ClarityAlliance

# VELAR STABLESWAP SECURITY REVIEW

**Conducted by:**
KRISTIAN APOSTOLOV, STORMY

SEPTEMBER 18TH, 2024

## CONTENTS



# 1. About Clarity Alliance

**Clarity Alliance** is a team of expert whitehat hackers specialising in securing protocols on Stacks.

They have disclosed vulnerabilities that have saved millions in live TVL and conducted thorough reviews for some of the largest projects across the Stacks ecosystem.

Learn more about Clarity Alliance at [clarityalliance.org](clarityalliance.org).

## CONTENTS



# 2. Disclaimer

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Clarity Alliance to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Clarity Alliance’s position is that each company and individual are responsible for their own due diligence and continuous security. Clarity Alliance’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree
to analyze.

The assessment services provided by Clarity Alliance are subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis.

Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third parties. Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Clarity Alliance does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

CONTENTS



# 3. Introduction

A time-boxed security review of the Velar Stableswap implementation, where Clarity Alliance reviewed the scope and provided remediation recommendations.

# 4. About Velar Stableswap

Velar Stableswap is an AMM designed for efficient, low-slippage trading of stable assets, drawing inspiration from Curve V1's whitepaper. It serves as a middle ground between constant sum and constant product formulas, enabling stable pricing and high liquidity for stablecoin pairs. For more details on the protocol's underlying logic, refer to the Curve.fi Whitepaper.

Read more about Velar here.

# 5. Risk Classification

| Severity | Impact: High | Impact: Medium | Impact: Low |
|---|---|---|---|
| Likelihood: High | Critical | High | Medium |
| Likelihood: Medium | High | Medium | Low |
| Likelihood: Low | Medium | Low | Low |

## 5.1 Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

## CONTENTS



## 5.2 Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

## 5.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

CONTENTS



# 6. Security Assessment Summary

**Review Commit Hash:**

3cfe226131efbf03e1b6c2c8ac81092fc739430e

## Scope

The following contracts were in the scope of the security review:

- `curve-fees.clar`
- `curve-lp-token.clar`
- `curve-math.clar`
- `curve-pool.clar`

## CONTENTS



# 7. Executive Summary

Over the course of the security review, Kristian Apostolov, Stormy engaged with Velar to review Velar Stableswap. In this period of time a total of **3** issues were uncovered.

## Protocol Summary

| | |
|---|---|
| **Protocol Name** | Velar Stableswap |
| **Date** | September 18th, 2024 |
| **Protocol Type** | Stableswap AMM |

## Findings Count

| Severity | Amount |
|---|---|
| Low | 1 |
| QA | 2 |
| **Total Findings** | **3** |

## CONTENTS



# Summary of Findings

| ID | Title | Severity | Status |
|---|---|---|---|
| [L-01] | Missing Limit on Numerator in update-swap-fee Allows DoS of Swaps | Low | Resolved |
| [QA-01] | init Lacks Check to Prevent LP Token from Doubling as a Pair Token | QA | Resolved |
| [QA-02] | Lack of Trait Validation in collect May Result in Revenue Reset Without Fee Collection | QA | Resolved |

CONTENTS



# 8. Findings

## 8.1. Low Findings

### [L-01] Missing Limit on Numerator in `update-swap-fee` Allows DoS of Swaps

#### Description

The `update-swap-fee` function adjusts the swap fee used in the curve pool. Currently, when changing the swap fee, a safeguard ensures that the new values do not exceed the maximum allowed fee per swap.

The system performs two checks: one to ensure the denominator is set to 10,000 units, and another to verify that the numerator is at least 9,950 units. However, a critical third check is missing, which should ensure that the numerator does not exceed 10,000 units. Without this check, the system allows a swap fee to be set with a numerator greater than the denominator. This issue causes the `calc-fees` function, which is called during a swap, to revert when calculating fees, leading to DoS for the swap function.

#### Recommendation

Add a third check in the `update-swap-fee` function to ensure that the numerator value does not exceed 10,000 units.

CONTENTS



# 8.2. QA Findings

## [QA-01] `init` Lacks Check to Prevent LP Token from Doubling as a Pair Token

### Description

The `init` function is used by the deployer to set the initial values of the pool. Currently, the function includes a check to ensure that the same token is not used for both `t0` and `t1` . However, it does not check to ensure that the LP token is not used as one of the pair tokens.

### Recommendation

Add a precondition to the `init` function to ensure that the LP token is not used as one of the pair tokens ( `t0` or `t1` ).

## CONTENTS



# [QA-02] Lack of Trait Validation in `collect` May Result in Revenue Reset Without Fee Collection

## Description

The system generally ensures that the correct traits are provided across functions. However, validation is missing in the restricted `collect` function. The `collect` function is intended to gather any revenue earned from protocol fees. If incorrect traits are provided, the system could mistakenly reset the revenue without actually collecting the fees, leading to potential accounting discrepancies.

```
(define-public
  (collect
    (token0 <ft-trait>)
    (token1 <ft-trait>))
(let ((user tx-sender)
(protocol (as-contract tx-sender))
        (rev      (get-revenue))
        (amt0     (get token0 rev))
        (amt1     (get token1 rev)) )
    ;; Pre-conditions
    (try! (check-protocol-fee-to))
    ;; Update global state
(if (> amt0 u0)
    (try! (as-contract
       (contract-call? token0 transfer amt0 protocol user none)))
    false)
(if (> amt1 u0)
(try! (as-contract
         (contract-call? token1 transfer amt1 protocol user none)))
        false)
    ;; Update local state
    (unwrap-panic (reset-revenue))
```

## Recommendation

Implement trait validation in `collect` to ensure that the correct traits are used.