3JANE · SECURITY REVIEW

3Jane

3Jane LCC Leveraged Fund Helper Review.

A security review of 3Jane LCC Leveraged Fund Helper Review.

CLIENT	PROJECT	AUDITORS	AUDIT WINDOW	FINAL REPORT
3Jane	3Jane LCC Leveraged Fund Helper Review	Felipe Gomez, Vlad Zotov, Michael Lett, Ferran Celades	October 19 – October 27, 2026	November 11, 2026

Published at guardian.security/portfolio/3jane/lcc-leveraged-fund-helper-review-2026-11-11

§ 00

Contents.

15 pages: overview, confidence ranking, scope, 3 findings, disclaimer, and about Guardian.

§ 01 **Overview** P. 03

§ 02 **Confidence Ranking** P. 04

§ 03 **Scope & Classifications** P. 05

Files in Scope P. 06

§ 04 **Findings** P. 07

Main Review · 3 findings P. 08

§ 05 **Disclaimer** P. 14

§ 06 **About Guardian** P. 15

§ 01

Overview.

Guardian reviewed 3Jane LCC Leveraged Fund Helper Review across 2 review rounds. The final report includes 3 Informational findings.

AUDIT FIRM
Guardian

CLIENT
3Jane

PROJECT
3Jane LCC Leveraged Fund Helper Review

PREPARED BY
Felipe Gomez, Vlad Zotov, Michael Lett, Ferran Celades

AUDIT WINDOW
October 19 – October 27, 2026

FINAL REPORT
November 11, 2026

REPOSITORY
https://github.com/3jane-protocol/moneymarket-contracts

METHODOLOGY
Smart contract · Manual · Main Review · Remediation Review

MAIN REVIEW COMMIT
fdd6a7222195

REMEDIATION REVIEW COMMIT
8d1fd2e4a09a

1.1 Confidence Ranking

5 / 5

Very High Confidence

Guardian assigns a Confidence Ranking of 5 out of 5, combining finding severity, remediation outcome, and the reviewer's assessment of latent risk. The full scale is on the next page.

1	2	3	4	5
VERY LOW CONFIDENCE	LOW CONFIDENCE	MODERATE CONFIDENCE	HIGH CONFIDENCE	VERY HIGH CONFIDENCE

1.2 Findings at a Glance

SEVERITY	TOTAL	PENDING	ACK	PARTIAL	RESOLVED
Critical	**0**	0	0	0	0
High	**0**	0	0	0	0
Medium	**0**	0	0	0	0
Low	**0**	0	0	0	0
Informational	**3**	0	0	0	3
All severities	**3**	**0**	**0**	**0**	**3**

§ 02

Confidence Ranking.

Guardian assigns a project-level confidence score using the scale below. The ranking combines finding severity, remediation outcome, and the reviewer's assessment of latent risk.

CONFIDENCE RANKING	DEFINITION AND RECOMMENDATION	RISK PROFILE
5 Very High Confidence ASSIGNED	Codebase is mature, clean, and secure. No High or Critical vulnerabilities were found. Follows modern best practices with high test coverage and thoughtful design. Recommendation: code is highly secure at time of audit.	0 High/Critical findings and few Low/Medium severity findings.
4 High Confidence	Code is clean, well-structured, and adheres to best practices. Design patterns are sound and test coverage is strong. Recommendation: suitable for deployment after remediations; consider periodic review with changes.	0-1 High/Critical findings per engagement week and little to no Medium severity findings.
3 Moderate Confidence	Code is functional but shows gaps in defensive programming or test coverage. Recommendation: remediate the findings and re-review the changes before deployment.	Some High/Critical findings, with various Medium severity findings.
2 Low Confidence	Recurring issues suggest systemic gaps rather than isolated mistakes. Recommendation: remediate, then commission a follow-up review at a frozen commit.	Over 6 High and Critical findings across the engagement.
1 Very Low Confidence	The volume and severity of findings indicate the codebase is not ready for production. Recommendation: substantial rework, then a full re-review.	Over 12 High and Critical findings across the engagement.

§ 03

Scope & Classifications.

Guardian reviewed 3Jane LCC Leveraged Fund Helper Review in `https://github.com/3jane-protocol/moneymarket-contracts`. The scope includes 709 source lines across 2 files.

