SOLIDIFIED

Audit Report for MyBit. April 29, 2018.

Summary

Audit Report prepared by Solidified for MyBit covering the token swap contract.

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the below token swap. The debrief took place on April 29, 2018 and the final results are presented here.

Audited Files

The following files were covered during the audit:

- ERC20.sol
- ERC20Interface.sol
- SafeMath.sol
- TokenSwap.sol
- TokenSwap.sol (with receiveApproval).

Intended Behavior

The purpose of these contracts is to create a new token to replace an existing one, in a token swap event.

Notes:
The audit was conducted on commit `49f060af7d675fb0c2a78eb1aaa9519d60b3e31d`
The audit was based on the Solidity compiler version `0.4.19+commit.c4cbbb05`

Issues Found

Major

1. Minting old tokens can deprive existing holders

The owner of the contract can mint old tokens after the sanity check was made, allowing him/her to claim new tokens. This would result in other users being unable to claim all their tokens, since there would be more old tokens than specified in `circulatingSupply`.

Recommendation
There's a business decision to be made here about whether it's better to leave this as-is (meaning users have to trust the contract owner to not mint more tokens) or to uncap the token swap, which would effectively make the new token mintable as well.

AMENDED[07.07.2018]
Function to prevent swaps if old tokens are minted was added. Issue fixed by MyBit team in commit `2b2151903c786bbc35c4126f10179303cbee8bf2`

Notes

2. ERC20 constructor should trigger a Transfer event

The ERC20 standard states that a transfer event should be triggered when tokens are being created, using the `0x0` address as `from`.

Recommendation
Add the event in the ERC20 constructor.

AMENDED[07.07.2018]
Issue fixed by MyBit team in commit `2b2151903c786bbc35c4126f10179303cbee8bf2`

3. sanityCheck() can be improved

The `sanityCheck()` function mostly checks for hardcoded values, which might not actually represent the real world values, since the old token total supply can be changed.

Recommendation
For values which can be fetched at execution time from the old token, the sanity check should do so. For any values which cannot be fetched at execution time, the sanity check isn't doing anything useful and should be removed.

If the sanity check remains, it would reduce code complexity to do it in the constructor rather than as a separate step.

AMENDED[07.07.2018]
Sanity check was removed, but some checks are still being made in the constructor. Issue fixed by MyBit team in commit `2b2151903c786bbc35c4126f10179303cbee8bf2`

4. Derivation of variable oldTotalSupply (awaiting clarification)

The old token contract does not make any reference to a total supply cap, only to the existing tokens. This variable allows for arbitrary minting of new tokens to the MyBit foundation, so we would like clarification on where this value came from.

AMENDED[07.07.2018]
There is a public article by MyBit talking about the supply cap:
https://medium.com/@MyBit_Dapp/post-tokensale-summary-b44169003be0
Issue fixed by MyBit team in commit `2b2151903c786bbc35c4126f10179303cbee8bf2`

5. Consider adding a function to retrieve mistakenly sent tokens

Any token sent directly to the contract (not approving first) will be forever locked. Consider adding a function to retrieve such tokens, since the appropriate usage might be confusing for some users.

Recommendation
Add a function to manually return tokens sent directly to the contract.

6. Include a swap function in addition to receiveApproval

Some users may prefer to approve the old token and swap for the new token in separate transactions. (For example, they may use a tool that already knows how to approve ERC20 tokens, or they may want to double check the approval before committing the swap.)

Recommendation
In addition to `receiveApproval()`, include a specific `swap()` function that allows users to do the approval and swap in separate transactions.

AMENDED[07.07.2018]

Function to prevent swaps if old tokens are minted was added. Issue fixed by MyBit team in commit `2b2151903c786bbc35c4126f10179303cbee8bf2`

7. Update compiler version

We recommend using the latest compiler, currently `0.4.23`. This version includes support for a less ambiguous constructor syntax, “reason strings” to communicate why transactions are reverted, and a clearer “emit” syntax for events.

AMENDED[07.07.2018]
Issue fixed by MyBit team in commit `2b2151903c786bbc35c4126f10179303cbee8bf2`

Closing Summary

One major issue was found during the audit that could break the desired behaviour. MyBit addressed this issue, along with several other recommendations as stated by the amendments in the report.

Disclaimer

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