SOLIDIFIED

Audit Report for Trident August 20, 2018.

Summary

Audit Report prepared by Solidified for Trident covering the Trident, Golden Gate, and PONO token contracts.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code below. The debrief took place on August 20, 2018, and the final results are presented here.

Audited Files

The following contracts were covered during the audit:

- Trident
- PONO
- GoldenGate

Notes

The audit was performed on the contracts deployed on the Ethereum mainnet at the following addresses, using verified code downloaded from Etherscan.io:

1. Trident - `0x4eea6844a4dc5bf3127decf034b3f4a7211ef2e7`
2. PONO - `0x61e3a9254a50ac93d806ba79adf0db3455cd0dd5`
3. GoldenGate - `0xb0db6f32f98b3c14ae45ff01b11efb953ad6a3d9`

Intended Behavior

The three contracts are identical except in their names. Each is an ERC20 token with the added ability for the owner to name “mint delegates” and “burn delegates”, who can mint and burn tokens respectively.

The issues listed below apply to all three contracts.

SOLIDIFIED

Audit Report for Trident August 20, 2018.

Issues Found

1. Ownership transfer does not clear delegates (minor)

Transferring ownership leaves the list of mint delegates and burn delegates intact. A new owner might reasonably expect that upon transfer, the previous owner's ability to mint or burn tokens has been revoked. This issue is mitigated by the new owner's ability to remove the previous delegates.

Recommendation
Clear the `mintDelegates` and `burnDelegates` arrays during ownership transfer.

2. Large delegates arrays could allow denial-of-service (note)

All operations involving the delegates — searching through the array, adding a new delegate, and removing a delegate — are linear in complexity with respect to the number of delegates. This risk is mitigated by the fact that the owner is the only one who can grow the array. A comment in the code notes that the number of delegates is expected to be small.

Recommendation
The code could be made safer by adding a mapping of addresses to indexes in the arrays. This would make all operations constant time (except enumeration, which would remain linear).

3. Update compiler version and base contracts (note)

The contracts were compiled with Solidity 0.4.19, and use older versions of OpenZeppelin base contracts.

Recommendation
Upgrade to 0.4.24 (the latest release) and adopt the new `constructor` and `emit` syntax. Check for relevant updates to base contracts and review and apply those as needed.

Audit Report for Trident August 20, 2018.

Closing Summary

Trident's contracts do not contain any critical issue that would require their redeployment. If the code is reused in the future it is strongly advised that the issues reported above are remediated.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of the Trident platform 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.

Solidified Technologies Inc.