

ABRACADABRA MONEY · SECURITY REVIEW

Abracadabra Money Kodiak MIM-HONEY Island Market.

A security review of Kodiak MIM-HONEY Island Market.

CLIENT	PROJECT	AUDITORS	AUDIT WINDOW	FINAL REPORT
Abracadabra Money	Kodiak MIM-HONEY Island Market	Guardian	February 2 – February 8, 2025	February 8, 2025

§ 00

Contents.

24 pages: overview, confidence ranking, scope, 16 findings, disclaimer, and about Guardian.

§ 01	Overview	P. 03
§ 02	Confidence Ranking	P. 04
§ 03	Scope & Classifications	P. 05
§ 04	Findings	P. 06
	Main Review · 16 findings	P. 07
§ 05	Disclaimer	P. 23
§ 06	About Guardian	P. 24

§ 01

Overview.

Guardian reviewed Kodiak MIM-HONEY Island Market across a single review round. The final report includes 3 Medium, 13 Low findings.

AUDIT FIRM

Guardian

METHODOLOGY

Smart contract · Manual · Main Review

CLIENT

Abracadabra Money

PROJECT

Kodiak MIM-HONEY Island Market

PREPARED BY

Guardian

AUDIT WINDOW

February 2 – February 8, 2025

FINAL REPORT

February 8, 2025

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 Findings at a Glance

SEVERITY	TOTAL	PENDING	ACK	PARTIAL	RESOLVED
Critical	0	0	0	0	0
High	0	0	0	0	0
Medium	3	0	1	0	2
Low	13	0	2	0	11
Informational	0	0	0	0	0
All severities	16	0	3	0	13

§ 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 Kodiak MIM-HONEY Island Market.

IN SCOPE

./script/MagicKodiakMimHoney.s.sol

./src/harvesters/MagicKodiakVaultHarvester.sol

./src/tokens/MagicKodiakVault.sol

./src/oracles/aggregators/KodiakIslandAggregator.sol

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.

§ 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					
M-01	Harvest DoS For Zero Reward Tokens	DoS	MEDIUM	RESOLVED	p. 7
M-02	Reward Distributions Can Be Sandwiched	Sandwich Attack	MEDIUM	ACKNOWLEDGED	p. 8
M-03	Harvester Is Not Set As Operator	Configuration	MEDIUM	RESOLVED	p. 9
L-01	Vault Address Points To Mock	Warning	LOW	RESOLVED	p. 10
L-02	Dummy Honey-USD Price Feed	Warning	LOW	RESOLVED	p. 11
L-03	Repetitive Cauldron Naming	Warning	LOW	RESOLVED	p. 12
L-04	Reward Vault Unwhitelisted DoS	Warning	LOW	ACKNOWLEDGED	p. 13
L-05	Lacking Staleness Checks In KodiakIslandAggregator	Validation	LOW	ACKNOWLEDGED	p. 14
L-06	Lacking Aggregator Validations	Validation	LOW	RESOLVED	p. 15
L-07	Incorrect Decimals Assumption	Warning	LOW	RESOLVED	p. 16
L-08	LP Token Rewards Cannot Be Harvested	Warning	LOW	RESOLVED	p. 17
L-09	Dangling Approvals If Reward Tokens Are Removed	Warning	LOW	RESOLVED	p. 18
L-10	New Reward Tokens Require More Approvals	Warning	LOW	RESOLVED	p. 19
L-11	Unnecessary SwapData Fields	Optimization	LOW	RESOLVED	p. 20
L-12	Unnecessary Import	Suggestion	LOW	RESOLVED	p. 21
L-13	Lacking Fee Configuration	Configuration	LOW	RESOLVED	p. 22

MEDIUM **M-01**

RESOLVED

Harvest DoS For Zero Reward Tokens

CATEGORY DoS
LOCATION MagicKodiakVault.sol
REVIEW Main Review

— DESCRIPTION

In the MagicKodiakVault `harvest` function the `transfer` function is attempted to be used for every reward token. However if there are currently zero rewards for any individual reward token that reverts on zero transfers then the `harvest` function will revert and prevent the claiming of all rewards.

