

ABRACADABRA MONEY · SECURITY REVIEW

Abracadabra Money Berachain Cauldron.

A security review of Berachain Cauldron.

CLIENT

Abracadabra
Money

PROJECT

Berachain
Cauldron

AUDITORS

Zdravko Hristov,
0xCiphky .

AUDIT WINDOW

March 17 – March
19, 2025

FINAL REPORT

March 19, 2025

§ 00

Contents.

22 pages: overview, confidence ranking, scope, 14 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 · 14 findings	P. 07
§ 05	Disclaimer	P. 21
§ 06	About Guardian	P. 22

§ 01

Overview.

Guardian reviewed Berachain Cauldron across a single review round. The final report includes 2 Medium, 12 Low findings.

AUDIT FIRM

Guardian

METHODOLOGY

Smart contract · Manual · Main Review

CLIENT

Abracadabra Money

PROJECT

Berachain Cauldron

PREPARED BY

Zdravko Hristov, 0xCiphky .

AUDIT WINDOW

March 17 – March 19, 2025

FINAL REPORT

March 19, 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.

1VERY LOW
CONFIDENCE**2**LOW
CONFIDENCE**3**MODERATE
CONFIDENCE**4**HIGH
CONFIDENCE**5**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	2	0	2	0	0
 Low	12	0	12	0	0
 Informational	0	0	0	0	0
All severities	14	0	14	0	0

§ 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 Berachain Cauldron.

IN SCOPE

MagicBex.s.sol (overview that the deployment is ok, currently deployed)

added interfaces

" libraries

" aggregators

" tokens

" harvesters

" swappers

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

§ 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	Reward Distributions Can Be Sandwiched	Sandwich Attack	MEDIUM	ACKNOWLEDGED	p. 7
M-02	Reward Distributions Can Be Sandwiched	Sandwich Attack	MEDIUM	ACKNOWLEDGED	p. 8
L-01	BEX Vault Can Be Paused Causing A DoS	DoS	LOW	ACKNOWLEDGED	p. 9
L-02	MINIMUM_SHARE Requirement May Block Swaps	DoS	LOW	ACKNOWLEDGED	p. 10
L-03	Swapper Approvals Are Unrestricted	Warning	LOW	ACKNOWLEDGED	p. 11
L-04	Unset exchangeRouter Address In Harvester Contracts	Warning	LOW	ACKNOWLEDGED	p. 12
L-05	Potential Pending Rewards in MagicInfraredVault	Warning	LOW	ACKNOWLEDGED	p. 13
L-06	Vault Contribution May Not Be Rewarded	Warning	LOW	ACKNOWLEDGED	p. 14
L-07	Withdrawing Users Lose Unharvested Rewards	Warning	LOW	ACKNOWLEDGED	p. 15
L-08	New Reward Tokens Require More Approvals	Warning	LOW	ACKNOWLEDGED	p. 16
L-09	Incompatibility With fee-on-transfer Tokens	Warning	LOW	ACKNOWLEDGED	p. 17
L-10	Dangling Approvals If Reward Tokens Are Removed	Warning	LOW	ACKNOWLEDGED	p. 18
L-11	MagicInfraredVault's Harvest Logic Breaks if Reward Token Matches the Underlying Asset	Logical Error	LOW	ACKNOWLEDGED	p. 19
L-12	Unset exchangeRouter Address In Harvester Contracts	Warning	LOW	ACKNOWLEDGED	p. 20

MEDIUM **M-01**

ACKNOWLEDGED

Reward Distributions Can Be Sandwiched

CATEGORY Sandwich Attack
LOCATION MagicInfraredVault.sol
REVIEW Main Review

— DESCRIPTION

The `MagicInfraredVault`'s `distributeRewards` function increases the `_totalAssets` value immediately upon execution. This immediate increment creates an opportunity for sandwich attack arbitrage, especially when rewards are substantial. Attackers can monitor the mempool, identify pending reward distribution transactions, and frontrun or backrun them to profit instantly without holding the vault exposure equivalent to genuine vault depositors who provided capital to generate the rewards.

Even if a private RPC is used, preliminary transactions such as `harvest` might signal an impending reward distribution, indirectly informing attackers of upcoming profitable trades. While such indirect signals may not guarantee profit, they still significantly increase the likelihood of arbitrage opportunities.

— 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.

PROOF OF CONCEPT <https://gist.github.com/ZdravkoHr/b8b99e6c31c360202bb1765faa74258d>

— RESOLUTION

Status: Acknowledged.

MEDIUM **M-02**

ACKNOWLEDGED

Reward Distributions Can Be Sandwiched

CATEGORY Sandwich Attack
LOCATION MagicInfraredVault.sol
REVIEW Main Review

— DESCRIPTION

The `MagicInfraredVault`'s `distributeRewards` function increases the `_totalAssets` value immediately upon execution. This immediate increment creates an opportunity for sandwich attack arbitrage, especially when rewards are substantial. Attackers can monitor the mempool, identify pending reward distribution transactions, and frontrun or backrun them to profit instantly without holding the vault exposure equivalent to genuine vault depositors who provided capital to generate the rewards.

