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

STABLE TOKENS

ETHEREUM SMART CONTRACT AUDIT

1. Audit description

One contract was checked: **StableToken.sol**
The purpose of this audit is to check all functionalities of **StableToken.sol** contract, and to determine level of security and probability of adverse outcomes.
StableToken.sol is a contract that generates "Stable Tokens" that always should have value equal to the value of a specific fiat currency. These tokens can be bought for ETH or any other token that has been listed, by calling certain functions within other project contracts, anytime at price of the fiat currency in ETH/Token that is provided by a centralized Oracle service.
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

StableToken.sol contains all characteristics and functionalities of ERC20 standard.

- ✓ All functions and state variables are well commented which is good in order to understand quickly how everything is supposed to work.
- ✓ The number of token decimals is 2 which is not generally recommended, but responds to problem's demands (smallest unit of USD is its second decimal, etc).
- ✓ After deployment of the contract, no stable tokens are generated, and the initial totalSupply equals 0.
- ✓ Stable Tokens are generated afterwards when someone wants to exchange ETH/Tokens. This is done through other contracts that have permission to call `mint` function that generates stable tokens.

3. A brief review of contract's functionalities

All ERC20 functions are contained in the contract.
Initially, totalSupply equals 0, and stable tokens are generated when someone sends ETH or other tokens to other project contracts that are not part of this particular audit.
Functions that are not part of ERC20 standard:

- ✓ `mint` function generates stable tokens and mapps them to defined address' balance (can be called only by "mint authorized" addresses).
- ✓ `setPriceInEth` and `setPriceInToken` functions are called when stable token price in ETH or other tokens needs to be updated. Only authorized addresses can call this function.
- ✓ `reset` function resets prices of ETH, UpES and DoES tokens to given values and only "superAdmin" can call it.

4. Functionalities test

- ✓ Total amount of generated tokens = 0 (initially) : ✓
- ✓ Token symbol : ✓
- ✓ Token name : ✓
- ✓ Token decimals : ✓
- ✓ Current price in Eth: ✓
- ✓ Current price in tokens: ✓
- ✓ setPriceInEth : ✓
- ✓ setPriceInToken : ✓
- ✓ mint : ✓
- ✓ reset: ✓

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

- ✓ Functionalities of the ERC20 standard were not changed, so only functions that do not belong to ERC20 will be analyzed and stated.
- ✓ Stable Tokens name and symbol are defined during deployment, and number of decimals equals 2.
- ✓ Total amount of generated stable tokens after deployment is 0, and can be both increased (by calling `mint` function) and decreased (by calling `burn` or `burnFrom` function).

State variables of the contract

- address **mainContractAddress** - Main.sol contract address (administration contract that regulates privileges)
- string **name** - stable token name
- string **symbol** - stable token symbol
- uint256 **decimals (=2)** - number of stable token decimals
- uint256 **totalSupply** - amount of generated stable tokens
- uint256 **currentPriceInEth** - price of the smallest unit of stable token in WEI
- uint256 **currentPriceInDoES** - price of the smallest unit of stable token in smallest units of DoES token
- uint256 **currentPriceInUpES** - price of the smallest unit of stable token in smallest units of UpES token

Mappings

- (address => uint256) currentPriceInToken - mapps token address to the price of the smallest stable token unit in smallest units of that token

Modifiers

- onlySuperAdmin - checks if msg.sender is “superAdmin”
- onlyAuthorized - checks if msg.sender is authorized to call `setPriceInEth` and `setPriceInToken` function
- onlyMintAuthorized - checks if msg.sender is authorized to call `mint` function

During deployment, creator must specify the address of Main.sol contract that regulates both “standard” and “mint” authorization, stable token name and symbol.

After deployment, currentPriceInEth will be equal to 0, as well as currentPriceInToken for each token. Those will remain 0 until setPriceInEth and setPriceInToken functions aren’t called for the first time.

- ✓ setPriceInEth - function that updates the price of smallest stable token unit in WEI. If the new price differs from the previous one for more than 5%, all processes in other contracts (that are not part of this audit) will be paused for one hour.
- ✓ setPriceInToken - function that updates the price of smallest stable token unit in smallest units of other token. If the new price differs from the previous one for more than 5%, all processes in other contracts will be paused for one hour.
- ✓ mint - function that generates stable token.
- ✓ Parameters are `_amount` (amount of stable tokens that will be created), `_to` (address where these tokens will appear).
- ✓ Only “mint authorized” addresses can call this function (only contracts that are part of this project will be “mint authorized”, and “mintAuthorization” is regulated inside Main.sol contract, which is not the part of this audit, by “superAdmin addresses”).
- ✓ reset - function that resets ETH, DoES and UpES prices, and can be called only by superAdmin.

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: StableToken
EVM Code Coverage : 83.4%
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

An additional event may be added that will fire when `mint` function is called. Also, `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, but number of superAdmins is regulated inside **Main.sol** contract.