SOLIDIFIED

Audit Report for Brane Capital - February 21, 2021

Summary

Audit Report prepared by Solidified covering the Brane Vault smart contract.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code below. The final debrief took place on February 8, 2021, and the results are presented here.

Audited Files

The source code has been supplied in the form of a private GitHub repository:

https://gitlab.com/bigindex/vault-audit/-/blob/master/brane.sol

Commit number: `24fe1cf993c8d544896f75ba91d6a72b5da9bc9f`

Intended Behavior

The smart contract implements vault for storing Ether that is controlled by a number of roles, with the following characteristics:

- Members can propose the addition or removal of other members
- All directors have to approve membership operation
- A requester can initiate transfers
- All directors have to approve transfers, but can only do so after the auditor has already approved it

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	Low	
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	N/A	Whilst no tests have been supplied with code, a textual description of 50 unit tests has been supplied to the auditors.

Issues Found

Solidified found that the Brane Vault contract contains no critical issue, 1 major issue, 3 minor issues, in addition to 1 informational note.

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	Function voteOnModerationRequest() counts None votes as Yea votes.	Major	Resolved
2	performTransfer() can potentially fail if it transfers ETH to a smart contract	Minor	Resolved
3	Moderator count not enforced	Minor	Resolved
4	Potential directorCount mismatch when adding/deleting directors	Minor	Resolved
5	Missing require error message in cancelModerationRequest()	Note	Resolved

Critical Issues

No critical issues have been found.

Major Issues

1. Function voteOnModerationRequest() counts None votes as Yea votes.

Function `voteOnModerationRequest()` will count any `None` vote as a `Yea` vote. This can result in a moderation request being approved without any participating members voting as `Yea`.

Recommendation
Require that `_vote != Vote.None`.

Updated
Resolved

Minor Issues

2. performTransfer() can potentially fail if it transfers ETH to a smart contract

Function `performTransfer()` calls `transfer()` to send ETH to `txn.destination`, which only forwards 2300 gas. In cases where `txn.destination` is a smart contract whose fallback function consumes more than 2300 gas, the call will always fail. This will have the side effect of preventing smart contracts (e.g. DAOs) from receiving any transfers.

For a more in-depth discussion of issues with `transfer()` and smart contracts, please refer to `https://diligence.consensys.net/blog/2019/09/stop-using-soliditys-transfer-now/`

Recommendation
Replace instances of `transfer()` with `call()`.

Updated
Resolved

3. Moderator count not enforced

The moderator count is not consistently enforced which can lead to the condition of at least 4 moderators (2 directors not being met) and `getMemberCount()` returning an incorrect result. When a vault is initialized, the constructor does not check for `address(0)` in the `moderators` array. This means that 0 address directors could be added. Furthermore, directors can be removed.

Recommendation
Check of `address(0)` when adding the moderators.

Updated
Resolved

4. Potential directorCount mismatch when adding/deleting directors

When adding a director that has already been added before, `directorCount` will be inconsistent with the actual number of directors. Furthermore, if two or more moderation requests are made to delete the same director and all get approved, `directorCount` will also be incorrect.

Recommendation
Check that a director address exists before adding or removing it.

Updated
Resolved

Notes

5. Missing `require` error message in `cancelModerationRequest()`

Function `cancelModerationRequest()` is missing an error message for `require(moderationRequest.state == TxState.Active)`.

Updated
Resolved

Audit Report for Brane Capital - February 21, 2021

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Brane Capital 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.