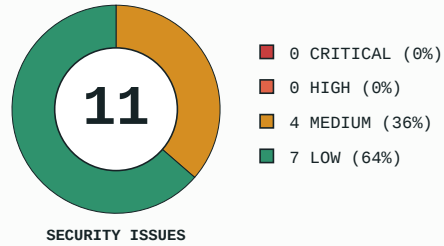


// AI Security Analysis & Human Triaging

Project Name	Bench
Project Type	Opportunity Market
Codebase	Solana Anchor + Arcium
Manually Triaged Findings	9/19 (47%)
Repository	Bench-games/opportunity-markets
Commit	2dc1caf4b533
Report By	AI Security Engine + human validation

// SEVERITY DISTRIBUTION



// EXECUTIVE SUMMARY

The reviewed on-chain program has no confirmed Critical or High severity findings after validation. The main security risks are liveness and accounting edge cases around reveal/finalization, reward calculation, and lifecycle cleanup. Principal recovery is generally preserved, but several paths can deny rewards, strand rent, or make operational cleanup difficult.

// VALIDATED FINDINGS

ID	TITLE	SEVERITY	CATEGORY
M-01	Resolved market is permanently stuck if no winning option has positive finalized score	MEDIUM	LIFECYCLE
M-02	<code>reveal_stake</code> overwrites an in-flight reveal computation	MEDIUM	LIFECYCLE
M-03	<code>reveal_stake</code> is still callable after <code>reveal_ended</code>	MEDIUM	LIFECYCLE
M-04	<code>claim_rewards</code> reward numerator can overflow <code>u128</code>	MEDIUM	ARITHMETIC
L-01	Claim/withdraw destinations are unconstrained	LOW	ACCESS CONTROL
L-02	Zero-net winning stakes can forfeit rewards if their option is closed first	LOW	LIFECYCLE
L-03	Pre-open sponsor deposits are unrecoverable if the market never opens	LOW	ACCOUNTING
L-04	Expired, unresolved markets cannot close option accounts	LOW	LIFECYCLE
L-05	Durable authorities can be set to unusable/default addresses in one step	LOW	ACCESS CONTROL
L-06	<code>resolve_market</code> and <code>withdraw_reward</code> deadlines overlap at exactly <code>select_deadline</code>	LOW	LIFECYCLE
L-07	Winning-option rent is stranded forever if any finalized winner never claims	LOW	ACCOUNTING

M-01

Resolved market is permanently stuck if no winning option has positive finalized score

SEVERITY

MEDIUM

CATEGORY

Lifecycle

DESCRIPTION

`end_reveal_period` requires `market.winning_option_active_bp > 0`, but that value only increases when a winning option has a finalized positive score. If the market resolves to winning option(s) that have no finalized stakes or only zero-score stakes, `reveal_ended` can never be set.

Principal remains recoverable through `unstake`, but reward claims, creator-fee refunds, sponsor reward settlement, and account cleanup can remain blocked for the affected market.

PRACTICAL IMPACT

A low-activity resolved market can get stuck in limbo. Users can recover principal, but rewards, refunds, and cleanup may remain unavailable.

LOCATION

[programs/opportunity_market/src/instructions/end_reveal_period.rs:13](#)
[programs/opportunity_market/src/instructions/finalize_reveal_stake.rs:109](#)
[programs/opportunity_market/src/score.rs:41](#)

RECOMMENDATION

Add an explicit no-positive-winner path that allows the reveal period to end after the normal reveal window, while preserving principal recovery and appropriate fee/refund behavior. Add a regression test where the selected winning option has `total_score == 0`.

M-02

`reveal_stake` overwrites an in-flight reveal computation

SEVERITY

MEDIUM

CATEGORY

Lifecycle

DESCRIPTION

`reveal_stake` is permissionless and overwrites `pending_reveal_computation` without checking whether a reveal computation is already in flight. A griever can repeatedly queue newer reveal computations for a victim stake, making earlier valid callbacks fail as stale.

The attacker cannot forge the selected option because the reveal args are built from stored stake account data. The impact is reward-liveness griefing: if the attack is sustained until the reveal period ends, a winning stake may fail to finalize and lose reward eligibility.

