Lympo_Audit_Report_18_02_2019.md

Audit Report for Lympo Tokens and Multi-signature wallet contracts

Summary

This audit report covers the new LympoToken contract and a multi-signature wallet Multisig contract.
The audit was done by Gundas Vilkelis on 16-18th of February, 2019.
All the important issues were addressed by Lympo Team and no blocking issues were identified in the final versions of the contracts.

Contracts audited

- **Lympo.sol** - new Lympo Token contract. The latest reviewed version of the contract is deployed at: https://rinkeby.etherscan.io/address/0xff65d22223f3fe1d0e0cf7b01cf310bce9b8db6f
- **Multisig.sol** - Multi-Signature wallet contract which will hold Ether and Lympo Tokens' funds. The latest reviewed version of the contract is deployed at: https://rinkeby.etherscan.io/address/0x377536f039d62f33dc8d58546d7044427fe2682a

Summary of Issues Found

No critical issues were reported.
Two major issues were reported - issue **#1** was fixed and issue **#2** was agreed to be not valid.
One minor issue **#3** was reported and was addressed by the Lympo team.
A few suggestions for improvement were identified (**#4**, **#5** and **#6**).

Major

1. Locked funds can be transferred using transferFrom()

When using `transferFrom(...)` function, the `_ownerTransfer(..)` and `_ecosystemAddrTransfer(...)` checks do not work correctly (they check the minimum balance of `msg.sender`), thus the owner's and eco system's token vesting (time-locking) is not enforced.

Follow-up 17.02.2018: The issue was addressed by Lympo team.

2. LympoToken owner() can transfer time-locked funds

The `owner()` of LympoToken contract can transfer ownership to another account. After doing this, the previous owner will be able to transfer all the time-locked tokens, because the time-lock checks will be checking the balance of the new `owner()`.

Suggested fix: The gas optimization suggestion item **#6** would address this issue, since the time-locked funds would not be part of the owner's balance.

Follow-up 17.02.2018: This issue is **not valid**, because the LympoToken contract does not allow changing the owner.

Minor

3. Funds would be locked if Multisig wallet is created with only one owner

If the Multisig wallet is created with only one owner, any funds transferred to such a wallet will be locked forever, because the creator of a transfer request is always added to the approvers list and consequently he cannot call the approve functions anymore.

Suggested fix: Either implement a check in the `constructor` that a number of owners must be greater than `1`, or change the implementation of `createNewEtherWithdrawRequest(...)` and `createTransferTokensRequest(...)` to call the `approve*Request(...)` instead of modifying the `confirmators` list.

Follow-up 18.02.2018: The issue was addressed by adding a check in the Multisig constructor that a number of owners must be greater than two. Additionally the modification was made in constructor which requires that a minimum `confirmationCount` is greater than one.

Notes and suggestions for improvements

4. Add Recover Token functionality to LympoToken contract

The current LympoToken contract deployed on the MainNet (https://etherscan.io/address/0x57ad67acf9bf015e4820fbd66ea1a21bed8852ec) has around half million LYM tokens "trapped" inside the contract, because the LYM tokens have been transferred to the Lympo contract itself by mistake. Adding a Token Recovery function would allow the owner to recover such tokens:

```
function recoverToken(address _token) public onlyOwner returns (bool) {
  ERC20 token = ERC20(_token);
  uint256 balance = token.balanceOf(address(this));
  token.transfer(owner, balance);
}
```

Follow-up 18.02.2018: The suggestion was partially implemented in LympoToken contract - it will be possible to recover "trapped" tokens only after March, 2020.

5. The Multisig contract does not follow the recommended Checks-Effects-Interactions pattern

Multisig contract code does not follow the recommended Checks-Effects-Interactions pattern (see https://solidity.readthedocs.io/en/v0.5.4/security-considerations.html#re-entrancy for details):

```
withdrawEther[withdrawEtherId].toAddr.transfer(withdrawEther[withdrawEtherId].amount);
withdrawEther[withdrawEtherId].completed = true;
```

The code above is not currently exploitable (since transfer() function forwards only a limited amount of gas), but it is highly recommend to follow the best practices and update the state before calling the `transfer()` on an external contract.

Suggested fix: Mark the `withdrawEther` request completed before calling `transfer(...)`.

Follow-up 18.02.2018: The suggestion was implemented in Multisig contract.

6. transfer(...) and transferFrom(...) gas consumption reduction suggestion

Current implementation of `transfer(...)` and `transferFrom(...)` functions enforce the time-locking of certain portion of owner's and EcoSystem's funds. These checks consume gas for every token transfer. A more gas efficient implementation would have no time-lock checks in `transfer(...)` and `transferFrom(...)` functions (thus, there is no need for `LympoToken` contract to even override those functions implemented in `ERC20` base contract). The time-locking feature could be implemented by initially not assigning the time-locked funds to the owner's and EcoSystem's balances, but alternatively providing a function for the owner and the EcoSystem to claim those funds once the time-lock period has passed.

Follow-up 18.02.2018: The suggestion was implemented in LympoToken contract, see the item #7 below.

7. EcoSystem's balance migration points of attention

This item is a result of suggestion #6 implementation (as implemented in the final version of LympoToken contract). The latest implementation imposes the certain requirements on the token balances migration process:

1. The EcoSystem's old (current) LympoToken balance should not be migrated during the batch migration (airdrop), because the EcoSystem's funds will be stored and time-locked in the LympoToken contract once the new LympoToken is deployed.
2. The old (current) Lympo Token contract deployed on the MainNet does not allow the EcoSystem to spend any funds until February 28th, 2019. If the migration is done after this date, it is important to verify that the actual EcoSystem's balance in the old LympoToken contract matches the time-locked EcoSystem's funds assigned in the new LympoToken contract (which is currently 146652000 LYM tokens and matches the current EcoSystem's token balance in the old contract).

Disclaimer

The information appearing in this audit report is for general discussion purposes only and is not intended to provide any legal security guarantees to any individual or entity.