GMX · SECURITY REVIEW

GMX RewardRouterV3 & RewardRouterUtils.

A security review of RewardRouterV3 & RewardRouterUtils.

CLIENT	PROJECT	AUDITORS	AUDIT WINDOW	FINAL REPORT
GMX	RewardRouterV3 & RewardRouterUtils	Nicholas Chew, Andreas Zottl, Michael Lett, Ferran Celades	September 29 – October 20, 2026	November 26, 2026

Published at guardian.security/portfolio/gmx/rewardrouterv3-and-rewardrouterutils-2026-11-26

§ 00

Contents.

40 pages: overview, confidence ranking, scope, 20 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 · 14 findings P. 09

Remediation Review · 6 findings P. 31

§ 05 **Disclaimer** P. 39

§ 06 **About Guardian** P. 40

§ 01

Overview.

Guardian reviewed RewardRouterV3 & RewardRouterUtils across 2 review rounds. The final report includes 1 High, 8 Low, 11 Informational findings.

AUDIT FIRM
Guardian

CLIENT
GMX

PROJECT
RewardRouterV3 & RewardRouterUtils

PREPARED BY
Nicholas Chew, Andreas Zottl, Michael Lett, Ferran Celades

AUDIT WINDOW
September 29 – October 20, 2026

FINAL REPORT
November 26, 2026

REPOSITORY
`https://github.com/gmx-io/gmx-synthetics-private`

METHODOLOGY
Smart contract · Manual · Main Review · Remediation Review

MAIN REVIEW COMMIT
`f5f939a6861f`

REMEDIATION REVIEW COMMIT
`7fbd664f3cbf`

1.1 Confidence Ranking

4 / 5

High Confidence

Guardian assigns a Confidence Ranking of 4 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	**1**	0	0	0	1
Medium	**0**	0	0	0	0
Low	**8**	0	5	0	3
Informational	**11**	0	10	0	1
All severities	**20**	**0**	**15**	**0**	**5**

§ 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	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 ASSIGNED	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 RewardRouterV3 & RewardRouterUtils in `https://github.com/gmx-io/gmx-synthetics-private`. The scope includes 1,231 source lines across 10 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		
contracts/reader/RatioVesterReader.sol	51	62
contracts/staking/RewardRouterUtils.sol	109	139
contracts/staking/RewardRouterV3.sol	486	636
REMEDIATION REVIEW		
contracts/config/Config.sol	1	463
contracts/config/ConfigKeys.sol	100	261
contracts/config/TimelockConfig.sol	54	600
contracts/data/Keys.sol	19	2,522
contracts/error/Errors.sol	11	613
contracts/incentive/IncentiveEventUtils.sol	65	315
contracts/staking/RewardRouterUtils.sol	192	344
contracts/staking/RewardRouterV3.sol	105	728
contracts/staking/UnstakeRequestStoreUtils.sol	38	54

§ 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					
H-01	Default None voting mode leaves unbacked GovToken votes after V3 unstake	stale governance state / governance takeover	HIGH	RESOLVED	p. 9
L-01	A zero-rounded bnGMX reduction can block both exits from a mixed stake	Rounding	LOW	RESOLVED	p. 11
L-02	Vester changes can split account transfers	Business logic errors	LOW	RESOLVED	p. 13
L-03	Strict-transfer authorization is checked against the wrong asset and never re-bound, so a zero allowance sweeps the GMX/esGMX/bnGmx/GLP position	authorization / TOCTOU	LOW	ACKNOWLEDGED	p. 14
L-04	Legacy vester collateral can block V3 boost-cap reconciliation	reward-cap-bypass	LOW	RESOLVED	p. 16
I-01	Configuration and transfer lifecycle changes lack canonical router events	Events	INFORMATIONAL	ACKNOWLEDGED	p. 18
I-02	A lower boost cap requires collateral-aware account synchronization	Configuration	INFORMATIONAL	ACKNOWLEDGED	p. 20
I-03	Voting-basis changes require an explicit holder synchronization	Configuration	INFORMATIONAL	ACKNOWLEDGED	p. 22
I-04	Aggregate GMX reward handling can succeed after silently skipping designated vesters	Documentation	INFORMATIONAL	ACKNOWLEDGED	p. 24
I-05	Designated vesters lack address validation	Validation	INFORMATIONAL	RESOLVED	p. 25
I-06	V3 transfers leave legacy vester caps with sender	Informational	INFORMATIONAL	ACKNOWLEDGED	p. 26
I-07	V3 staking events require indexer migration from legacy router topics	Events	INFORMATIONAL	ACKNOWLEDGED	p. 27
I-08	Legacy bnGMX-only fee deposits can retain reward weight after V3 unstake	Informational	INFORMATIONAL	ACKNOWLEDGED	p. 28
I-09	Vester collateral can block legacy restaking	Informational	INFORMATIONAL	ACKNOWLEDGED	p. 30

Findings continued.

