GVISP 1
your space

2018.03.12.

COIN PAYMENT PROCESSOR.ORG

#OPEN CONSORTIUM cPRO

development@coinpaymentprocessor.org

GVISP1 LTD

#CP PROCESSOR PARTNER

office@gvisp.com

cPRO BUYBACK

SMART CONTRACT AUDIT

GVISP1 LTD.

GVISP1 Ltd. **** Your space**** Bul.Vojvode Misica 17, 11000 Belgrade, Serbia, EU

1. Audit description

One contract was checked**: buyback.sol.**

The purpose of this audit is to check all the functionalities of buyback.sol contract, and to determine the level of security and the probability of adverse outcomes.

buyback.sol is the contract that buys cPRO tokens at a price (in ETH) that is periodically pushed by a centralized oracle service.

The oracle sends the price that is in accordance with the price of cPRO established on external exchanges. In the following lines, the contract is being referenced by the name of the file where it was written in. The file contains exactly one contract, so there is no room for confusion.

2. Quick review

- ✓ All the functions and state variables are well commented which is good in order to understand quickly how everything is supposed to work.
- ✓ The contract was written in accordance with solidity's aesthetic standards (names of state variables and functions start with lowercase letter, names of events start with capital letter, etc)
- ✓ During deployment, creator must provide the constructor with the address of cPRO token, and with the initial address that is going to be authorized for certain actions.

3. A brief review of contract's functionalities

- ✓ State variables of the contract are the address of cPRO token (unchangeable), total amount of ETH located inside the contract (used for buying cPRO tokens) and current cPRO price in WEI.
- ✓ There is only one type of mapping that remembers whether the address is authorized or not.
- ✓ Authorize and unauthorize functions are used to grant/cancel privileges for certain address (they can be called only by authorized addresses).
- ✓ Fallback function is simple which is a good practice. It simply updates the information about the total amount of Eth located inside the contract.
- ✓ Withdraw Tokens is used to withdraw strayed tokens from the contract and can only be called by authorized addresses.
- ✓ Buyback function is used to sell cPRO for ETH. Sent cPROs are burnt and an equivalent amount of ETH is transfered to a sender's address.
- ✓ SetPrice function updates the currentPrice of cPRO in WEI. It will be called periodically by the oracle.

4. Functionalities test

After deployment

- ➢ totalFundsAvailable = 0 : ✓
- ➢ currentPrice = 0 (updates when setPrice is called) : ✓
- ➢ authorizedAddresses[initialAuthorizedAddress] = true : ✓
- ➢ authorize : ✓
- ➢ unauthorize : ✓
- ➢ fallback : ✓
- ➢ buyback : ✓
- ➢ setPrice : ✓
- ➢ withdrawTokens : ✓

5. Detailed code check (line-by-line)

- ✓ After deployment, initial authorized address is set and has privileges to call setPrice, withdrawTokens, authorize and unauthorize functions.
- ✓ buyback.sol file imports the erc20.sol file that provides the interface for token manipulation.

State variables of the contract

- ➢ address **cProAddress** - address of the cPRO token (unchangeable)
- ➢ uint256 **totalFundsAvailable** - total amount of ETH inside the contract
- ➢ uint256 **currentPrice** - current price of 1 cPRO in WEI (provided by the oracle)

- ✓ There is one type of mapping: (address => bool) authorizedAddresses that remembers whether the address is authorized or not.
- ✓ Authorize - function that grants privileges to a certain address (can be called only by authorized addresses).
- ✓ Unauthorize - function that cancel privileges for a certain address (can be called only by authorized addresses).
- ✓ Fallback - updates the totalFundsAvailable (notices the ETH sent to the contract).
- ✓ WithdrawTokens - transfers strayed tokens to sender's address. Only authorized addresses can call this function. If cPRO tokens are strayed ones, it wouldn't be possible to extract them from the contract because of the given condition "require(tokenContractAddress != cProAddress)", which is probably unnecessary (cPRO tokens are most likely to stray into the contract).
- ✓ Buyback - burns sender's cPRO tokens and transfers ETH.
- ✓ SetPrice - function that is periodically called by the oracle. It updates the currentPrice.

6. Static analysis test, vulnerabilities and outcomes

- ✓ Static analysis of the code was conducted and no security flows were found.

https://oyente.melon.fund

```
browser/buyback.sol:Buyback
EVM Code Coverage : 81.6%
Callstack Depth Attack Vulnerability : False
Re-Entrancy Vulnerability : False
Assertion Failure: False
Parity Multisig Bug 2 : False
Transaction-Ordering Dependence (TOD) : False
```

Over & Underflows

- ✓ Both over and underflows are not possible.

7. Final comments

buyback.sol file imports the erc20.sol file that contains the interface with only 3 functions, so it could be more practical to define the interface inside the buyback.sol file itself.

The contract is simple, short and well commented.Maybe “require(tokenContractAddress != cProAddress)” should be removed from withdrawTokens function, as it may cause locking of accidentally sent cPROs.