

Clanker A-4

Security Audit

August 18, 2025

Version 1.0.0

Table of Contents

- [Introduction](#)
- [Overall Assessment](#)
- [Specification](#)
- [Source Code](#)
- [Issue Descriptions and Recommendations](#)
- [Security Levels Reference](#)
- [Issue Details](#)
- [Disclaimer](#)

Introduction

This document includes the results of the security audit for Clanker's smart contract code as found in the section titled 'Source Code'. The security audit was performed by the Macro security team from August 6, 2025 to August 11, 2025.

The purpose of this audit is to review the source code of certain Clanker Solidity contracts, and provide feedback on the design, architecture, and quality of the source code with an emphasis on validating the correctness and security of the software in its entirety.

Disclaimer: While Macro's review is comprehensive and has surfaced some changes that should be made to the source code, this audit should not solely be relied upon for security, as no single audit is guaranteed to catch all possible bugs.

Overall Assessment

The following is an aggregation of issues found by the Macro Audit team:

Severity	Count	Acknowledged	Won't Do	Addressed
Code Quality	1	-	-	1

Clanker was quick to respond to these issues.

Specification

Our understanding of the specification was based on the following sources:

- Discussions on Telegram with the Clanker team.

Source Code

The following source code was reviewed during the audit:

- **Repository:** [contracts](#)
- **Commit Hash (initial):** b5fe9897d292b57a4425637103db762c27ce73b3
- **Commit Hash (final):** c57c323813da3087f2ffdbb58352123dab75b2e2

Specifically, we audited the following contracts within the repository:

Source Code	SHA256
src/hooks/ClankerHookDynamicFeeV2.sol	118e93cfe1a1f6f28369a982cabe9f0085fb4f5c08285c5fad82fe5c5a6aae4d
src/hooks/ClankerHookStaticFeeV2.sol	0d5ef6870201512399a05c3ee060d715c5586eb40f287c907a5ac9651935b403
src/hooks/ClankerHookV2.sol	0bf744bf8531294ac3e90e0703a5240a8ff2c2a84ec3bf581b3e7d01008ce9ef
src/hooks/interfaces/IClankerHookV2.sol	d057a501b4e2dcf4d43b811303379ef759e7883ce05ab77cfb057c118f50b792
src/hooks/interfaces/IClankerHookV2PoolExtension.sol	503abc915b849673b5e06b3338a9dfe5d7fd0c8809169d0fc4bbf92e1da1bc91
src/hooks/pool-extension-examples/UniV3SwapBack.sol	5280e5dc867dd8efa241e40c334d0222d9e03cf053ee9f09c1699cf14968ca77
src/mev-modules/ClankerMevDescendingFees.sol	e3433c35d491548398dc6817a5e1012ef97f8a64c63a9d9b3a93e31b36aedfaa
src/mev-modules/ClankerSniperAuctionV2.sol	fd3402c8b98a01ce6043872a3983e82b019213658bbd64a07a4beec6336fa044

Source Code	SHA256
src/mev-modules/sniper- utils/ClankerSniperUtilV2.sol	65cb614002481f16f64015494099bf1f82 f8da8fd224fe9c338798916ff42ce7

Note: This document contains an audit solely of the Solidity contracts listed above. Specifically, the audit pertains only to the contracts themselves, and does not pertain to any other programs or scripts, including deployment scripts.

Issue Descriptions and Recommendations

Click on an issue to jump to it, or scroll down to see them all.

 1 Unnecessary imports and code

Security Level Reference

We quantify issues in three parts:

1. The high/medium/low/spec-breaking **impact** of the issue:

- How bad things can get (for a vulnerability)
- The significance of an improvement (for a code quality issue)
- The amount of gas saved (for a gas optimization)

2. The high/medium/low **likelihood** of the issue:

- How likely is the issue to occur (for a vulnerability)

3. The overall critical/high/medium/low **severity** of the issue.

This third part – the severity level – is a summary of how much consideration the client should give to fixing the issue. We assign severity according to the table of guidelines below:

Severity	Description
(C-x) Critical	We recommend the client must fix the issue, no matter what, because not fixing would mean significant funds/assets WILL be lost.
(H-x) High	We recommend the client must address the issue, no matter what, because not fixing would be very bad, or some funds/assets will be lost, or the code's behavior is against the provided spec.
(M-x) Medium	We recommend the client to seriously consider fixing the issue, as the implications of not fixing the issue are severe enough to impact the project significantly, albeit not in an existential manner.
(L-x) Low	The risk is small, unlikely, or may not be relevant to the project in a meaningful way. Whether or not the project wants to develop a fix is up to the goals and needs of the project.
(Q-x) Code Quality	The issue identified does not pose any obvious risk, but fixing could improve overall code quality, on-chain composability, developer ergonomics, or even certain aspects of protocol design.
(I-x) Informational	Warnings and things to keep in mind when operating the protocol. No immediate action required.
(G-x) Gas Optimizations	The presented optimization suggestion would save an amount of gas significant enough, in our opinion, to be worth the development cost of implementing it.

Issue Details

Q-4 Unnecessary imports and code

TOPIC	STATUS	QUALITY IMPACT
Spec	Fixed ↗	Low

1. In the ClankerHookV2 and ClankerMevDescendingFees contracts, the following import is unnecessary.

```
import {console} from "forge-std/console.sol";
```

2. In the ClankerHookV2, explicit call syntax is used to access StateLibrary functions, therefore following is unnecessary

```
using StateLibrary for *;
```

3. In the ClankerHookV2, there is a duplicate import from BeforeSwapDelta.sol contract and second import with related usage can be removed as it unnecessary:

```
// duplicate import which can be removed
import {
    BeforeSwapDelta, BeforeSwapDeltaLibrary
} from "@uniswap/v4-core/src/types/BeforeSwapDelta.sol";

// usage
using BeforeSwapDeltaLibrary for BeforeSwapDelta;
```

Disclaimer

Macro makes no warranties, either express, implied, statutory, or otherwise, with respect to the services or deliverables provided in this report, and Macro specifically disclaims all implied warranties of merchantability, fitness for a particular purpose, noninfringement and those arising from a course of dealing, usage or trade with respect thereto, and all such warranties are hereby excluded to the fullest extent permitted by law.

Macro will not be liable for any lost profits, business, contracts, revenue, goodwill, production, anticipated savings, loss of data, or costs of procurement of substitute goods or services or for any claim or demand by any other party. In no event will Macro be liable for consequential, incidental, special, indirect, or exemplary damages arising out of this agreement or any work statement, however caused and (to the fullest extent permitted by law) under any theory of liability (including negligence), even if Macro has been advised of the possibility of such damages.

The scope of this report and review is limited to a review of only the code presented by the Clanker team and only the source code Macro notes as being within the scope of Macro's review within this report. This report does not include an audit of the deployment scripts used to deploy the Solidity contracts in the repository corresponding to this audit. Specifically, for the avoidance of doubt, this report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. In this report you may through hypertext or other computer links, gain access to websites operated by persons other than Macro. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such websites' owners. You agree that Macro is not responsible for the content or operation of such websites, and that Macro shall have no liability to your or any other person or entity for the use of third party websites. Macro assumes no responsibility for the use of third party software and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.