SHIELD Contracts Review

This document describes review results of SHIELD smart-contracts.

Provided files contains following contracts:

1. Coin contract;
2. CoinContract contract;
3. CoinStorage contract.

Coin contract

This section describes issues related to the Coin contract.

EIP-20 Compliance Issues

This section lists issues of smart contract related to EIP-20 requirements.

1. Line 120: A token contract which creates new tokens SHOULD trigger a Transfer event with the _from address set to 0x0 when tokens are created.
2. Line 165: Transfers of 0 values MUST be treated as normal transfers and fire the Transfer event.
3. Line 187: The function SHOULD throw if the _from account balance does not have enough tokens to spend.
4. Line 196: Transfers of 0 values MUST be treated as normal transfers and fire the Transfer event.
5. Line 220: The function SHOULD throw unless the _from account has deliberately authorized the sender of the message via some mechanism.

Arithmetic Overflow Issues

This section lists issues of the token smart contract related to the arithmetic overflows.

1. Line 128: Overflow may accrue since no precautions were made.

Documentation and Readability Issues

This section lists cases where the code is correct, but too involved and/or complicated to verify or analyze.

1. Line 152: Indirect call of method lowers readability.
2. General lack of documentation and comments, as well as circular dependencies between contracts lowers readability.
3. It became a common practice to separate “crowdsale” contracts which implement sell and refund mechanics from token contract. This practice allow to keep token clean from sell mechanics, and destroy crowdsale if necessary.

Unclear Behavior

This section lists issues of the token smart contract, where the contract behavior is unclear: the business logic might be violated here, but the documentation and functional requirements are not sufficiently documented to make a clear decision.

1. Token implements payable and refund features, which adds complexity and creates circular dependencies with CoinContract contract and may result in unpredictable behavior.
2. Line 64: Method name misleading, since current contract is also called “Coin”. Related issues: line 29 (event name: CurrentCoin), line 16 (field name: curr_coin). Those are clearly referencing another contract, not current one.
3. Line 121, 132: Coin will re-issue additional supplies if required, so it’s not clear when it’ll be required to issue anything in advance.
4. Line 192: Adds ability for token owner to transfer funds from any address.

Suboptimal Code

This section lists suboptimal code patterns, which were found in the smart contract.

1. Line 166: Check is redundant.
2. Lines 200-212: Code duplication which may be avoided.

Dangerous Behavior

This section lists problems related to the code vulnerability. Problems of this type require additional attention, as they can result in a serious loss.

-- *none* --

Moderate Flaws

This section lists moderate flaws found in the token smart contract.

1. Line 54: Function is vulnerable for reentrance attack if “coin” will be owned by another smart contract, not by wallet or account.
2. Line 224: Method is vulnerable for double approve attack.

Major Flaws

This section lists major flaws found in the token smart contract.

1. Line 128: Arithmetic overflow may accrue since no precautions were made.

Other Issues

This section lists stylistic and other minor issues which were found in the token smart contract.

-- *none* --

CoinContract contract

This section describes issues related to the CoinContract contract.

Arithmetic Overflow Issues

This section lists issues of the token smart contract related to the arithmetic overflows.

1. Line 123: Overflow may accrue since no precautions were made.

Unclear Behavior

This section lists issues of the token smart contract, where the contract behavior is unclear: the business logic might be violated here, but the documentation and functional requirements are not sufficiently documented to make a clear decision.

1. Purpose of this contract is unclear. It probably serves as a mediator between token and CoinStorage contracts.
2. Line 125: If “proxy” is Coin contract, than called method have no return value. Otherwise, return value of method do not handled explicitly.
3. Line 126: Same event will be logged twice here and in the token contract.
4. Line 128: Amount of ETH to return and corresponding amount of tokens to burn calculated elsewhere, which is not coherent with buy procedure, when amount of tokens to sell was calculated in the contract.

Moderate Flaws

This section lists moderate flaws found in the token smart contract.

1. Line 133: Refund events never emitted.

CoinStorage contract

This section describes issues related to the CoinStorage contract.

Arithmetic Overflow Issues

This section lists issues of the token smart contract related to the arithmetic overflows.

1. Line 52: Overflow is possible.
2. Line 57: Underflow is possible.

Unclear Behavior

This section lists issues of the token smart contract, where the contract behavior is unclear: the business logic might be violated here, but the documentation and functional requirements are not sufficiently documented to make a clear decision.

1. Purpose of this contract is unclear. It probably could be merged with CoinContract.

Recommendations

Based on my findings, we recommend the following:

1. Fix the major flaws.
2. Make the token EIP-20 compliant.
3. Check the issues marked as “unclear behavior” against functional requirements.
4. Fix the vulnerable code.
5. Refactor the code to remove suboptimal parts.
6. Improve code readability.
7. Fix the moderate issues.
8. Restructure contracts to match OpenZeppelin or TokenMarketNet impementations