3.1 Severity Matrix

	IMP · HIGH	IMP · MED	IMP · LOW
LIK · HIGH	Critical	High	Medium
LIK · MED	High	Medium	Low
LIK · LOW	Medium	Low	Low

IMPACT · HIGH	Significant loss of assets, harm to a group of users, or core protocol functionality is disrupted.
IMPACT · MEDIUM	Small amount of funds can be lost, or ancillary functionality is affected. Users may experience reduced or delayed receipt.
IMPACT · LOW	Unexpected behavior in some functionality that is notable but does not meet the criteria for higher severity.
LIKELIHOOD · HIGH	Possible with reasonable assumptions that mimic on-chain conditions; cost is low compared to amount gained.
LIKELIHOOD · MEDIUM	Possible only in uncommon cases or requires a large amount of capital relative to disruption caused.
LIKELIHOOD · LOW	Unlikely to ever occur in production.

Files in Scope.

FILE	NSLOC	TOTAL
MAIN REVIEW		
src/lcc/interfaces/ILCCLeveragedFundHelper.sol	101	343
src/lcc/LCCLeveragedFundHelper.sol	576	840
REMEDIATION REVIEW		
src/lcc/interfaces/ILCCLeveragedFundHelper.sol	2	351
src/lcc/LCCLeveragedFundHelper.sol	30	871

§ 04

Findings.

Grouped by round. Visible IDs reset by severity within each round, so a finding is named the same way in the report as it was in the channel it was written in. Status is where each finding ended up; its own page says how.

ID	TITLE	CATEGORY	SEVERITY	STATUS	PAGE
MAIN REVIEW					
I-01	A skipped permit stays replayable until its deadline	Documentation	INFORMATIONAL	RESOLVED	p. 8
I-02	Consider an award-share burn cap for takeAuction margin recycling	economic	INFORMATIONAL	RESOLVED	p. 10
I-03	maxUnwindable can count a Morpho market the helper cannot unwind	Documentation	INFORMATIONAL	RESOLVED	p. 12

INFORMATIONAL I-01 RESOLVED

A skipped permit stays replayable until its deadline

CATEGORY	Documentation
LOCATION	`src/lcc/interfaces/ILCCLeveragedFundHelper.sol:385`
REVIEW	Main Review

DESCRIPTION

`fundWithSignatures` does not submit the USDC permit when the contribution is zero, and does not submit the margin permit when no margin is used. The signature is still in the transaction.

The notice says signatures the helper does not apply "are ignored and may be zeroed." That allows an empty signature. It does not say what a non-zero leftover does. The signature stays valid until its own deadline. Anyone can submit it once. That spends the nonce, so the same signature cannot be used again, and it sets the caller's allowance to the helper to the signed value. A later `approve` can raise it. Until the deadline, that old signature can still fire once and put the allowance back.

The sentence about a third party submitting the same signature first is a different case. It covers a permit the helper does try to submit. If that permit was already applied and the allowance covers the pull, the helper tolerates the revert. A permit the helper never submits is not that case.

RECOMMENDATION

State that a non-zero signature the helper does not submit stays replayable until its deadline. The caller should zero that field or give it a short deadline. Do not submit every attached permit.

RESOLUTION

Status: Resolved.

3Jane Team: Acknowledged and fixed in documentation (a21efbc8). The `fundWithSignatures` notice now states that a signature the call does not apply is not submitted and stays unused: anyone can submit a nonzero one once until its deadline, restoring the allowance or Morpho authorization it signs for the helper. This covers the USDC permit on a zero contribution, the margin permit without margin, and the Morpho authorization when the caller is

I-01 A skipped permit stays replayable until its deadline CONTINUED

already authorized. The helper pulls and borrows only for `msg.sender`, so this serves only the caller's own later calls; callers should zero an unneeded signature or give it a short deadline. The front-running sentence is now scoped to submitted signatures.

https://github.com/3jane-protocol/moneymarket-contracts/commit/a21efbc820ca8b7fb656d818df1b40d24317cf4f

INFORMATIONAL I-02

RESOLVED

Consider an award-share burn cap for takeAuction margin recycling

CATEGORY	economic
LOCATION	packages/moneymarket-contracts/src/lcc/LCCLeveragedFundHelper.sol:_runTake:L573-L593
REVIEW	Main Review

DESCRIPTION

takeAuction withdraws the requested USDC from newly awarded margin shares without a caller-defined share-burn cap; minMarginAward protects the gross award, not the retained remainder. Unlike fund, it passes type(uint256).max as the withdrawal's maxShares. Both fork tests pass, demonstrating the API difference using a bespoke ERC4626 with explicit loss/recovery methods. They do not establish a reachable exploit against current intended margin assets. The behavior matches the documented interface, and conversion is fair value at execution.

Impact: No current supported-asset exploit, execution-time value loss or violated existing bound is established. With a future declining-rate USDC margin asset, callers could not independently bound shares sold for their fixed-USDC contribution. Burns remain bounded by the newly received award; the demonstrated opportunity cost additionally assumes subsequent recovery.