ID	TITLE	CATEGORY	SEVERITY	STATUS	PAGE
REMEDIATION REVIEW					
L-01	Separate setters revive expired unstake requests	Business logic errors	LOW	ACKNOWLEDGED	p. 31
L-02	Legacy pair locks block new vesting deposits	Business logic errors	LOW	ACKNOWLEDGED	p. 32
L-03	Router replacement can revive invalidated staking-account transfers	Business logic errors	LOW	ACKNOWLEDGED	p. 33
L-04	Residual balances pin a deactivated vester	Business logic errors	LOW	ACKNOWLEDGED	p. 35
I-01	Issuer vester binding is not checked	Validation	INFORMATIONAL	ACKNOWLEDGED	p. 37
I-02	Vester views and events disagree	Unexpected Behavior	INFORMATIONAL	ACKNOWLEDGED	p. 38

HIGH **H-01** RESOLVED

Default None voting mode leaves unbacked GovToken votes after V3 unstake

CATEGORY	stale governance state / governance takeover
LOCATION	packages/gmx-synthetics-private/contracts/staking/RewardRouterV3.sol:setVotingPowerType:L149-L151
REVIEW	Main Review

DESCRIPTION

RewardRouterV3 leaves votingPowerType at the Solidity default None, while migration separately grants it tracker and GOV_TOKEN_CONTROLLER permissions before a later config-mirror action. In None, every _syncVotingPower call returns without comparing or burning GovToken. Thus a V2-staked account can have V3 unwind its complete tracker position while retaining its V2-minted GovToken balance and delegated votes. The fork PoC passes with a real deployment against the live Arbitrum V2 trackers, demonstrating full unstake, stale votes, and repeated cycling of one GMX amount through fresh accounts to inflate supply; it also successfully uses the stale votes in the live Governor to grant ROLE_ADMIN.

Impact: An ordinary holder can recover all GMX while retaining the voting weight that was backed by it. During the migration window, the same stake can be cycled through fresh accounts, producing arbitrarily many unbacked GovTokens with no additional capital. The demonstrated stale votes satisfy the live Governor threshold/quorum and execute a RoleStore.grantRole call, yielding protocol-wide role control and the ability to seize governed assets/configuration.

RECOMMENDATION

Do not expose tracker/GovToken handler permissions while the router is in None. Initialize and atomically mirror the intended voting mode before enabling V3 operations, or make setVotingPowerType reconcile all affected accounts and ensure unstake/transfer burns stale balances even in None. If None is retained as a disable mode, explicitly burn/reconcile existing router-issued votes and block stake/unstake until reconciliation; emit mode-change events and provide a non-no-op resync path.

PROOF OF CONCEPT https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/default-none-voting-mode-leaves-unbacked-govtoken-votes-afte-4aa57a9d.sol

H-01 Default None voting mode leaves unbacked GovToken votes after V3 unstake

CONTINUED

RESOLUTION

Status: Resolved.

LOW **L-01** RESOLVED

A zero-rounded bnGMX reduction can block both exits from a mixed stake

CATEGORY	Rounding
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L538-L543
REVIEW	Main Review

DESCRIPTION

When a user withdraws GMX or esGMX, [`_unstakeGmx`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L524-L550) removes a proportional share of the account's staked bnGMX using integer division. If the account has 100 GMX, 100 esGMX, and exactly one bnGMX base unit staked, either full-component withdrawal calculates `floor(1 * 100 / 200) = 0`. The outer branch checks only that the staked bnGMX balance is nonzero, then calls both production reward trackers with the zero result. Their positive-amount guards revert the transaction, including the earlier principal unstake. Because each public route removes only one component, neither full-component exit can derive a positive reduction in this exact one-unit state.

The trigger requires a supported configuration with a nonzero boost cap, one compounded bnGMX base unit, positive GMX and esGMX deposits, and no additional bonus unit compounded before withdrawal. Formula-only analysis of the canonical two-floor reward calculation and an isolated custody-bearing tracker model demonstrate that the one-unit arithmetic state is conditionally reachable; they do not reproduce live clock-based emissions or prove that current emission rates create it. Active emissions may make recovery immediate by accruing another unit that the user can compound. Governance can also reduce the cap to zero and the user can compound to return the one staked unit to the wallet.

