# PSY Audit Report

**May 26, 2023**



# Table of Contents

# Summary

This report has been prepared for PSY smart contract, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

# Overview

## Project Summary

| | |
|---|---|
| Project Name | **PSY** |
| Codebase | **https://github.com/psystablecoin/psy-contract** |
| Commit | **39fbc591ab9b12dad163d7dbe1fc2714ee8dde82** |
| Language | **Solidity** |

## Audit Summary

| | |
|---|---|
| Delivery Date | **May 26, 2023** |
| Audit Methodology | **Static Analysis, Manual Review** |
| Total Isssues | **3** |

## [WP-L1] The implementation of `maxFlashLoan()` is incompatible with EIP-3156

**Low**

### Issue Description

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/BorrowerOperations.sol#L691-L700

```
691  /**
692       * @dev The amount of currency available to be lent.
693       * @param _token The loan currency.
694       * @return The amount of `token` that can be borrowed.
695       */
696      function maxFlashLoan(
697          address _token
698      ) external view override returns (uint256) {
699          return 0;
700      }
```

https://eips.ethereum.org/EIPS/eip-3156

> The maxFlashLoan function MUST return the maximum loan possible for token. If a token is not currently supported maxFlashLoan MUST return 0, instead of reverting.

### Recommendation

Consider changing to:

```
691   /**
692       * @dev The amount of currency available to be lent.
693       * @param token The loan currency.
694       * @return The amount of `token` that can be borrowed.
695       */
696  function maxFlashLoan(
697      address _token
698  ) external view override returns (uint256) {
```

```
699         if (_token == SLSDToken) {
700             return type(uint256).max - SLSDToken.totalSupply();
701         }
702         return 0;
703     }
704
```

## Status

✓ Fixed

# [WP-I2] Potential Issues with the Use of ETH as Collateral and FlashLoan in the Current System

**Informational**

## Issue Description

Currently, the system uses `address(0)` to represent ETH. The `asset` parameter for minting via `flashLoan` is also `address(0)` . If ETH is used as collateral and `emergencyStopMintingCollateral` is triggered, then `flashLoan` will be unable to function.

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/BorrowerOperations.sol#L1106-L1123

```
1106  function getMethodValue(
1107      address _asset,
1108      uint256 _amount,
1109      bool canBeZero
1110  ) private view returns (uint256) {
1111      bool isEth = _asset == address(0);
1112
1113      require(
1114          (canBeZero || (isEth && msg.value != 0)) || (!isEth && msg.value == 0),
1115          "BorrowerOp: Invalid Input. Override msg.value only if using ETH asset,
      otherwise use _tokenAmount"
1116      );
1117
1118      if (_asset == address(0)) {
1119          _amount = msg.value;
1120      }
1121
1122      return _amount;
1123  }
```

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/SLSDToken.sol#L71-L79

```
71  function mint(
72      address _asset,
73      address _account,
74      uint256 _amount
75  ) external override {
76      _requireCallerIsBorrowerOperations();
77      require(!emergencyStopMintingCollateral[_asset], "Mint is blocked on this
    collateral");
78      _mint(_account, _amount);
79  }
```

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/BorrowerOperations.sol#L656-L676

```
656 function flashLoan(
657     IERC3156FlashBorrower _receiver,
658     address _token,
659     uint256 _amount,
660     bytes calldata _data
661 ) external override returns(bool) {
662     require(msg.sender == flashloaner || flashloaner == address(0), "FlashLoan:
    Unauthorized caller");
663     uint256 _supplyBefore = SLSDToken.totalSupply();
664     SLSDToken.mint(address(0), address(_receiver), _amount);
665     require(
666         _receiver.onFlashLoan(msg.sender, address(SLSDToken), _amount, 0, _data)
    == CALLBACK_SUCCESS,
667         "FlashLoan: Callback failed"
668     );
669     SLSDToken.burn(address(_receiver), _amount);
670     uint256 _supplyAfter = SLSDToken.totalSupply();
671     require(
672         _supplyAfter == _supplyBefore,
673         "FlashLoan: Repay failed"
674     );
675     return true;
676 }
```

## Status

✓ Fixed

# [WP-N3] Inconsistent Event Emission for `TreasuryAddressChanged`

## Issue Description

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/PSY/PSYStaking.sol#L199-L203

```
199     function changeTreasuryAddress(address _treasury) public onlyOwner {
200         require(_treasury != address(0), "Treasury address is zero");
201         treasury = _treasury;
202         emit TreasuryAddressChanged(_treasury);
203     }
```

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/TroveManager.sol#L1245-L1248

```
1245     function changeTreasuryAddress(address _treasury) public onlyOwner {
1246         require(_treasury != address(0), "Treasury address is zero");
1247         treasury = _treasury;
1248     }
```

https://github.com/psystablecoin/psy-contract/blob/7ed354f6c9497614e01e07c691148ed4135bc077/contracts/BorrowerOperations.sol#L632-L635

```
632     function changeTreasuryAddress(address _treasury) public onlyOwner {
633         require(_treasury != address(0), "Treasury address is zero");
634         treasury = _treasury;
635     }
```

## Status

✓ Fixed

# Appendix

## Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by WatchPug; however, WatchPug does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

# Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Smart Contract technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.