SOLIDIFIED

Audit Report for Aventus Protocol November 15, 2019

Summary

Audit Report prepared by Solidified for Aventus protocol covering various contracts and libraries included with it.

Process and Delivery

Three independent Solidified experts performed an unbiased and isolated audit of the below contracts. The debriefing took place on November 15, 2019, and the final results are presented here.

Audited Files

The following files were covered during the audit:

Contracts

- AVTManager.sol
- AventusStorage.sol
- EventsManager.sol
- MerkleLeafChallenges.sol
- MerkleRootsManager.sol
- Migrations.sol
- Owned.sol
- ParameterRegistry.sol
- ProposalsManager.sol
- ValidatorsManager.sol
- Versioned.sol

Proxies

- PAVTManager.sol
- PAventusTime.sol
- PDelegate.sol
- PEvents.sol
- PLibraryDelegate.sol
- PMerkleLeafChallenges.sol
- PMerkleRoots.sol
- PProposals.sol
- PValidators.sol

Libraries

- LAVTManager.sol
- LAVTStorage.sol
- LAventusTime.sol
- LEnums.sol
- LEvents.sol
- LEventsEvents.sol
- LEventsRoles.sol
- LEventsRules.sol
- LEventsStorage.sol
- LMerkleLeafChallenges.sol
- LMerkleLeafChecks.sol
- LMerkleLeafRules.sol
- LMerkleRoots.sol
- LMerkleRootsStorage.sol
- LProposals.sol
- LProposalsEnact.sol
- LProposalsStorage.sol
- LProposalsVoting.sol
- LValidators.sol
- LValidatorsChallenges.sol
- LValidatorsStorage.sol

Notes

- The audit was based on the solidity compiler `0.5.12+commit.7709ece9`
- The audit was performed on *AventusProtocolFoundation/protocol/tree/audit_ready* and commit `5826e607d18602a7ac198651f4a20a24ddcbba2b`

Issues

Minor

1. Owner and accounts who can *write* have too much control

AventusStorage.sol: The storage contract manages almost all the state of AventusProtocol including the balances. This means that anyone with permissions to write to storage can directly manipulate all balances, as well as other storage items. This may also cause inconsistencies between the balance history and the actual balances.

Recommendation

We recommend a more fine-grained permissions system or at least a control to disallow arbitrary edits by admin to sensitive data like tokens held and votes.

Follow up [11.12.2019]

Artos informed us that this is by design, and all owners (except for the smart contracts) will be the Aventus Protocol Foundation's multisignature wallet (the same one that has been used since the token sale), located at address `0x035a401972f228b58dcae76bf318b54ed036d680` (verifiable once deployed). The team also informed us of other acceptable mitigating factors, largely based on trust. The issue was initially reported as of major severity, but has been lowered to minor based on Artos' response. Artos response in full:

"We have firm governance, processes, regulatory sign off, legal review in multiple jurisdictions and a structure of companies that means we are as good as we possibly can be right now at navigating the regulatory grey zone that is utility token definitions across various jurisdictions. In order to better reflect this and to address your concern, all owners of smart contracts on the protocol will be set to be the Aventus Protocol Foundation's multisignature wallet (same one that has been used since the token sale) and publicly validatable here: https://etherscan.io/address/0x035a401972f228b58dcae76bf318b54ed036d680. Any signoff requires the council of the foundation to give approval (they also control sufficient keys to block any transaction)."

2. Signatures Validation is Vulnerable to Malleable Signatures.

The signature recovery function in `LECRecovery.sol` allows for malleable signatures in the upper ranges. These are still supported by the built-in `ecrecover` function ad should, therefore, be rejected.

Recommendation
Check for malleable signatures. See the code example in OpenZeppelin library for reference. https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/cryptography/ECDSA.sol

Amended [11.12.2019]
The issue was fixed and is no longer present in commit `5b26bc872f40c42710b4f87bfc81fb874009ce99z`.

Notes

3. Arbitrary crafting of user-controlled function calls should be avoided

LProposals.sol: `LProposals.endGovernanceProposal` (lines 112-128) uses `delegatecall` to a user-controlled function id `(implemented,None) = lProposalsEnactAddress.delegatecall(encodedFunctionCall)`. This should be avoided to avoid modification to introduce a failed function call.

Artos Response
Only function calls that have been agreed by the community can be executed; we understand that the community can deliberately choose to run a bad function.

4. Solidity Compiler Version should be locked

Pragma should be simplified and version should be locked for published contract, ideally using the most recent Solidity compiler version supported by the build framework.

Amended [11.12.2019]
The issue was fixed and is no longer present in commit
`5b26bc872f40c42710b4f87bfc81fb874009ce99z`.

5. Library to get current time can be avoided

LAventusTime.sol: An additional library which is used only to get the current block time is redundant and can be avoided.

Artos Response
This is required for mocking the time in testing: see libraries/Testing/LAventusTimeMock.sol

6. Remove unused event table variable

LEventsStorage - L8: Consider removing the unused eventsTable variable if it is not meant to be used anywhere.

Amended [11.12.2019]
The issue was fixed and is no longer present in commit
`5b26bc872f40c42710b4f87bfc81fb874009ce99z`.

7. Use of delegatecall and call

LProposalsEnact L84 - L97: Low level calls like `call` and `delegatecall` are not recommended except for some cases where it is part of the feature. Consider verifying the governance proposal bytecode before ending the proposal to avoid any unexpected outcome.

Artos Response
As for community agreed proposals discussed above.

8. Consider gas usage

Gas usage of some functions can be very high depending on factors like rules and other inputs. It is recommended to set a soft cap in the front-end on such factors to avoid any failed transaction because of gas limit. Additionally, state variables, `.balance`, or `.length` of non-memory arrays are used in the condition of for loop. In these cases, every iteration of loop consumes extra gas.

Artos Response
Acknowledged. No change needed.

9. Consider having a function to revoke event role if required

Current event roles only support adding roles in the event for validators. It is recommended to have a revoke role function which can be used to remove a user from a role if needed for usability purposes.

Artos Response
Acknowledged. Removing event roles causes problems for challenges. We may consider adding this functionality at a later date.

10. Access Management allows arbitrary access types

AventusStorage.sol: The contract allows registering any string as a key for access types in access management, but only two roles are actually used. This may create confusion and clutter the contract with unused access types.

Recommendation
We recommend limiting the access types to these actually used to avoid confusion. An enum type could be used instead of strings as an access type key.

Artos Response
Acknowledged. This is only used at deployment time and potentially in the future to block old versions of contracts from using storage. There are only two values and strings are more human readable. We prefer strings for this.

Closing Summary

A few issues were found during the audit which could break the intended behaviour, as well as several informational notes with opportunities for improvement. It is recommended for the Aventus Protocol team to address the issues. It is furthermore recommended to post the contracts on public bounty following the implementation of the suggested fixes.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of the Aventus Protocol or its products. This audit does not provide a security or correctness guarantee of the audited smart contracts. 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.