Even if a private RPC is used, preliminary transactions such as `harvest` might signal an impending reward distribution, indirectly informing attackers of upcoming profitable trades. While such indirect signals may not guarantee profit, they still significantly increase the likelihood of arbitrage opportunities.

— 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.

LOW **L-01**

ACKNOWLEDGED

BEX Vault Can Be Paused Causing A DoS

CATEGORY	DoS
LOCATION	BexLib.sol
REVIEW	Main Review

— DESCRIPTION

Within the BEX Vault contract, the `joinPool` function contains the `whenNotPaused` modifier. Consequently, whenever the Vault is paused, calls to `joinPool` revert, effectively blocking any functionality that relies on depositing tokens (joining) into the pool. However, the `exitPool` function has no such pause check and continues to work even if the BEX Vault is paused.

In the `MagicBexVaultLevSwapper` (particularly the `swap` function) and the `MagicBexVaultHarvester` (the `run` function), both rely on `BexLib.joinPool` to complete their flows:

- `MagicBexVaultLevSwapper.swap` needs to join the BEX pool after swapping MIM for underlying tokens.
- `MagicBexVaultHarvester.run` must join the BEX pool to compound yield after swapping reward tokens.

Because the BEX Vault can be paused at any time, any calls to `joinPool` will fail while paused. That means both `MagicBexVaultLevSwapper.swap` and `MagicBexVaultHarvester.run` revert under pause conditions, causing a Denial of Service for those functions.

— RECOMMENDATION

Document that if the [BEX Vault] (<https://berascan.com/address/0x4Be03f781C497A489E3cB0287833452cA9B9E80B>) is paused, `MagicInfraredVault` depositors will not be able to receive any yield.

— RESOLUTION

Status: Acknowledged.

LOW **L-02**

ACKNOWLEDGED

MINIMUM_SHARE Requirement May Block Swaps

CATEGORY DoS

LOCATION MagicBexVaultSwapper.sol: 49, MagicBexVaultLevSwapper.sol: 49

REVIEW Main Review

— DESCRIPTION

When one of the two swappers used in the liquidation function of the Cauldron receives the collateral tokens, it's withdrawing them from the `DegenBox`. When withdrawing, the `degenbox` has a requirement that the shares after the withdrawal must be at least `MINIMUM_SHARE_BALANCE` (or 1000), unless it's a full withdrawal. Because of this, there may be some situations where the swapper is trying to withdraw an amount which triggers this revert and result in a failed liquidation.

— RECOMMENDATION

The `swapper` parameter in `CauldronV4.liquidate` is permissionless, so if a liquidation fails because of this, a different swapper can be provided. Therefore, it's enough to document the issue as a potential behavior.

— RESOLUTION

Status: Acknowledged.

LOW **L-03**

ACKNOWLEDGED

Swapper Approvals Are Unrestricted

CATEGORY Warning
LOCATION MagicBexVaultSwapper.sol; MagicBexVaultLevSwapper.sol
REVIEW Main Review

— DESCRIPTION

The `MagicBexVaultSwapper.swap` and `MagicBexVaultLevSwapper.swap` functions are permissionless which allows giving an approval to whatever to address the callers wants.

— RECOMMENDATION

Document this.

— RESOLUTION

Status: Acknowledged.

LOW **L-04**

ACKNOWLEDGED

Unset exchangeRouter Address In Harvester Contracts

CATEGORY	Warning
LOCATION	MagicBexVaultHarvester.sol, MagicKodiakVaultHarvester.sol
REVIEW	Main Review

— DESCRIPTION

The `exchangeRouter` address, intended for token swaps during the reward harvesting process, is currently unset (zero address) in the harvester contracts. An unset or incorrect `exchangeRouter` address will prevent swaps from executing correctly, causing reward harvesting operations to fail disabling the rewards distribution.

— RECOMMENDATION

Ensure the `exchangeRouter` address is properly set to a valid router.

— RESOLUTION

Status: Acknowledged.

LOW **L-05**

ACKNOWLEDGED

Potential Pending Rewards in MagicInfraredVault

CATEGORY	Warning
LOCATION	MagicInfraredVault.sol
REVIEW	Main Review

— DESCRIPTION

The run function in the `MagicBexVaultHarvester` can be called by the owner or an address with the operator role to harvest rewards from the staking contract and distribute them to the vault. However, users can also call the `getRewardForUser` function in the `InfraredVault` contract, specifying the `MagicInfraredVault` contract's address. This leads to an unfinished sequence where rewards are transferred to the `MagicInfraredVault` contract but are not harvested until run is called next on the `MagicBexVaultHarvester`. As a result, these rewards may not be properly accounted for when determining the swap parameters.

— RECOMMENDATION

Ensure that the run function parameters account for any pending rewards in the `MagicInfraredVault` contract to maintain accurate swap calculations.