— RECOMMENDATION

Consider only transferring rewards for the reward token if there is a nonzero contract balance for that reward.

— RESOLUTION

Status: Resolved.

MEDIUM **M-02**

ACKNOWLEDGED

Reward Distributions Can Be Sandwiched

CATEGORY Sandwich Attack
LOCATION MagicKodiakVault.sol: 61
REVIEW Main Review

— DESCRIPTION

In the `MagicKodiakVault` the `distributeRewards` function increments the `_totalAssets` of the vault in a stepwise manner. This can allow for sandwich attack arbitrage opportunities when the reward amount is large.

A malicious actor can see a reward distribution in the mempool and frontrun/backrun it to immediately gain from the distributed amount without holding exposure to the vault for the same time as the users who provided the capital to generate the rewards.

Furthermore, even if a private RPC is used a preliminary `harvest` transaction may be used as an indicator that `distributeRewards` is about to be called after swapping reward tokens to the target asset. This would allow a malicious actor to frontrun and backrun reward distributions without necessary knowledge of the `distributeRewards` transaction in the mempool, however this is not a guaranteed profit.

— RECOMMENDATION

Be sure to use a private RPC to call the `distributeRewards` function. Otherwise be sure to keep the reward distributions small and infrequent.

A more long term solution is to consider refactoring the reward distribution logic so that the `totalAssets` increment is gradual rather than instantaneous.

— RESOLUTION

Status: Acknowledged.

MEDIUM **M-03**

RESOLVED

Harvester Is Not Set As Operator

CATEGORY Configuration

LOCATION Global

REVIEW Main Review

DESCRIPTION

The Harvester address `0xB76403D52CA4D4f5C791859d93553E14742cc9d1` is currently not assigned as an operator for the vault proxy address `0x548eAf30BAD1B4f697fBAC88712fc8148Dc0aA75`. Instead the deployer address `0xfb3485c2e209a5cfbdc1447674256578f1a80ee3` is currently assigned as an operator.

RECOMMENDATION

Add the operator role to the harvester and consider removing the operator role from the deployer.

RESOLUTION

Status: Resolved.

LOW **L-01**

RESOLVED

Vault Address Points To Mock

CATEGORY Warning
LOCATION bera.json
REVIEW Main Review

— DESCRIPTION

Within bera.json, address `0x7FC70DfE186b1eF1B792d1c42C3b7De09F4a68a7` does not point to the Infrared MIM-Honey vault but instead to a mock contract.

— RECOMMENDATION

Ensure the address is updated prior to MagicKodiakVault deployment.

— RESOLUTION

Status: Resolved.

LOW **L-02**

RESOLVED

Dummy Honey-USD Price Feed

CATEGORY	Warning
LOCATION	script/MagicKodiakMimHoney.s.sol
REVIEW	Main Review

— DESCRIPTION

Currently the `redstoneHoneyAggregator` is set to an aggregator that will always return a \ \$1 fixed price which will affect the calculated price of the collateral.

— RECOMMENDATION

Ensure the aggregator is updated to the proper Redstone address for live deployment.

— RESOLUTION

Status: Resolved.

LOW **L-03**

RESOLVED

Repetitive Cauldron Naming

CATEGORY Warning
LOCATION MagicKodiakMimHoneyScript
REVIEW Main Review

DESCRIPTION

"MagicKodiakCauldron" is supplied as the `name` to function `_deployCauldron`. Consequently, when the cauldron is actually deployed, it's deployment name will be `Cauldron_MagicKodiakCauldron`.

RECOMMENDATION

Consider adjusting the name as necessary such that it adheres to Abracadabra's standards.

RESOLUTION

Status: Resolved.

