# Float Protocol
# Security Review Report

August 2, 2025



Version 1.0


| Email | Telephone |
| --- | --- |
| audits@extropy.io | +44 1865338228 |

| | |
|---|---|
| Data Classification | Client Confidential |
| Client Name | Float Protocol |
| Document Title | Float Protocol Audit |
| Document Version | 1.0 |
| Author | Extropy Audit Team |

# Contents


**1 Executive Summary** **1**

**2 Audit summary** **2**
2.1 Audit scope . . . 2
2.2 Issues Summary . . . 2
2.3 Methodology . . . 2
2.4 Approach . . . 2

**3 System Overview** **3**

**4 Findings** **4**
4.1 [INFO] Check function arguments for valid values . . . 4
4.2 [INFO] Use a consistent compiler version . . . 4
4.3 [INFO] State variable visibility . . . 4
4.4 [INFO] Function Visibility . . . 5
4.5 [INFO] Data type size . . . 5
4.6 [INFO] Default Variable Assignment . . . 5
4.7 [INFO] Unneeded Library Call . . . 5

**A Disclaimers** **6**
A.1 Client Confidentiality . . . 6
A.2 Proprietary Information . . . 6

# 1 | Executive Summary

This report presents the findings of the security assessment conducted on behalf of Float Protocol. The assessment was conducted between 21/04/21 and 04/05/2021 and was authorised by Float Protocol. The project was of high quality, consequently we have only Informational issues to report. The code followed best practices and was appropriate to its purpose. There was clearly a testing framework in place, complemented by the use of mock contracts. All issues reported here are reported for information only, nevertheless, it was recommended that these be reviewed and addressed , all of which was done.

# 2 | Audit summary

The assessment was conducted between 21/04/21 and 04/05/2021 and was authorised by Float Protocol.

- The Audit was conducted from commit 521f9c79f32b4ad66bf6192d72a5229d75c6ccba to commit bfcb47b228ff8ef65d942bba26fb8b346912395b

## 2.1 | Audit scope

The scope of the audit was the code contained in the primary-protocol branch of the repo, with the following focus. Primary - contracts/auction/* - contracts/funds/* - contracts/policy/* - contracts/lib/* Secondary - contracts/oracle/* - contracts/tokens/* Out of scope: - contracts/external-lib - contracts/staking/

## 2.2 | Issues Summary

| ID | Finding | Status |
|---|---|---|
| 4.1 | [INFO] Check function arguments for valid values | Resolved |
| 4.2 | [INFO] Use a consistent compiler version | Resolved |
| 4.3 | [INFO] State variable visibility | Resolved |
| 4.4 | [INFO] Function Visibility | Resolved |
| 4.5 | [INFO] Data type size | Resolved |
| 4.6 | [INFO] Default Variable Assignment | Resolved |
| 4.7 | [INFO] Unneeded Library Call | Resolved |

## 2.3 | Methodology

### 2.3.1 | Risk Rating

The risk rating given for issues follows the standard approach of the OWASP Foundation. We combine two factors :

- Likelihood of exploit
- Impact of Exploit

The Categories we use are *Critical*, *High*, *Medium*, *Low* and *Informational* These categories may not align with the categories used by other companies.
The informational category is used to contain suggestions for optimisation (where this is not seen as causing significant impact), or for alternative design or best practices.

## 2.4 | Approach

The project was assessed mainly by code inspection, with auditors working independently or together, looking for possible exploits.
Tests were written were possible to validate the issues found.
Static analysis tools were also used, and summary results are attached to this report. These tools are seen as an adjunct to code inspection.

## 3 | System Overview

The FLOAT token is a stablecoin whose value can change over time in response to demand. Expansion and contraction in supply are handled by auctions, and an additional token BANK is used as a governance token.

# 4 | Findings

## 4.1 | [INFO] Check function arguments for valid values

- **Location(s):**. MonetaryPolicyV1.sol
- **Description:** Check function arguments for valid values

  The values passed to functions should be checked for validity

  ```
  function setTargetPrice(uint256 _targetPrice)
  ```

  However the function has XX vulnerability....
- **Recommendation:** Use a require statement to ensure on valid values are accepted by a function.
- **Status:** Resolved.

## 4.2 | [INFO] Use a consistent compiler version

- **Location(s):**. Math.sol
- **Description:** Use a consistent compiler version

  A fixed rather than floating compiler version that is consistent between contracts should be used.

  ```
  pragma solidity >=0.5.0;
  ```

  References :https://consensys.net/blog/developers/solidity-best-practices-for-smart-contract-security/
- **Recommendation:** Fix the compiler version with pragma solidity =0.7.6; in all contracts
- **Status:** Resolved.

## 4.3 | [INFO] State variable visibility

- **Location(s):**. AuctionHouse.sol, Staged.sol
- **Description:** State variable visibility

  State variables can be made immutable or internal. For optimisation, ff a getter function is not needed, it is recommended to declare state variables as immutable or internal.

  ```
  ISupplyControlledERC20 public override float;
  ```

  References : https://medium.com/coinmonks/gas-optimization-in-solidity-part-i-variables- 9d5775e43dde
- **Recommendation:** Declare variables as internal or immutable where suitable
- **Status:** Resolved.

## 4.4 | [INFO] Function Visibility

- **Location(s):**. ChainlinkEthUsdConsumer.sol, Staged.sol
- **Description:** Function Visibility

  Marking functions as external rather than public reduces gas cost. Arguments in external functions use call data rather than memory, which is a cheaper allocation.

  ```
  function updateStage() public timedTransition returns (Stages)
  ```

  References : https://medium.com/coinmonks/gas-optimization-in-solidity-part-i-variables- 9d5775e43dde
- **Recommendation:** Prefer external over public whenever possible.
- **Status:** Resolved.

## 4.5 | [INFO] Data type size

- **Location(s):**. AuctionHouse.sol
- **Description:** Data type size

  The cost of allocation of storage variables depends on their size.

  ```
  uint256 _auctionCooldown
  ```
- **Recommendation:** Use the smallest size possible for variables and constants
- **Status:** Resolved.

## 4.6 | [INFO] Default Variable Assignment

- **Location(s):**. AuctionHouse.sol
- **Description:** Default Variable Assignment

  It is unnecessary to set variables to their default value

  ```
  uint256 public round = 0;
  ```
- **Recommendation:** Remove the assignment
- **Status:** Resolved.

## 4.7 | [INFO] Unneeded Library Call

- **Location(s):**. AuctionHouse.sol
- **Description:** Unneeded Library Call

  Where overflow is infeasible, a call to the SafeMath library is unnecessary

  ```
  round = round.add(1);
  ```
- **Recommendation:** Use the increment operator to save gas.
- **Status:** Resolved.

# A | Disclaimers

The audit makes no statements or warranty about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purpose

## A.1 | Client Confidentiality

This document contains Client Confidential information and may not be copied without written permission.

## A.2 | Proprietary Information

The content of this document should be considered proprietary information. Extropy gives permission to copy this report for the purposes of disseminating information within your organisation or any regulatory agency.

# extropy

