The client made changes in commits [73bc544ef7cbcdc7e017001ec5de9e2a3d654ecd](#) and [72e1b37676587dbcb0e8ea1502081c9646d52f3b](#).

All updates to `assetsReserves` are now made before transfers to unknown destinations. All updates and reads of the state of `poolsAssetsReserves` now include a reentrancy lock and, in the case of the function `getPoolAssetReserve()`, a lock check.

CNC-04 | NO UPPER BOUND ON `_loopsLimit`

Category	Severity	Location	Status
Volatile Code	● Informational	TokenConverter/ConverterNetwork.sol (base): 45~46, 208~209, 217~218	● Resolved

Description

The `_loopsLimit` and any update to it should be ensured to be less than `2**128 - 1`.

This is because `2**128 - 1` is the upper limit on the return value for `_findConverterIndex()`, and a return of this max value indicates that a tokenConverter does not exist in the array `allConverters`. Since the `_loopsLimit` is a `uint256` value, it may be set larger than this maximum value. This also ensures the casting of `allConverters.length` to a `uint128` value is safe.

Recommendation

We recommend ensuring upon update that `_loopsLimit` is set to a value strictly less than `2**128 - 1`.

Alleviation

[Certik, 11/22/2023]: The client made changes resolving the finding in commit [2ba750391140832648f374fe4d1e570173330942](#).

CNC-05 DISCUSSION ON AMOUNT OF CONVERTERS WITH CONFIGURATION FOR FIXED `_tokenAddressIn` AND `_tokenAddressOut`

Category	Severity	Location	Status
Logical Issue	● Informational	TokenConverter/ConverterNetwork.sol (base): 144~202	● Acknowledged

Description

We would like further clarification on the amount of converters that have a configuration for a fixed `_tokenAddressIn` and `_tokenAddressOut`.

In particular, each converter must have the configurations for `tokenAddressIn` to be the `destinationBaseAsset`. If there is only to be one `SingleTokenConverter` for each destination base asset, then there should only be one possible `SingleTokenConverter` that supports conversion with `_tokenAddressIn` and `_tokenAddressOut`.

Thus the only potential other converter would be the `RiskFundConverter` as the `convertibleBaseAsset` may be the destination base asset for one of the `SingleTokenConverter`. If there is only to be one `RiskFundConverter`, then with the assumptions above this leaves at most 2 converters that may support the configuration.

If these assumptions are correct, then the quick sort may be unnecessarily complex as there should only be 2 converters to sort.

Recommendation

We would like further clarification on the amount of converters that have a configuration for a fixed `_tokenAddressIn` and `_tokenAddressOut`.

Alleviation

[Venus, 11/22/2023] : "Assuming:

28 markets in the Core pool 19 different underlying tokens in the markets of the Isolated pools 1 underlying token in the Core pool and in the Isolated pools at the same time (USDT) Total: 46 different underlying tokens So, each converter will have 45 `ConversionConfig` instances, where the `tokenAddressIn` will be the destination base asset (see image), and the `tokenAddressOut` will be the rest of tokens supported by the protocol (excluding the destination base asset)."

Contract	Converter name	Destination base asset
SingleTokenConverter	XVSVaultConvert	XVS
SingleTokenConverter	USDCPrimeConverter	USDC
SingleTokenConverter	USDTPrimeConverter	USDT
SingleTokenConverter	BTCPrimeConverter	BTC
SingleTokenConverter	ETHPrimeConverter	ETH
RiskFundConverter	RiskFundConverter	USDT

[Certik, 11/22/2023] : From the information provided above it is confirmed that at most two token converter contracts at any given moment will be collecting the same base asset (e.g. `USDTPrimeConverter` and `RiskFundConverter` both collect USDT at the start). So for a provided combination of `_tokenAddressIn` and `_tokenAddressOut` within function `_findTokenConverters()`, even though each array `converters` and `convertersBalance` may start out with up to a length of six, there will only ever be as much as two converter addresses flagged for the given combination of token addresses. Since this is the case, it appears unnecessary to implement a quick sort algorithm, because the arrays passed to function `sort()` would have at most a length of two after they are revised with inline assembly to be the actual number of matching converters.

[Venus, 11/23/2023] : We'll keep the quicksort algorithm. The overhead and complexity are acceptable, in our opinion, and it will be ready for scenarios with more converters in the network to be sorted

IAC-01 | RECORDED INFORMATION MEANING WILL CHANGE AT THE TIME OF UPGRADE

Category	Severity	Location	Status
Design Issue	● Informational	TokenConverter/IAbstractTokenConverter.sol (base): 23~24	● Resolved

Description

Any configurations that are currently enabled will have their bool translate into an enum representation. That is, `enabled = false` will now translate to `conversionAccess` of `NONE`, and `enabled = true` should translate to `conversionAccess` of `ALL`.