LOW **L-04**

ACKNOWLEDGED

Reward Vault Unwhitelisted DoS

CATEGORY	Warning
LOCATION	MagicKodiakVault.sol
REVIEW	Main Review

— DESCRIPTION

Infrared vaults rely on being whitelisted so that they can interact with the Berachain reward vaults during staking and unstaking. However, `setVaultWhitelistedStatus` can be used by governance to revoke whitelisted status (see [doc](https://docs.berachain.com/learn/pol/rewardvaults)), which would ultimately prevent the ability for new stakes to occur into the MagicKodiakVault.

— RECOMMENDATION

Clearly document this risk.

— RESOLUTION

Status: Acknowledged.

LOW **L-05**

ACKNOWLEDGED

Lacking Staleness Checks In KodiakIslandAggregator

CATEGORY Validation
LOCATION KodiakIslandAggregator.sol
REVIEW Main Review

— DESCRIPTION

The `KodiakIslandAggregator` contract performs no staleness checks on the underlying aggregator feeds. Therefore this aggregator is at risk of reporting inaccurate prices when one of the underlying feeds is outdated.

— RECOMMENDATION

Be aware of this issue and consider adding staleness validation controls depending on if a feed implements staleness checks.

— RESOLUTION

Status: Acknowledged.

LOW **L-06**

RESOLVED

Lacking Aggregator Validations

CATEGORY	Validation
LOCATION	KodiakIslandAggregator.sol
REVIEW	Main Review

DESCRIPTION

The `KodiakIslandAggregator` contract is missing several validations which can be crucial in avoiding a mispricing or manipulation in edge case scenarios.

Firstly, the `latestAnswer` function lacks validation that requires the `totalSupply` to be above a minimal threshold to avoid unexpected low precision manipulations with small amounts of collateral. It also lacks validation which would prevent a divide by zero revert when the `totalSupply` is zero.

The Maker DAO `GUniLPOracle` [validates this by ensuring that the](<https://github.com/makerdao/univ3-lp-oracle/blob/0d25676e0956fc3a5359431c708106547c4fef26/src/GUniLPOracle.sol#L260>)[`totalSupply`] (<https://github.com/makerdao/univ3-lp-oracle/blob/0d25676e0956fc3a5359431c708106547c4fef26/src/GUniLPOracle.sol#L260>)[is above 1e9] (<https://github.com/makerdao/univ3-lp-oracle/blob/0d25676e0956fc3a5359431c708106547c4fef26/src/GUniLPOracle.sol#L260>).

Secondly there are several casts performed which do not include safety reverts if an overflow would have occurred. The

`_getSqrtPriceX96` function does not include any reverts if the uint160 cast would overflow, though such an overflow would be extremely unlikely. Additionally, there is no validation that the result of `totalValue / totalSupply` does not overflow the `int256` result. Both scenarios are unlikely, however should be validated against to avoid any unexpected manipulations.

RESOLUTION

Status: Resolved.

LOW **L-07**

RESOLVED

Incorrect Decimals Assumption

CATEGORY	Warning
LOCATION	KodiakIslandAggregator.sol
REVIEW	Main Review

DESCRIPTION

The `KodiakIslandAggregator` contract reports a decimals of the `IERC20Metadata(address(island)).decimals()` result, however this is incorrect when the `IERC20Metadata(address(island)).decimals()` result is not equal to 18.

Consider the existing case, when the island decimals are 18:

- `normalizedReserve0` and `normalizedReserve1` are 1e18 decimals
- `normalizedPriceFeed0` and `normalizedPriceFeed1` are 1e18 decimals
- The resulting `totalValue` is 1e36 decimals
- Assume `totalValue` is \$1, e.g. 1e36
- Assume `totalSupply` is 1 token, e.g. 1e18
- The resulting price is $1e36/1e18 == 1e18$ which is correctly \$1 with 18 decimals

However for a case where the island decimals are 8:

- `normalizedReserve0` and `normalizedReserve1` are 1e18 decimals
- `normalizedPriceFeed0` and `normalizedPriceFeed1` are 1e18 decimals
- The resulting `totalValue` is 1e36 decimals
- Assume `totalValue` is \$1, e.g. 1e36
- Assume `totalSupply` is 1 token, e.g. 1e8
- The resulting price is $1e36/1e8 == 1e28$ which is incorrectly reporting a very large price for an 8 decimal feed

RECOMMENDATION

Instead of assigning the decimals for the feed as the `IERC20Metadata(address(island)).decimals()` result, either hardcode the decimals as 18 and document that this aggregator is incompatible with any other island decimals. Or assign the decimals as: `36 - IERC20Metadata(address(island)).decimals()`.

