SOLIDIFIED

Audit Report for B-Protocol - February 15, 2021

Summary

Audit Report prepared by Solidified covering the B-Protocol smart contracts for Compound.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code below. The debrief took place on February 01, 2021, and the results are presented here.

Fixes have been provided on February 15 and are reflected in this report.

Audited Files

The contracts audited were supplied in the following source code repository:

https://github.com/backstop-protocol/BCompound/pull/new/audit_solidified

commit: `c824d34b202a63ec5a00e0b34a07c83d0366557a`

Fixes were supplied in the following PR:

https://github.com/backstop-protocol/BCompound/pull/92

commit: `b958efe1c0a7fc2c266b431de0d854f263494400`

The scope of the audit was limited to the following components (files within the “contracts” folder):

- ./bprotocol/btoken/
  - BErc20.sol
  - AbsBToken.sol
  - BEther.sol
- ./bprotocol/
  - BComptroller.sol
  - Pool.sol
  - Registry.sol
  - Import.sol
- ./bprotocol/avatar/
  - AbsCToken.sol
  - AbsComptroller.sol
  - Avatar.sol

    - AbsAvatarBase.sol
- ./bprotocol/scoring/IBTokenScore.sol

Intended Behavior

The B-Protocol smart contracts provide an interface for existing lending protocols (Compound in this current version) aimed at avoiding gas wars during liquidation. This is achieved by the selected set of liquidators (members) that have priority for liquidations and share the liquidation proceeds based on a user scoring mechanism.

Findings

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than a security audit and vice versa.

Criteria	Status	Comment
Code complexity	Medium	-
Code readability and clarity	High	-
Level of Documentation	High	In addition to the detailed source code comments, the audit team has been provided with a comprehensive specification document.
Test Coverage	High	Unit and integration tests with good coverage have been supplied to the audits.

Issues Found

Solidified found that the B-Protocol contracts contain no critical issue, no major issue, 2 minor issues, in addition to 4 informational notes.

We recommend all issues are amended, while the notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	AbsAvatarBase.sol: Use of send() for Ether transfers may run out of gas	Minor	Acknowledged
2	AbsAvatarBase.sol: Explicit check without pre-defined modifier	Note	Resolved
3	AvbBToken.sol: Unused modifier	Note	Resolved
4	Incorrect comments	Note	Resolved
5	Registry.sol: Delegations are always revoked after single executions	Note	Acknowledged

Critical Issues

No critical issues have been found.

Major Issues

No major issues have been found.

Minor Issues

1. AbsAvatarBase.sol: Use of send() for Ether transfers may run out of gas

In line 204 the `send()` function is used to transfer Ether. Since the Istanbul hard fork, the gas stipend forwarded with this function may not be enough on the receiver's side to perform basic operations.

Recommendation
Whilst the intention of using `send()` according to the code comments is to avoid a DoS scenario, we recommend using `call()` to transfer Ether, to ensure enough gas is forwarded. DoS avoidance can usually be achieved by using a withdrawal pattern instead of pushing payments out.

Team Response
"We decided not to address issue #1 as preventing DOS is the highest priority here, and any other solution will either allow DOS or will force us to put hardcoded gas limitation on call, so we are fine with the default gas limitation of `send()`."

Notes

2. AbsAvatarBase.sol: Explicit check without pre-defined modifier

The function `_doLiquidateBorrow()` checks whether the caller is the pool contract, but does not use the `onlyPool()` modifier defined for the purpose.

Recommendation
Consider using the `onlyPool()` modifier to check the precondition for code clarity.

Update
Resolved

3. AvbBToken.sol: Unused modifier

The function `onlyPool()` modifier is defined but never used in this contract.

Recommendation
Consider using the `onlyPool()` modifier or removing it.

Update
Resolved

4. Incorrect comments

The comments document the code incorrectly in a couple of places:

`_untop()`: *AbsAvatarBase (L182 - L211)* - the return value in the comments does not exist

`uint public selectionDuration = 60 minutes; // member selection duration for round robin, default 10 mins`: *Pool.sol (L50)* - comment seems to be incorrect.

Recommendation
Correct the comments.

Update
Resolved

5. Registry.sol: Delegations are always revoked after single executions

The function `delegateAndExecuteOnce()` always revokes `delegate()` at the end, even in the case when the delegation had been granted earlier for this `msg.sender`. This invalidates any previous delegations.

Recommendation
Only revoke delegation in the case the delegation event was the result of this particular call.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of B-Protocol or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.

Solidified Technologies Inc.