There is no attacker profit or accounting theft, but both components of the user's principal can be temporarily unavailable. At the reviewed Arbitrum block `509962082`, the configured V2 router had a zero boost cap and aggregate extended-tracker bnGMX deposits were zero. The V3 launch plan mirrors that cap, so the condition is dormant in the reviewed launch configuration. The same proportional arithmetic exists in V2, making this inherited behavior rather than a V3 regression.

L-01 A zero-rounded bnGMX reduction can block both exits from a mixed stake CONTINUED

RECOMMENDATION

Be aware of and document this behavior or consider skipping the tracker unstake-and-burn sequence when `reductionAmount == 0`. The remaining one-unit bonus balance can then be consumed when a later withdrawal produces a positive proportional result. Add a regression case with positive GMX and esGMX deposits and exactly one staked bnGMX base unit.

RESOLUTION

Status: Resolved.

LOW **L-02** RESOLVED

Vester changes can split account transfers

CATEGORY	Business logic errors
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L382-L444
REVIEW	Main Review

DESCRIPTION

`signalTransfer` validates the current designated vester list but records only the receiver. A timelock action can add or remove a vester before `acceptTransfer`, which validates and migrates only the list present at acceptance.

If an omitted vester holds the sender's cap, converted usage, or unpaid claims, the stake and remaining vester states move while that state remains on the old address. The receiver is then no longer fresh in the vesters that already moved, so the residual rights cannot be reunited with the same receiver. This can happen during partial initial registration or when a vester is removed after a transfer is signalled. The migration script warns that initial registrations must be atomic, but the contract does not enforce that rule or protect pending transfers from later removals.

RECOMMENDATION

Be aware of and document this behavior. A signal records the receiver only, so adding or removing a designated vester before acceptance can leave that vester's cap, converted usage, or unpaid claims on the sender.

RESOLUTION

Status: Resolved.

LOW **L-03**

ACKNOWLEDGED

Strict-transfer authorization is checked against the wrong asset and never re-bound, so a zero allowance sweeps the GMX/esGMX/bnGmx/GLP position

CATEGORY	authorization / TOCTOU
LOCATION	`packages/gmx-synthetics-private/contracts/staking/RewardRouterV3.sol:signalTransfer:L382-L397`
REVIEW	Main Review

DESCRIPTION

In strict-transfer mode, signalTransfer checks only the sender's current feeGmxTracker staked amount against the receiver allowance, then records only the receiver. It does not cover the independently migrated GLP leg and does not snapshot an amount or deadline. acceptTransfer does not revalidate the allowance before migrating the sender's then-current GMX, esGMX, bnGMX, and GLP position. Consequently, a GLP-only or empty sender can signal with zero allowance, and a delayed receiver can later accept after the position grows even though the allowance remains zero. This also contradicts the MultichainStakingUtils assumption that the approved exact amount cannot become stale.

Impact: After the sender has explicitly signaled a receiver, that receiver—or anyone later controlling its key—can accept an unbounded, non-expiring authorization and obtain GLP plus assets added after signaling that were never covered by the strict-mode allowance. This defeats strict mode's secondary approval safeguard, though exploitation still requires a prior sender signal, so Medium rather than High is appropriate.

RECOMMENDATION

At acceptance, revalidate strict-transfer authorization against the position being migrated. Bind the pending transfer to amounts for every migrated leg or invalidate it when the position changes, and add an expiry and explicit cancellation mechanism. Do not treat a zero feeGmxTracker balance as approval for an unrelated GLP position.

PROOF OF CONCEPT `https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/strict-transfer-authorization-is-checked-against-the-wrong-a-98c21688.sol`

L-03 Strict-transfer authorization is checked against the wrong asset and never re-bound, so a zero allowance sweeps the GMX/esGMX/bnGmx/GLP position CONTINUED

RESOLUTION

Status: Acknowledged.

LOW **L-04** RESOLVED

Legacy vester collateral can block V3 boost-cap reconciliation

CATEGORY	reward-cap-bypass
LOCATION	`contracts/staking/RewardRouterV3.sol:593-603; contracts/staking/RewardRouterUtils.sol:106-120`
REVIEW	Main Review

DESCRIPTION

RewardRouterV3._restakeForAccount temporarily pulls feeGmxTracker receipt tokens held by the legacy gmxVester as pair collateral, converts an old-layout bonus position, then returns the same collateral. RewardRouterUtils.stakeBnGmx can meanwhile remove over-cap extended-tracker bnGMX by unstaking its outer fee-tracker deposit. That first burns the account's fungible fee-tracker receipts. If too few receipts remain to restore the vester collateral, the return reverts and the entire reconciliation rolls back. For example, 100 base stake, 50 old bonus-in-fee, 1,000 extended bnGMX, a 100 bnGMX cap, and 400 receipt tokens of vester collateral cause a 900-receipt trim to leave only 250 for the 400-receipt return.