— RESOLUTION

Status: Acknowledged.

LOW **L-06**

ACKNOWLEDGED

Vault Contribution May Not Be Rewarded

CATEGORY	Warning
LOCATION	MagicInfraredVault.sol
REVIEW	Main Review

— DESCRIPTION

When the `InfraredVault` accumulates rewards in the `RewardVault`, the `InfraredVault.notifyRewardAmount()` function is called to update the reward rate. Because of this design some users may not be properly rewarded for their contributions.

For example,

- `rewardDuration` in `RewardVault` is 7 days
- only staker in `RewardVault` is `InfraredVault` and only staker in `InfraredVault` is `MagicInfraredVault`
- user A and user B deposit equal amount of LP in the `MagicInfraredVault`
- These are staked in the `RewardVault` on the behalf of `InfraredVault`
- The `RewardVault` receives 600 BGT in rewards
- After one week passes the rewards are claimed and notified to the `InfraredVault`
- Reward distribution in `InfraredVault` has just began
- If user B leaves the MagicVault they won't receive any additional rewards (even if harvest is called before the withdraw) even if he contributed for them

— RECOMMENDATION

In the above example, the user loses rewards for the whole rewardPeriod in the `RewardVault`. There is no perfect solution to this as it's the way `Infrared` operates.

Calling `harvest()` frequently is one way to reduce the losses because each `harvest()` claims the rewards from the `RewardVault` to `InfraredVault`.

You can do that and also document this behavior so users are aware of it.

— RESOLUTION

Status: Acknowledged.

LOW **L-07**

ACKNOWLEDGED

Withdrawing Users Lose Unharvested Rewards

CATEGORY Warning
LOCATION MagicInfraredVault.sol
REVIEW Main Review

— DESCRIPTION

Withdrawers lose their rewards which have accumulated between the previous `Harvester.run` call and the current moment because at the time of the next `Harvester.run` call, their balance will be reduced.

— RECOMMENDATION

Make sure to harvest the rewards frequently.

— RESOLUTION

Status: Acknowledged.

LOW **L-08**

ACKNOWLEDGED

New Reward Tokens Require More Approvals

CATEGORY	Warning
LOCATION	MagicBexVaultHarvester.sol, 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: Acknowledged.

LOW **L-09**

ACKNOWLEDGED

Incompatibility With fee-on-transfer Tokens

CATEGORY Warning
LOCATION Global
REVIEW Main Review

— DESCRIPTION

All the contracts assume that tokens do not apply transfer fees or rebases. Tokens that implement fees upon transfers (commonly referred to as "fee-on-transfer tokens") or other mechanics like rebasing would lead to incorrect balance calculations, breaking core functionality.

— RECOMMENDATION

Explicitly document that fee-on-transfer tokens are not supported or enforce restrictions to disallow fee-on-transfer and rebasing tokens.

— RESOLUTION

Status: Acknowledged.

LOW **L-10**

ACKNOWLEDGED

Dangling Approvals If Reward Tokens Are Removed

CATEGORY Warning

LOCATION MagicBexVaultHarvester.sol, 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: Acknowledged.

LOW **L-11**

ACKNOWLEDGED

MagicInfraredVault's Harvest Logic Breaks if Reward Token Matches the Underlying Asset

CATEGORY Logical Error
LOCATION MagicInfraredVault.sol
REVIEW Main Review

— DESCRIPTION

Within the `MagicInfraredVault` contract, the `harvest` function collects reward tokens from `staking.getAllRewardTokens`, swaps them via aggregator calls, and then re-deposits them as the Vault's underlying asset (`_asset`). If the staking contract unexpectedly returns a reward token that is exactly the same address as `_asset`, the assets would be sent to the harvester and they would be wrongly seen as rewards as in the `distributeRewards` call the following update would be performed: `_totalAssets += amount;`. The code does not check that each reward token differs from the main `_asset`. If `_asset` appears among the reward tokens, the logic within the `harvest` and `distributeRewards` functions will malfunction, causing broken harvesting flows and incorrect accounting.

— RECOMMENDATION

`MagicInfraredVault` should explicitly validate that none of the tokens in `staking.getAllRewardTokens` equals `_asset`.

— RESOLUTION

Status: Acknowledged.

LOW **L-12**

ACKNOWLEDGED

Unset exchangeRouter Address In Harvester Contracts

CATEGORY	Warning
LOCATION	MagicBexVaultHarvester.sol, MagicKodiakVaultHarvester.sol
REVIEW	Main Review

— DESCRIPTION

The `exchangeRouter` address, intended for token swaps during the reward harvesting process, is currently unset (zero address) in the harvester contracts. An unset or incorrect `exchangeRouter` address will prevent swaps from executing correctly, causing reward harvesting operations to fail disabling the rewards distribution.

— RECOMMENDATION

Ensure the `exchangeRouter` address is properly set to a valid router.

— RESOLUTION

Status: Acknowledged.

§ 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.