PRACTICAL IMPACT

A griever can make winning users miss reward eligibility by repeatedly disrupting their reveal. This can damage user trust in reward fairness.

LOCATION

[programs/opportunity_market/src/instructions/reveal_stake.rs:24](#)
[programs/opportunity_market/src/instructions/reveal_stake.rs:83](#)
[programs/opportunity_market/src/instructions/reveal_stake.rs:151](#)
[programs/opportunity_market/src/instructions/stake.rs:52](#)

RECOMMENDATION

Reject new reveal attempts while `pending_reveal_computation.is_some()`, but pair that with a retry timeout or owner-authorized override so genuinely stuck computations can be recovered. Also reject reveals after `reveal_ended`.

M-03

reveal_stake is still callable after reveal_ended

SEVERITY

MEDIUM

CATEGORY

Lifecycle

DESCRIPTION

The documentation states that once `end_reveal_period` runs, both `reveal_stake` and `finalize_reveal_stake` are no longer callable. The code blocks `finalize_reveal_stake` after `reveal_ended`, but `reveal_stake` only checks that the market is resolved.

After the reveal period has ended, a permissionless caller can still reveal an otherwise unrevealed stake. The stake cannot finalize or earn rewards, but the late reveal can disclose the user's selected option and mutate the stake account state.

PRACTICAL IMPACT

Users' choices can still be exposed after the reveal phase is supposed to be closed. Funds are not stolen, but the promised privacy window is weaker than advertised.

LOCATION

[programs/opportunity_market/src/instructions/reveal_stake.rs:75](#)
[programs/opportunity_market/src/instructions/finalize_reveal_stake.rs:56](#)
[docs/README.md:142](#)

RECOMMENDATION

Add `require(!(market.reveal_ended, ErrorCode::RevealPeriodEnded))` to `reveal_stake`, and add a callback-side guard as defense in depth.

M-04

claim_rewards reward numerator can overflow u128

SEVERITY

MEDIUM

CATEGORY

Arithmetic

DESCRIPTION

`claim_rewards` forms the full numerator `user_score * market.reward_amount * option.reward_bp` before division. With large enough token amounts, the checked multiplication can overflow `u128`, making the reward claim revert even though the final payout would be in range.

Because the claim reverts before `rewards_claimed = true`, the affected winner cannot complete reward claim or cleanly close the stake account. Net principal remains recoverable via `unstake`, but the reward and stake-account rent can be permanently blocked.

PRACTICAL IMPACT

Large legitimate winners may be unable to claim rewards. This mostly affects high-value markets or high-supply tokens.

LOCATION

[programs/opportunity_market/src/instructions/claim_rewards.rs:133](#)
[programs/opportunity_market/src/instructions/close_stake_account.rs:82](#)
[programs/opportunity_market/src/score.rs:83](#)

RECOMMENDATION

Reorder the arithmetic to divide earlier, or use a wider integer type. Add whale-scale regression tests where both reward pool and user score are near the reachable upper bound.

L-01

Claim/withdraw destinations are unconstrained

SEVERITY LOW	CATEGORY Access control
DESCRIPTION Claim and withdraw destinations only require the correct token mint and token program. They do not require the destination to be owned by the expected signer, and they do not reject the market vault ATA itself as the destination. This creates two bounded issues. First, a self-transfer to the vault can succeed while program accounting is cleared, leaving value in the vault with no remaining claim path. Second, a malicious UI or transaction builder can route an authorized user's payout to an arbitrary same-mint token account.	PRACTICAL IMPACT Fee and sponsor payouts can be misdirected by bad transaction construction or malicious UIs. In some cases, funds can remain stuck in the vault with accounting marked as claimed.
LOCATION programs/opportunity_market/src/instructions/claim_fees.rs:39 programs/opportunity_market/src/instructions/claim_creator_fees.rs:37 programs/opportunity_market/src/instructions/withdraw_reward.rs:42	RECOMMENDATION Require <code>destination.key() != market_token_ata.key()</code> . If arbitrary destinations are not intended, also constrain destination authority to the expected signer or configured treasury account.

