SOLIDIFIED

Audit Report for Topl. August 14, 2018.

Summary

Audit report prepared by Solidified, for Topl covering the Midgard sale contracts.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the below token sale. The debrief took place on August 14th, 2018 and the final results are presented here.

Audited Files

The following files were covered during the audit:

- arbits_presale.sol
- database.sol
- Iconiq_presale.sol
- topl_database_lib.sol
- safe_math_lib.sol

Notes:
The audit was performed on commit `fbce67ce6d1110768c3086ebaae76d85ad91f452`
The audit was based on the solidity compiler `0.4.24+commit.e67f0147`

Topl Midgard github repository: https://github.com/Topl/Midgard/

Intended Behavior

The purpose of these contracts is to operate a crowdsale for Topl's platform token. The detailed specifications are available at: (./docs/Topl Presale contract spec.pdf)

Issues Found

Critical

No critical vulnerabilities were identified.

SOLIDIFIED

Audit Report for Topl. August 14, 2018.

Major

1. The sale state is shared between `iconiq_presale` and `arbits_presale`

Although there is functionality to store different states for the two mentioned presales, the Iconiq presale sets and checks the `arbits_presale_open` boolean variable in the database contract.

So when the arbits presale is opened it will also open the Iconiq presale once again.

Recommendation

States should be isolated preventing one sale from interfering with past sales. Sale specific variables would benefit from isolation if stored on the sale contract, leaving only data supposed to be persistent, such as KYC and balance related data stored on the DB contract.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

Minor

2. Sale parameters can be arbitrarily changed by owner after sale started

The contracts give the owner absolute control over all aspects of the sales. By design users are required to trust Topl until their tokens are redeemed after Topl's mainnet launch. Other aspects of their sale are also centrally controlled, some of them trespassing the trustlesness level expected from a token sale, such as:

- Owner can destroy the sale contracts at any time, draining all the Ether.
- Owner can open and close sales at will.

- Owner can change rates, discounts and bonuses, at any time, including after the sale started and finished.
- Contract specifications state that "Topl will not withdraw more than the amount of Ether that has gone through KYC", but there is no requirement in the code preventing this action. There is no withdraw functionality for users that did not go through KYC.

Recommendation

Although the goal of the developer is not to create a full token specification, a minimum level of trustlessness will ensure that the sale goes as expected and is fair to the users.
Token sale parameters should remain constant after the sale start (and is communicated to users).
Allowing users to withdraw non-KYC Ether after a period of time is also, although not legally required, a best practice.
Lastly, contracts should not be destructible if Ether from customers is still stored in it.

Topl's Response [09-24-2018]

"While we appreciate and understand the views raised by the auditors that overall the Topl pre sale contracts are not fully trustless and require the pre sale participants to place a certain amount of trust in Topl. We believe this is an acceptable position since by the nature of how we have chosen to approach this sale, we are requiring users to place trust in us (specifically regarding the issuance of tokens on the Topl mainnet early next year). With this design approach in mind, we have addressed the concerns raised by the auditors below.

A. The sale is not set to close at a specific time so this is intended functionality
B. While we only expect to set these values once, we do not see any security vulnerability in maintaining the flexibility to change these variables if needed and announced to our community.
C. Although not a best practice, we will manually refund ether from users who do not pass the KYC check."

3. Presale hard cap `presale_arbits_total` can be bypassed

The token hard cap is stored in the database can be bypassed if the user's balance is directly updated through `set_participant_arbits` or `set_participant`

Recommendation

All transactions that update the amount of ERB tokens sold should be reflected in `presale_arbits_sold` and checked against `presale_arbits_total`.

Topl's Response [09-24-2018]

"This functionality is by design since manual updating will be done for fiat investments that occur off-chain, which may exceed the original hard cap that is in place for the publicly accessible sale."

4. Parameters not checked when presales are opened

Currently a presale can be opened without an Ether rate, minimum and maximum contributions, and bonus rates. This results in a the buying functions reverting.

Recommendation

We recommend that all parameters are checked when opening a sale, or accepted and checked in the constructor, preventing variables from not being initialized.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

5. `amount_of_pro_rata_tokens_subject_can_get` will be capped by previously owned arbits

If a user participates in the `arbits_presale` and purchases X amount of tokens and also wishes to take part in the `iconiq_presale` to obtain more arbits at a discount, the `amount_of_pro_rata_tokens_subject_can_get` will in actuality be (`amount_of_pro_rata_tokens_subject_can_get` - X) since the contract only checks against the total arbits a user owns.

