GVISP 1
your space

2018.03.12.

COIN PAYMENT PROCESSOR.ORG

#OPEN CONSORTIUM cPRO

development@coinpaymentprocessor.org

GVISP1 LTD

#CP PROCESSOR GOLD PARTNER

office@gvisp.com

MAIN

ETHEREUM SMART CONTRACT AUDIT

GVISP1 LTD.

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

1. Audit description

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

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

Main.sol is a contract that regulates privileges and sets up the entire system. 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

Main.sol is the administration contract for entire system. It regulates privileges other contracts and accounts have.

- ✓ Constructor parameter for **Main.sol** is the list of addresses that will be able to give authorization to other contracts and accounts.
- ✓ “SuperAdmins” number will be equal to the length of that list, and it cannot change in time.
- ✓ All functions and state variables are well commented which is good in order to understand quickly how everything is supposed to work.
- ✓ This contract must be deployed first as its address will be parameter for all other contract deployments.
- ✓ After deployment of **Main.sol**, only superAdmin addresses are set. After that, superAdmins can call Main.sol’s functions in order to setup the system and give authorization to other contracts and accounts.

3. A brief review of contract’s functionalities

Contract contains many functions and almost all of them can be called only by superAdmins. One function can be called by authorized addresses, and one can be called by “mint authorized” addresses.

Functions only superAdmin can call are:

- ➢ \`authorize\`, \`unauthorize\`, \`mintAuthorize\`, \`mintUnauthorize\` (functions that grant/deny privileges to other addresses);
- ➢ \`setOracleAddress\`, \`setBuybackAddress\`, \`setCProTokenAddress\` (functions that set addresses for oracle, buyback contract, and cPRO token);
- ➢ \`setMinCProAmount\` (functions that sets the minimum amount of cPRO tokens one must have in order to call certain functions within the system, it is probably unnecessary for this function to activate only by superAdmin, it would make more sense if all authorized addresses could call it);
- ➢ \`setDoToAddress\`, \`setEuToAddress\`, \`setKrToAddress\`, \`setYeToAddress\`, \`setYuToAddress\` (functions that set addresses of stable tokens; each of these functions will be called only once after each stable token deployment);

- `setDoESAddress`, `setUpESAddress` (functions that set addresses of supplements tokens;
- each of these functions will be called only once after deployment of DoES and UpES tokens);
- `activateStableToken`, `deactivateStableToken` (functions that allow/permit conversion of specific stable token within other project contracts)

Function only authorized addresses can call:

- `setOraclePercent` (sets the percentage of fee from PersonalEthConversion.sol contract that should go to oracle account)

Function only “mint authorized” addresses can call:

- `setPause` (sets pause for entire system - no conversions between values can be done while “pause” is on)

4. Functionalities test

- authorize: ✓
- unauthorize: ✓
- mintAuthorize: ✓
- mintUnauthorize: ✓
- setPause: ✓
- setOracleAddress: ✓
- setBuybackContractAddress: ✓
- setOraclePercent: ✓
- setCProTokenAddress: ✓
- setMinCProAmount: ✓
- setDoToAddress: ✓
- setEuToAddress: ✓
- setKrToAddress: ✓
- setYeToAddress: ✓
- setYuToAddress: ✓
- setDoEsAddress: ✓
- setUpEsAddress: ✓
- activateStableToken: ✓
- deactivateStableToken: ✓

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

After deployment, only “superAdmin” addresses are set. The number of “superAdmin” addresses is constant during time, and cannot change. “superAdmin” addresses are the ones that can give “standard” and “mint” authorization to other addresses, and call majority of contract’s functions.

“superAdmin” addresses actually are able to call all system’s functions, even the ones only authorized or “mint authorized” addresses can call.

This can be done simply by calling `authorize` and `mintAuthorize` functions from “superAdmin” addresses, by giving the same address as a parameter to these functions. This is why it is so important for “superAdmin” addresses to remain unchangeable and secure.