L-02

Zero-net winning stakes can forfeit rewards if their option is closed first

SEVERITY LOW	CATEGORY Lifecycle
DESCRIPTION In extreme fee configurations, a winning stake can have <code>stake_account.amount == 0</code> after fees while still having a positive score and reward entitlement. <code>close_option_account</code> uses <code>option.unclaimed_stake == 0</code> as a signal that the option is no longer needed, but zero-net winners add <code>0</code> to <code>unclaimed_stake</code> . The winning option account can therefore be closed before the zero-net winner claims rewards. Once the option account is closed, <code>claim_rewards</code> can no longer load it and the reward is forfeited.	PRACTICAL IMPACT In unusual high-fee markets, a winner can lose a valid reward because the option account is closed too early. This is mostly a niche configuration risk.
LOCATION programs/opportunity_market/src/instructions/finalize_reveal_stake.rs:67 programs/opportunity_market/src/instructions/finalize_reveal_stake.rs:83 programs/opportunity_market/src/instructions/close_option_account.rs:31 programs/opportunity_market/src/instructions/claim_rewards.rs:34 programs/opportunity_market/src/instructions/close_stake_account.rs:78	RECOMMENDATION Track reward-claim liveness separately from net principal, or block winning option closure while any finalized winning stake may still have a reward claim.

L-03

Pre-open sponsor deposits are unrecoverable if the market never opens

SEVERITY

LOW

CATEGORY

Accounting

DESCRIPTION

`add_reward` allows a sponsor to deposit rewards before the market opens, when `staking_window_end` is still unset. `withdraw_reward` requires `staking_window_end` to exist before checking expiry, so a sponsor cannot withdraw if the market creator never opens the market.

The affected funds are the sponsor's voluntary pre-open reward deposit.

PRACTICAL IMPACT

Sponsors can lose access to reward funds if they deposit before the market opens and the market is never opened. The sponsor becomes dependent on the market creator.

LOCATION

[programs/opportunity_market/src/instructions/add_reward.rs:60](#)
[programs/opportunity_market/src/instructions/withdraw_reward.rs:62](#)

RECOMMENDATION

Either disallow `add_reward` before `open_market` , or add a pre-open abandon/refund path for sponsors.

L-04

Expired, unresolved markets cannot close option accounts

SEVERITY

LOW

CATEGORY

Lifecycle

DESCRIPTION

`close_option_account` contains logic for expired, unresolved markets, but the account constraint requires `market.reveal_ended` . Unresolved markets can never set `reveal_ended` , because `end_reveal_period` requires the market to have resolved.

The expired-unresolved branch is therefore unreachable, and option creators cannot reclaim option-account rent in that lifecycle path.

PRACTICAL IMPACT

Option creators cannot recover option-account rent when a market expires unresolved. User principal is unaffected.

LOCATION

[programs/opportunity_market/src/instructions/close_option_account.rs:22](#)
[programs/opportunity_market/src/instructions/close_option_account.rs:52](#)
[programs/opportunity_market/src/instructions/end_reveal_period.rs:28](#)

RECOMMENDATION

Move the `reveal_ended` requirement into handler logic so the explicit expired-unresolved branch can run.

L - 05

Durable authorities can be set to unusable/default addresses in one step

SEVERITY LOW	CATEGORY Access control
DESCRIPTION Several durable authority fields accept raw Pubkey or unchecked accounts without rejecting the default/system address or requiring a two-step accept flow. A typo or default pubkey assignment can permanently disable future updates, market operations, or fee claims. This is an authorized-action risk rather than an external attacker path.	PRACTICAL IMPACT A single admin typo can permanently break important operations such as updates, fee claiming, or market resolution.
LOCATION programs/opportunity_market/src/instructions/create_market.rs:96 programs/opportunity_market/src/instructions/create_market.rs:101 programs/opportunity_market/src/instructions/init_platform_config.rs:12 programs/opportunity_market/src/instructions/init_platform_config.rs:41 programs/opportunity_market/src/instructions/set_update_authority.rs:17 programs/opportunity_market/src/instructions/set_fee_claim_authority.rs:17 programs/opportunity_market/src/state.rs:79	RECOMMENDATION Reject the default pubkey for durable authority fields. Consider two-step nominate/accept flows for update and fee-claim authority changes.