Since the `converterNetwork` will correspond to `address(0)` until updated, this should mean that all previously enabled configurations continue to function as they did previously at the time of upgrade.

Recommendation

We recommend taking the above information into consideration during upgrades and ensuring that the consequences of translating from `enabled` to `ALL` are intended.

Alleviation

[Certik, 11/20/2023]: The client states they plan to deploy the token converters with support for private conversions independently, rather than upgrade previously deployed converters to the private conversion layout.

TCP-04 | TYPOS AN INCONSISTENCIES

Category	Severity	Location	Status
Inconsistency	● Informational	TokenConverter/AbstractTokenConverter.sol (base): 88~90, 19 9, 515, 696, 704; TokenConverter/ConverterNetwork.sol (base): 114	● Resolved

Description

AbstractTokenConverter

- In the comments at the start of the contract it has a comment for `findTokenConverter()`. However, the contract `ConverterNetwork` has two functions `findTokenConverters()` and `findTokenConvertersForConverters()`.
- In the comments above error `InsufficientPoolLiquidity` and within functions `getAmountOut()`, `_doTransferOut()`, the word "liquidity" is misspelled as "liquity."

ConverterNetwork

- The comments for the function `isTokenConverter()` state "This function checks for given address is converter or not" when it should be "This function checks if the given address is a converter or not".

Recommendation

We recommend fixing the typos and inconsistencies mentioned above.

Alleviation

[Certik, 11/20/2023]: The client made changes resolving the finding in commit [f8a8179d23d378262a4213708731d0594dfa591f](https://github.com/certiklabs/venus-private-conversions/commit/f8a8179d23d378262a4213708731d0594dfa591f).

TCP-05 | MISSING AND INCOMPLETE NATSPEC COMMENTS

Category	Severity	Location	Status
Inconsistency	● Informational	TokenConverter/AbstractTokenConverter.sol (base): 717, 791~793, 866~874, 990; TokenConverter/ConverterNetwork.sol (base): 42, 84~88, 96~100, 144~147, 204~206; TokenConverter/RiskFundConverter.sol (base): 93~100, 143~146, 267~272, 324~329, 358~364, 410; TokenConverter/SingleTokenConverter.sol (base): 36~38, 56~58, 71	● Resolved

Description

The `initialize()` functions do not consistently have a NatSpec comment such as `/// @notice ConverterNetwork initializer`.

AbstractTokenConverter

- The comments for `_updateAssetsState()` do not include the return value.
- The function `_doTransferIn()` is missing NatSpec comments.
- The comments for `_getAmountOut()` do not include the return value `tokenInToOutConversion`.
- The comments for `_getDestinationBaseAsset()` do not include the return value.

RiskFundConverter

- The comments for `getPoolAssetReserve()` do not include the error.
- The comments for `updatePoolAssetsReserve()` do not include the return value.
- The comments for `_updateAssetsState()` do not include the return value, do not include the error, and states it emits `AssetsReservesUpdated` in certain cases, which it does not.
- The function `_postPrivateConversion()` does not have any NatSpec comments.
- The comments for `_getDestinationBaseAsset()` do not include the return value.

SingleTokenConverter

- The comments for `_updateAssetsState()` do not include the return value.
- The comments for `_getDestinationBaseAsset()` do not include the return value.

ConverterNetwork

- The comments for `initialize()` do not include the parameter `_loopsLimit`.

- The comments for the functions `findTokenConverters()` and `findTokenConvertersForConverters()` do not indicate any difference between the functions.
- The comments for `_findTokenConverters()` do not include the parameter `forConverters` and does not indicate it returns two separate arrays.
- the comments for `_findConverterIndex()` do not include the return value.

Recommendation

We recommend adding or updating the NatSpec comments mentioned above.

Alleviation

[Certik, 11/20/2023] : The client made the recommended changes in commit [cdd9e14e670829ae70515ee7b5c6a39b20790314](#).

OPTIMIZATIONS | VENUS - PRIVATE CONVERSIONS

ID	Title	Category	Severity	Status
<u>CNC-02</u>	Unneeded Check	Gas Optimization	Optimization	● Resolved

CNC-02 | UNNEEDED CHECK

Category	Severity	Location	Status
Gas Optimization	● Optimization	TokenConverter/ConverterNetwork.sol (base): 67~68	● Resolved

Description

In function `removeTokenConverter()`, if a `_tokenConverter` was added successfully through `addTokenConverter()`, then the corresponding address was necessarily nonzero.

Recommendation

We recommend removing the unnecessary check.

Alleviation

[Certik, 11/20/2023]: The client made changes resolving the finding in commit [f52d56a08e482f63d204320354a60dff1024245e](#).

APPENDIX | VENUS - PRIVATE CONVERSIONS

Finding Categories

Categories	Description
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.
Concurrency	Concurrency findings are about issues that cause unexpected or unsafe interleaving of code executions.
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.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

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.