State variables of the contract:

- uint256 **pausedUntil** - timestamp of the moment when conversions between values inside other system contracts will be possible again
- uint256 **minCProAmount** - minimum amount of cPRO tokens one must have in order to call certain functions within the system
- uint256 **oraclePercent** - percentage of fee (in ETH) from PersonalEthConversion.sol contract that should go to oracle account
- address **oracleAddress** - oracle account address
- address **buybackContractAddress** - cPRO buyback contract address
- address **cProTokenAddress** - cPRO token address
- address **dotoAddress** - DoTo stable token address
- address **eutoAddress** - EuTo stable token address
- address **krtoAddress** - KrTo stable token address
- address **yetoAddress** - YeTo stable token address
- address **yutoAddress** - YuTo stable token address
- address **doesAddress** - DoES token address
- address **upesAddress** - UpES token address

Mappings:

- (address => bool) activatedStables - tracks whether specific stable token is activated for conversion
- (address => bool) superAdminAddresses - remembers superAdmin addresses set during deployment
- (address => bool) authorizedAddresses - remembers authorized addresses set by some “superAdmin” address
- (address => bool) mintAuthorizedAddresses - remembers “mint authorized” addresses set by some “superAdmin” address

Modifiers:

- onlySuperAdmin - checks if msg.sender is “superAdmin”
- onlyAuthorized - checks if msg.sender is authorized

- ➢ onlyMintAuthorized - checks if msg.sender is “mint authorized”

<u>Functions:</u>

- ➢ \`authorize\`, \`unauthorize\`, \`mintAuthorize\` and \`mintUnauthorize\` functions can be called only by “superAdmin” addresses; they grant/deny privileges to other addresses
- ➢ \`setPause\` function can only be called by “mint authorized” addresses (stable token contracts will be the ones mint authorized)
- ➢ \`setOracleAddress\` - function that sets (changes) the oracle address; only superAdmin can call it and it would be more practical if authorized addresses could call it, so that superAdmin addresses may be used only in extreme situations
- ➢ \`setBuybackContractAddress\` - function that sets (changes) the buyback contract address; only superAdmin can call it and the same goes here as for \`setOracleAddress\`
- ➢ \`setOraclePercent\` - function that sets the percentage of fee (in ETH) that should go to oracle account
- ➢ \`setCProTokenAddress\` - function that sets the address of cPRO contract, should be called only once
- ➢ \`setMinCProAmount\` - function that changes the minimum amount of cPRO tokens one must have in order to call certain functions inside other project contracts; only superAdmin may change it and the same goes here as for \`setOracleAddress\` and \`setBuybackContractAddress\`
- ➢ \`setDoToAddress\`, \`setEuToAddress\`, \`setKrToAddress\`, \`setYeToAddress\`, \`setYuToAddress\` functions that set stable tokens addresses; they should be called only once after deployment of stable tokens
- ➢ \`setDoESAddress\`, \`setUpESAddress\` - functions that set derivative tokens addresses; they should be called only once after deployment of derivatives
- ➢ \`activateStableToken\` and \`deactivateStableToken\` - functions that make specific stable token available/unavailable for conversions inside the system

<u>6. Static analysis test, vulnerabilities and outcomes</u>

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

https://oyente.melon.fund
browser/stable.sol:Main
EVM Code Coverage :
Callstack Depth Attack Vulnerability : False
Re-Entrancy Vulnerability : False
Assertion Failure: False
Parity Multisig Bug 2 : False
Transaction-Ordering Dependence (TOD) : False

✓ Over and underflows are not possible in this contract.

7. Final comments

`resume` function that breaks pause may be added as well (such that only superAdmin could call it).

More than one superAdmin address is also advised for backup reasons. It would be more logically consistent if authorized addresses could call `setOracleAddress`, `setBuybackContractAddress` and `setMinCProAmount` functions instead of superAdmins.

Generally, the code is simple and well commented. Contract does not import any interfaces and all functions are simple and short which is good.