This requires an existing old-layout, over-cap account with legacy-vester collateral and a cap/configuration transition. The linked fork PoC uses live trackers and vester code, but grants the harness privileged handler/minter roles, mints the input tokens, sets legacy-vester stake history, and selects router modes to construct that state. It does not establish a comparable ordinary-user position at the planned zero-cap cutover. For an affected account, compound and restaking-dependent exits or batches can revert, while claimFees still pays against its unreconciled fee-tracker weight. BaseAndBonus voting effects require that governance-selected mode. Withdrawing from gmxVester releases collateral and permits reconciliation but interrupts the vesting position; no permanent principal loss is shown. This is a conditional, recoverable Low-severity migration/cap-enforcement issue related to I-01 and I-14.

RECOMMENDATION

Document the collateral constraint and identify affected accounts before V3 cutover or a boost-cap decrease. Reconcile old-layout and over-cap positions first; for an affected account, release legacy-vester collateral before retrying and record the vesting tradeoff. If reconciliation must preserve an open vester position, update router and vester collateral accounting together so the cap trim cannot consume receipts owed back to the vester. Merely moving the collateral return before the trim does not provide enough free receipts.

L-04 Legacy vester collateral can block V3 boost-cap reconciliation CONTINUED

PROOF OF CONCEPT https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/v3-boost-cap-trim-reverts-on-legacy-vester-collateral-shield-67dcfe98.sol

RESOLUTION

Status: Resolved.

INFORMATIONAL **I-01** ACKNOWLEDGED

Configuration and transfer lifecycle changes lack canonical router events

CATEGORY	Events
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L137-L177
REVIEW	Main Review

DESCRIPTION

The router's [timelock configuration and list functions](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L137-L177) update storage without a router or `EventEmitter` record. Likewise, [`signalTransfer` and `acceptTransfer`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L382-L448) do not emit canonical router events when a pending receiver is created, overwritten, or consumed. An intended receiver therefore cannot discover an incoming signal from router logs without already knowing the sender, and an event-only indexer cannot distinguish an initial signal from an overwrite.

Useful evidence remains available: current values are readable from storage, governance calldata and traces reveal configuration calls, tokens and trackers emit downstream logs, and each successful acceptance produces designated-vester transfer records containing both parties. An indexer can therefore infer some state transitions, but it must correlate several dependency events and transaction data instead of consuming a stable router event schema. This is an observability gap rather than an authorization or asset-safety failure; it can cause delayed alerts, incomplete historical dashboards, and transfer-status ambiguity.

RECOMMENDATION

Be aware of and document this behavior or consider emitting standardized `EventEmitter` records for each configuration or list update and for transfer signal, overwrite, and acceptance. Include old and new configuration values and both transfer parties so indexers do not need to reconstruct a single logical operation from storage polling, calldata, traces, and downstream logs.

I-01 Configuration and transfer lifecycle changes lack canonical router events CONTINUED

RESOLUTION

Status: Acknowledged.

INFORMATIONAL **I-02** ACKNOWLEDGED

A lower boost cap requires collateral-aware account synchronization

CATEGORY	Configuration
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterUtils.sol#L106-L120
REVIEW	Main Review

DESCRIPTION

[`setMaxBoostBasisPoints`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L145-L147) changes the boost cap without synchronizing existing accounts. A lower cap reaches an account only when a later compound-style action calls [`RewardRouterUtils.stakeBnGmx`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterUtils.sol#L106-L120); direct fee claims do not rebalance it.

If a designated `RatioVester` holds the account's fee-tracker receipt tokens as pair collateral, a cap-down rebalance can fail. The rebalance first asks the fee tracker to unstake excess extended-tracker receipts, which requires burning fee-tracker receipts from the account, but those receipts are held by the vester. When the account's free balance is insufficient, both user and administrative compound attempts revert while fee claims remain available. The account can therefore retain its old above-cap reward weight and dilute other fee-reward recipients until the collateral is released.

This is a trusted policy-transition constraint inherited from V2: users cannot cause governance to lower the cap, and the setter does not promise atomic synchronization of all accounts. At the reviewed Arbitrum block `509962082`, the configured V2 router had a zero boost cap and aggregate extended-tracker bnGMX deposits were zero. The V3 launch plan mirrors that cap, so the trigger is dormant in the reviewed launch configuration. Principal is not permanently locked. The user can withdraw from the vester to recover the pair receipts and then compound to apply the lower cap, although doing so interrupts the existing vesting schedule.