L - 06

resolve_market and withdraw_reward deadlines overlap at exactly select_deadline

SEVERITY LOW	CATEGORY Lifecycle
DESCRIPTION resolve_market allows resolution while current_timestamp <= select_deadline , while withdraw_reward allows sponsor withdrawal when current_timestamp >= select_deadline . At exactly the deadline timestamp, a market is both still resolvable and sponsor-withdrawable. If a sponsor withdraws first in that slot and the market authority resolves afterward in the same timestamp, winners split a smaller reward pool than expected.	PRACTICAL IMPACT At the exact deadline, sponsors may withdraw rewards while the market can still resolve. Winners may receive a smaller reward pool than expected.
LOCATION programs/opportunity_market/src/instructions/resolve_market.rs:39 programs/opportunity_market/src/instructions/resolve_market.rs:43 programs/opportunity_market/src/instructions/withdraw_reward.rs:66 programs/opportunity_market/src/instructions/withdraw_reward.rs:70	RECOMMENDATION Make one side strict. For example, require current_timestamp > select_deadline in withdraw_reward .

L-07

Winning-option rent is stranded forever if any finalized winner never claims

SEVERITY LOW	CATEGORY Accounting
DESCRIPTION For winning options, <code>close_option_account</code> remains blocked until <code>option.unclaimed_stake</code> returns to zero. <code>unclaimed_stake</code> is decremented only when finalized winners claim rewards. If any finalized winner never claims, the option creator's rent remains stranded indefinitely, even though the abandoned winner is only forfeiting their own reward.	PRACTICAL IMPACT One inactive winner can permanently block the option creator from recovering rent. The financial impact is small, but cleanup can remain stuck forever.
LOCATION programs/opportunity_market/src/instructions/finalize_reveal_stake.rs:67 programs/opportunity_market/src/instructions/claim_rewards.rs:84 programs/opportunity_market/src/instructions/close_option_account.rs:31	RECOMMENDATION Allow option closure after a post-resolution claim window, or add a sweep/expiry mechanism that decouples option rent recovery from every individual winner claiming.

METHODOLOGY

REPORT ID: AIP-20260617-CA43B6 | DATE: 2026-06-17

AI ANALYSIS + HUMAN TRIAGE

This preaudit combines automated analysis with human validation. The AI security engine first generates a broad set of candidate findings. A security engineer then triages each one:

- False positives are removed. Candidates that are not real issues on closer inspection are discarded.
- Severity is recalibrated. Ratings are adjusted up or down to reflect actual exploitability and impact in this codebase, rather than the engine's initial guess.
- Non-security items are excluded. Pure code-quality or style suggestions are not security findings and are left out of the report.

DISCLAIMER

PREAUDIT SCOPE

This document is a preaudit, not a full security audit. A preaudit is a time-boxed readiness assessment: its goal is to gauge how prepared a codebase is for a comprehensive audit, not to exhaustively identify every vulnerability. The review combined an AI security engine with human validation over the scope and commit listed above, and was deliberately not exhaustive in coverage or depth.

READINESS SIGNAL

Findings in this report indicate areas to address before a full audit. The number and severity of findings should be read as a readiness signal: many findings suggest further hardening is needed before a full audit is worthwhile, while few or no findings suggest the codebase is reasonably prepared for one. A low finding count does not mean the code is free of vulnerabilities, nor that it is safe to deploy to production.

FULL AUDIT REQUIRED

This preaudit does not replace a full security audit. Adevar Labs strongly recommends a comprehensive audit of the final, frozen codebase before any mainnet deployment or handling of user funds. No warranty is given, express or implied, that the reviewed code is free from vulnerabilities or fit for any particular purpose.

TIME-SENSITIVE RISK

Risk ratings reflect the available evidence, stated assumptions, and severity calibration at the time of review, and may omit context outside the provided source. Security is time-sensitive: code, dependencies, deployment settings, and external integrations can change after this report is issued. Teams should re-review after material changes and maintain monitoring, incident response, and upgrade procedures appropriate for the protocol.