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

SUPPLEMENTS

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: **Supplements.sol.**

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

Supplements.sol is a contract that tracks the global supplements of stable tokens, e.g. the difference between the value of total convert ETH and the value of circulating stable tokens (minted amount - the amount stored as convert for derivatives).

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

Supplements.sol tracks the global supplements of stable tokens.

- ✓ Constructor parameters for **Supplements.sol** are the address of **Main.sol** and the list of stable tokens addresses.
- ✓ Both **Main.sol** and stable tokens must be deployed before **Supplements.sol**. If not all stable tokens are deployed at the same time, it wouldn't present a problem for Supplements.sol contract because of its function `setStableTokenAddresses`.
- ✓ 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 after **Main.sol** as it takes **Main.sol's** address as a parameter.
- ✓ No stable tokens must be deployed before **Supplements.sol**, but they can be.

3. A brief review of contract's functionalities

Contract contains many functions and almost all of them can be called only by "mint authorized" addresses.

All functions are used to change state variables of the contract so that `supplementsInStables` (constant) function can return the right value.

All functions that change state variables can be called only by "mint authorized" addresses, except `setListedTokens` and `setStableTokens` functions that can be called only by "superAdmin". This should probably be changed so that authorized addresses can call this function (in order for system to remain consistent).

- ➢ `increaseTotalConvertEth`/`decreaseTotalConvertEth` - increases/decreases the number of convert ETH and is supposed to be called by **PersonalEthConversion.sol** contract when conversion happens.
- ➢ `increaseTotalConvertTokens`/`decreaseTotalConvertTokens`- increases/decreases the number of convert tokens and is supposed to be called by **PersonalTokenConversion.sol** contract when conversion happens.

- `increaseTotalStableConvert`/`decreaseTotalStableConvert`- increases/decreases the number of convert stable tokens and is supposed to be called by **PersonalEthConversion.sol** and **PersonalTokenConversion.sol** contracts when conversion happens.
- `setStableTokens` - function that sets addresses of stable tokens and can be called only by superAdmin (should be changed so that authorized addresses can call it).
- `setListedTokens` - function that sets address of tokens that may be put as convert for stable tokens within PersonalTokenConversion.sol contract.
- `supplementsInStables` - (constant) function that calculates global supplements and returns it in terms of stable tokens based on specified stable token address; can be called by any address

4. Functionalities test

- setStableTokens: ✓
- increaseTotalConvertEth: ✓
- decreaseTotalConvertEth: ✓
- increaseTotalConvertTokens: ✓
- decreaseTotalConvertTokens: ✓
- increaseTotalStableConvert: ✓
- decreaseTotalStableConvert: ✓
- setListedTokens: ✓
- supplementsInStables: ✓

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

- ✓ After deployment, only **Main.sol** contract address is set, and optionally the addresses of stable tokens (not necessary addresses of all stable tokens)
- ✓ **Supplements.sol** contract "reads authorizations" from Main.sol contract in order to secure function calls only from system contracts and superAdmins.

State variables of the contract:

- address **mainContractAddress** - address of Main.sol contract
- uint265 **totalConvertEth** - total amount of ETH stored as convert inside the system
- address[] **stableTokens** - the list of stables addresses
- address[] **listedTokens** - the list of tokens that can be stored as convert within PersonalTokenConversion.sol contract in exchange for stable tokens

Mappings:

- (address => uint256) totalConvertTokens - total amount of specific token stored as convert inside PersonalTokenConversion.sol contract
- (address => bool) listed - mapps token address to “true” if token is listed; this information is not used in any other contract so it should probably be removed
- (address => uint256) totalStableConvert - total amount of specific stable token stored as convert inside PersonalEthConversion.sol and PersonalTokenConversion.sol contracts

Modifiers:

- onlyMintAuthorized - checks if msg.sender is “mint authorized”
- onlySuperAdmin - checks if msg.sender is “superAdmin”
- Both modifiers “read authorization” from Main.sol contract

Functions:

- `setStableTokens` - function that sets addresses of stable tokens; this means not all stable tokens must be active (deployed) at the same time; only “superAdmins” can call this function, and it would make more sense if other authorized addresses could call it
- `increaseTotalConvertEth` / `decreaseTotalConvertEth` - increases/decreases total amount of convert ETH and is supposed to be called only by **PersonalEthConversion.sol** contract (actually contracts, as there is 1 PersonalEthConversion contract for each stable token)
- `increaseTotalConvertTokens` / `decreaseTotalConvertTokens` - increases/decreases total amount of convert tokens and is supposed to be called only by PersonalTokenConversion.sol contract (actually by 5N **PersonalTokenConversion.sol** contracts as there is one contract for each token for each stable token, and N is the number of listed tokens)
- `increaseTotalStableConvert` / `decreaseTotalStableConvert` - increases/decreases total amount of stable tokens that are stored as convert for DoES and/or UpES tokens; is supposed to be called only by **PersonalEthConversion.sol** and **PersonalTokenConversion.sol** contracts
- `setListedTokens` - sets the addresses of tokens that are listed on the platform; only “superAdmin” can call this function, and it would make more sense if authorized addresses could call it
- `supplementsInStables` - (constant) function that returns the amount of non-convertalized stable tokens circulating expressed in units of specified stable token.

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/stable.sol: Supplements
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

mapping (address => bool) listed should be removed as it is not used by any of contracts.

It would be more logically consistent if authorized addresses could call `setStableTokens` and `setListedTokens` functions instead of superAdmins.

Generally, the code is simple and well commented. Contract imports **Main.sol** and **StableToken.sol** interfaces and all functions are simple and short, except for `supplementsInStables` that must loop through listedTokens and StableTokens lists, which is good.