RECOMMENDATION

Document that marginAssets requests a fixed-USDC sale at the execution rate and minMarginAward is gross. Before supporting declining-rate USDC margin assets, consider a scaled maxMarginShares or minimum retained-award bound.

PROOF OF CONCEPT https://github.com/GuardianOrg/SentryRepo-3Jane-LCC-Leveraged-Fund-Helper-Review--Round-1-01a15963ab9077d0afeee62d107cbbc9/blob/refinery/poc-01a15963-c0c0-781b-b5ea-ccdfb72b1091/test/refinery-poc/consider-an-award-share-burn-cap-for-takeauction-margin-recy-1fb95e9c.sol

RESOLUTION

Status: Resolved.

3Jane Team: Acknowledged; documented in 4d7eaf79. The TakeParams docs now state that a nonzero marginAssets is a fixed-USDC withdrawal from the award at the margin asset's conversion rate at execution, with shares burned bounded only by the award received in that take (MarginSharesUnavailable) and no caller-set share bound, and that minMarginAward bounds the

I-02 Consider an award-share burn cap for takeAuction margin recycling CONTINUED

gross award, not the remainder forwarded to the caller. Margin sourcing requires a margin asset whose asset() is USDC (LCCLeveragedFundHelper.sol:899-907), which today is waEthUSDC only. Before admitting a USDC margin asset whose rate can decline, we will revisit a scaled maxMarginShares or retained-award bound.

`https://github.com/3jane-protocol/moneymarket-contracts/commit/4d7eaf79e4260835cf08eee689b9f1010ab3b745`

INFORMATIONAL I-03

RESOLVED

maxUnwindable can count a Morpho market the helper cannot unwind

CATEGORY	Documentation
LOCATION	`src/lcc/LCCLeveragedFundHelper.sol:278`
REVIEW	Main Review

DESCRIPTION

`maxUnwindable` returns how many shares a user can unwind in a market. While the USD3l cooldown is on, that amount is the smaller of the user's Morpho collateral and the helper's matured cooldown shares for that market id. A market the helper never used has an empty book, so the result is zero.

While the cooldown is waived, the read skips the book and returns the user's whole Morpho collateral for the id it was given. The waiver is a USD3l shutdown, or a cooldown length of zero. This deployment cannot set the length to zero. Shutdown is the switch, and only USD3l management or the emergency admin can turn it on.

`unwind` rejects a market whose loan token is not USDC or whose collateral token is not USD3l. The read does not make that check. After shutdown, an id where the user has some other Morpho position returns a positive number, and `unwind` of that market reverts.

The notice calls the result a ceiling by the cooldown gate and live collateral. It says USD3's withdraw limit and Morpho's health check are not included. It does not promise that `unwind` will succeed, and it does not say the id must be a USDC/USD3l market. No cooldown credit is created, and the wrong market withdraws nothing.

RECOMMENDATION

Document that `marketId` must be a market on the pinned Morpho whose loan token is USDC and whose collateral token is USD3l. While the cooldown is waived, any other id can return that position's collateral even though `unwind` reverts. Alternatively, read the Morpho market data from the market id and return zero for any other pair.

RESOLUTION

Status: Resolved.

I-03 maxUnwindable can count a Morpho market the helper cannot unwind CONTINUED

3Jane Team: Acknowledged; documented in 4d7eaf79. The maxUnwindable notice now states that marketId is meant to be a market on the pinned Morpho lending USDC against USD3l (the markets unwind accepts), which the view does not check: while the USD3l cooldown is waived (shutdown or zero cooldownDuration) it returns the user's whole collateral for any market id, including one unwind rejects, and outside a waiver a market with no booked cooldowns returns zero. We kept the view unchanged rather than adding a Morpho market-params read, since an off-market result withdraws nothing and creates no cooldown credit.

`https://github.com/3jane-protocol/moneymarket-contracts/commit/4d7eaf79e4260835cf08eee689b9f1010ab3b745`

§ 05

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 Guardian 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. Guardian's position is that each company and individual are responsible for their own due diligence and continuous security. Guardian'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 Guardian is 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, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

§ 06

About Guardian.

Guardian is a smart contract security firm. Every engagement is worked by two independent teams competing on the same scope, each with its own tooling and methods, so no single reviewer's blind spot survives.

$45B+
DIGITAL ASSETS UNDER REVIEW

200+
ENGAGEMENTS COMPLETED

500+
CRITICAL & HIGH FINDINGS REPORTED

Reach us at audits@guardianaudits.com, or read more at **guardian.security**.

This report and its findings are also published at guardian.security/portfolio/3jane/lcc-leveraged-fund-helper-review-2026-11-11.