SOLIDIFIED

Audit Report for Spring Role November 1st, 2018.

Summary

Audit Report prepared by Solidified for Spring Role covering the Spring Token smart contract.

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the code below. The debrief took place on November 1st, 2018, and the final results are presented here.

Audited Files

The following contracts were covered during the audit:

- SPRINGToken.sol

Notes

The audit was based on the commit hash `a97c3056e86dd53f20c1c77ef41b8549021f4ea5`, solidity compiler `0.4.25+commit.59dbf8f1`

Intended Behavior

SPRINGToken.sol is an ERC20 token featuring an emergency pausable feature, the ability to mint tokens, and increase/decrease allowance steps, allowing for mitigation of the ERC20 “API Attack” (also known as Allowance Double-Spend).

SOLIDIFIED

Audit Report for Spring Role November 1st, 2018.

Issues Found

1. Base unit utilized within constructor is non-customary (note)

The SPRINGToken constructor takes in _maxSupply in a base unit that is not customary for token contracts. It is usually expected that all token amounts are passed to the contract using the minimum base token unit.

Recommendation
The line reading `maxSupply = _maxSupply.mul(10**decimals);` should be changed to `maxSupply = _maxSupply;`.

Amended [05-11-2018]
The issue was fixed and is no longer present in commit `93d83385cfa2f68d586dc6eb123e11e07f4b1609`.

2. Static analysis denotes potential integer overflow (note)

Static analysis warns of a potential integer overflow on line 116 and 213, each resulting from an arithmetic operation (addition) in which the potential summation could exceed the maximum allowed number size for the integer.

As arithmetic on line 116 is contained within a `require` statement, concern regarding potential overflow is mitigated. Similarly, addition on line 213 is followed by an `assert` statement, ensuring the two prior integers summed have not overflown (validating that the result is larger than the two integers that compose it), thereby negating potential impact.

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. Where applicable, utilize `npm` to update these dependencies in order to assure the most recent version of OpenZeppelin is pulled.

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. Where applicable, providing error messages for each require statement is suggested.

Amended [05-11-2018]

The issue was fixed and is no longer present in commit `93d83385cfa2f68d586dc6eb123e11e07f4b1609`.

SOLIDIFIED

Audit Report for Spring Role November 1st, 2018.

Closing Summary

Spring Token's contract does not contain any critical issue that should withhold its deployment. It is strongly advised that the issues reported above are remediated prior to usage.

Updated [05-11-2018]
All issues reported were amended and are no longer present iin commit `93d83385cfa2f68d586dc6eb123e11e07f4b1609.`

Disclaimer

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