SOLIDIFIED

Audit Report for Reserve - January 4, 2022

Summary

Audit Report prepared by Solidified covering the Reserve token smart contract.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief was on 3 January 2022.

The fixes were verified on 4 january 2022.

Audited Files

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

https://github.com/reserve-protocol/rsr-mainnet
Commit hash: `9af5fa4f3b020b822f3c10bcb5aef6c9cbcc309a`
Commit hash for fixes: `137182742078d9a9a91c7f66245f47b75fc30b39`

Contracts
├── Enchantable.sol
├── ForkSpell.sol
├── RSR.sol
├── SiphonSpell.sol
└── Spell.sol

Intended Behavior

The smart contracts implement an ERC20 token with additional features to migrate the balances of accounts from a previous contract.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases have their 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 equate to a higher risk. Certain bugs are more easily detected in unit testing than a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Medium	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

Issues Found

Solidified found that the Reserve token contracts contain no critical issues, no major issues, no minor issues and 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	Missing input address validation	Note	Resolved

Critical Issues

No issues found

Major Issues

No issues found

Minor Issues

No issues found

Notes

1. RSR.sol: Missing input address validation

The method `changePauser` does not restrict the sending `address(0)` as an input parameter. This is harmless and can be reverted when the contract is in the `SETUP` phase. When moved to the `WORKING` phase, the owner address becomes zero and any accidental update cannot be reverted back.

```
183 |   function changePauser(address newPauser) external onlyAdminOrPauser {
184 |       emit PauserChanged(pauser, newPauser);
185 |       pauser = newPauser;
186 |   }
```

Recommendation
Consider adding `address(0)` validation to the input parameter.

Update: Resolved

Audit Report for Reserve - January 4, 2022

Disclaimer

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