Recommendation

Track users contributions to specifically the Iconiq presale contract, or encourage users to ensure they interact solely with the Iconiq presale contract if they would like to claim their full Iconiq discount.

Topl's Response [09-24-2018]

"Iconiq users should only interact with the iconiq_presale contract which they will be informed of. Purchasing through the general public contract means they are foregoing their discount rights."

6. If `presale_arbits_per_ether` is not set, any purchase of tokens during the Iconiq sale over the pro-rata token amount is not concluded.

Purchases in the Iconiq sale over the `num_of_pro_rata_tokens_alloted` are not processed if the variable `presale_arbits_per_ether` is not set, but Ether from the user is retained by sale contract and the transaction fails silently.

Recommendation

We recommend, as stated in the issue above, that all parameters are checked before initiating a sale. Transactions that depend on variables should revert if one of them has not been initialized, returning Ether send by users.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

7. Presale contracts are not deployable

Presale contracts call the database contract in their constructor, the transaction reverts because access to the contract has to be previously granted in the database contract. We also inspected the migrations and they do not run successfully.

Recommendation

Either the contract addresses will have to be calculated before or during deployment (`keccak256(address tx.origin, nonce)`) or they will have to be set later in the process.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

8. Presale `owner` mappings are not used

The `add_owner` and `remove_owner` functions in `arbits_presale.sol` only modify the local mapping of owners which is unused and does not add them to the database instance of owners. Similarly, the owners mapping in `iconiq_presale.sol` is unused.

Recommendation

Remove unused functionality or update it to interact with `database.sol` central owner mapping.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

9. Outdated version of SafeMath

The SafeMath contract was copied to the project and differs from current OpenZeppelin's implementation (https://github.com/OpenZeppelin/openzeppelin-solidity/pull/1187).

Recommendation

We recommend importing the last version of SafeMath through NPM, ensuring the version is the latest and also conforming with their licensing requirements.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

Notes

10. Ether change from a transaction is kept by the contract

The presale contracts, by design, retain the remainder of Ether form a transaction. Although a valid design decision, returning the funds to the customers within the same transaction is preferable. Other option in this regard is to add a require in the sale limits check within the `participate_in_arbits_presale_crypto` functions to ensure the sent ether is evenly divisible by the contracts designated `tokens_per_ether` value.

Topl's Response [09-24-2018]

"By design, we plan to educate users through instructions and to include a calculator on our website so users can avoid sending amounts that will be rounded."

11. Include messages in require statements

Starting in solidity 0.4.22, requires are able to add details of a revert. We suggest the addition of descriptive messages to the requires, such as the Iconiq member check, to allow for users to have more information about why their transaction reverted.

Topl's Response [09-24-2018]

"Chose not to implement this to avoid higher gas fees"

12. Testing coverage

There are currently `arbits_presale` tests (adapted from the database tests) but no `iconiq_presale` tests. We strongly recommend that tests are designed for each of the contracts.

Topl's Response [09-24-2018]

"We are aware and will work to expand coverage before deployment."

13. Documentation

The current documentation has several inconsistencies, as follows:

Funding contracts blue paper

- It states "11) To check the efficacy of step 15", but should read "11) To check the efficacy of step 9"
- If states "17) To check the efficacy of step 21", but should read "17) To check the efficacy of step 15"
- "Note: While not strictly necessary, if the price of ether shifts dramatically at any time during the arbits presale, the reader may choose to repeat steps 21-23 and 34-36 of Setup part 1 with new, price adjusted, values." but there is no section 36, so it is likely renumbering has occurred and these ranges should be updated.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

14. Styling

Styling makes the contract more readable and easier to understand, effectively minimizing the time needed for a developer, auditor or client to understand and be able to spot deficiencies in the source code, the following are recommendations for improvement:

- Use alias `1 ether` instead of `1000000000000000000` for easier readability
- Style guidelines would indicate using capitalized "Participant" for the name of a struct.

Amended [09-24-2018]

The issue has been fixed and is no longer present in commit `904b7d42284433496acfb203b90269adeef238f2`.

SOLIDIFIED

Audit Report for Topl. August 14, 2018.

Closing Summary

One major and several minor issues were found during the audit which can break the intended behaviour. It is strongly advised that these issues are corrected before proceeding with the crowdsale. It is furthermore recommended to post the contracts on Solidified public bounty afterwards.

Beyond the issues mentioned, the contracts were also checked for overflow/underflow issues, DoS and re-entrancy vulnerabilities. None were discovered.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of the Aventus Protocol Foundation 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.