RESOLUTION

Status: Resolved.

LOW **L-08**

RESOLVED

LP Token Rewards Cannot Be Harvested

CATEGORY	Warning
LOCATION	MagicKodiakVaultHarvester.sol
REVIEW	Main Review

— DESCRIPTION

In the `run` function the LP token balance of the harvester contract before the `addLiquidity` call is removed from the amount which can be rewarded to the vault.

This prevents LP token rewards that were harvested from being distributable back to the `MagicKodiakVault`. However it is unlikely that LP tokens will be supported as a direct reward token.

— RECOMMENDATION

This finding simply serves as a warning since it is unlikely that the LP token will be a reward token for the Infrared vault. However if it is then this logic should be amended in a new harvester contract. Additionally, consider adding a token rescue function in the event that LP tokens are sent to this contract either as a reward or by accident and cannot be recovered.

— RESOLUTION

Status: Resolved.

LOW **L-09**

RESOLVED

Dangling Approvals If Reward Tokens Are Removed

CATEGORY Warning
LOCATION MagicKodiakVaultHarvester.sol
REVIEW Main Review

DESCRIPTION

In the `setExchangeRouter` function the approvals are removed for all existing reward tokens from the previous `exchangeRouter` contract.
However if a reward is removed from the infrared vault with the `removeReward` function then this token approval will not be removed from the previous `exchangeRouter` upon the next `exchangeRouter` update.

RECOMMENDATION

Consider adding an additional function to remove targeted approvals from an existing or past `exchangeRouter` contract for cases where the reward tokens are updated.

RESOLUTION

Status: Resolved.

LOW **L-10**

RESOLVED

New Reward Tokens Require More Approvals

CATEGORY Warning
LOCATION MagicKodiakVaultHarvester.sol
REVIEW Main Review

DESCRIPTION

New reward tokens may be added to the Infrared vault with the `addReward` function. These reward tokens will require additional approvals to be made to the `exchangeRouter` contract.

RECOMMENDATION

Currently this can be achieved by using the `setExchangeRouter` function to set the same `exchangeRouter` implementation in order to approve all new tokens. Be aware that this scenario may arise.

RESOLUTION

Status: Resolved.

LOW **L-11**

RESOLVED

Unnecessary SwapData Fields

CATEGORY Optimization
LOCATION MagicKodiakVaultHarvester.sol
REVIEW Main Review

DESCRIPTION

**The **`SwapInfo` struct contains two unnecessary fields, the `token` and `amount` fields, which are not used in the implementation anywhere.

RECOMMENDATION

Remove these unnecessary fields from the `SwapInfo` struct.

RESOLUTION

Status: Resolved.

LOW **L-12**

RESOLVED

Unnecessary Import

CATEGORY Suggestion
LOCATION MagicKodiakVaultHarvester.sol
REVIEW Main Review

DESCRIPTION

The `IERC20` import in the `MagicKodiakVaultHarvester` contract is unused.

RECOMMENDATION

Consider removing this import.

RESOLUTION

Status: Resolved.

LOW L-13

RESOLVED

Lacking Fee Configuration

CATEGORY	Configuration
LOCATION	Global
REVIEW	Main Review

DESCRIPTION

In the deployed harvester contract the fee basis points and fee receiver have not been configured.

RECOMMENDATION

Consider if this is the expected state, and if not be sure to configure the fee amount and receiver as they are expected to be.

RESOLUTION

Status: Resolved.

§ 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.