RECOMMENDATION

Be aware of and document this behavior or consider requiring a cap-decrease runbook that identifies over-cap accounts, handles designated-vester collateral, and verifies the

I-02 A lower boost cap requires collateral-aware account synchronization CONTINUED

resulting extended-tracker positions before treating the new cap as effective. If seamless enforcement is required, make cap-down cleanup collateral-aware, similar to the temporary legacy-vester receipt return in `_restakeForAccount`.

RESOLUTION

Status: Acknowledged.

INFORMATIONAL I-03

ACKNOWLEDGED

Voting-basis changes require an explicit holder synchronization

CATEGORY	Configuration
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L149-L151
REVIEW	Main Review

DESCRIPTION

[`setVotingPowerType`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L149-L151) updates the basis used by future account synchronizations but does not recalculate existing `GovToken` balances or delegated votes. For example, a holder with 13 votes under the base-plus-bonus basis can retain 13 after governance selects a base-only target of 10 by using only direct claim routes, which do not call [`_syncVotingPower`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L559-L590). The balance is corrected only when a later stake, unstake, compound, reward-handling, transfer, or complete administrative batch synchronizes that holder.

Selecting `None` stops router maintenance of existing voting balances because `_syncVotingPower` returns immediately. It does not clear those balances. If governance activity proceeds before a transition is complete, holders can have balances and checkpoints calculated under different bases, potentially affecting proposal thresholds, quorum, or vote outcomes.

This is a trusted policy transition, and the setter does not promise an atomic update of all holders. The reviewed launch retains the existing `BaseStakedAmount` basis, so no transition is currently planned. An active-to-active change can be completed through holder synchronization and a verified administrative batch. If `None` is intended to retire voting power rather than stop router maintenance, balances must be explicitly burned or migrated before selecting it.

RECOMMENDATION

Be aware of and document this behavior or consider publishing a holder-synchronization procedure for changes between the two active voting bases and verifying `GovToken` balances against the new basis before relying on it for governance. Define the intended `None`

I-03 Voting-basis changes require an explicit holder synchronization CONTINUED

semantics explicitly: the current implementation stops router maintenance without clearing outstanding balances, so any required burn or migration should occur before selecting `None`.

RESOLUTION

Status: Acknowledged.

INFORMATIONAL I-04

ACKNOWLEDGED

Aggregate GMX reward handling can succeed after silently skipping designated vesters

CATEGORY	Documentation
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L462-L478
REVIEW	Main Review

DESCRIPTION

The `_shouldClaimGmx` option combines extended-tracker GMX and designated-vester GMX into one best-effort action, but [`_claimVestedGmx`](https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L462-L478) silently skips designated vesters when the V2-compatible handler sends tracker rewards to a receiver other than the account, when a vester's claim feature is disabled, or when the account is frozen on that vester. The extended-tracker claim and any eligible vester claims can still succeed, so transaction success does not establish that every requested GMX source paid. The handlers return no per-source result, and no event identifies the skipped vester or reason.

The skip-on-pause and skip-on-freeze behavior is intentional so one vester does not disable the entire reward route. No value is lost: skipped GMX stays in the vester. For a disabled or frozen vester, both aggregate handling and direct `vesterClaim` remain subject to the same checks until the condition clears. For the external-receiver skip, the user can call `vesterClaim` directly or later use aggregate handling with the account as receiver. The practical effect is delayed collection, an extra transaction, and a risk that automation records a partially processed request as complete.

RECOMMENDATION

Be aware of and document this behavior or consider emitting a per-vester skip outcome containing the vester and reason. Document that an external GMX receiver applies only to tracker-paid GMX and intentionally suppresses designated-vester claims.

RESOLUTION

Status: Acknowledged.

INFORMATIONAL I-05 RESOLVED

Designated vesters lack address validation

CATEGORY	Validation
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L161-L165
REVIEW	Main Review

DESCRIPTION

`addDesignatedVester` rejects only duplicate addresses. It accepts the zero address, an EOA, or a contract that does not implement the expected vester interface.

Reward handling and account transfers later call every designated entry. An invalid entry therefore makes those aggregate paths revert until the timelock removes it. This requires a trusted configuration mistake and is recoverable, so it is an operational configuration risk rather than an attacker-controlled denial of service.

RECOMMENDATION

Reject the zero address and addresses without deployed code. Verify the expected vester interface, token wiring, provisioning state, and controller permissions as part of the configuration process.

RESOLUTION

Status: Resolved.

INFORMATIONAL I-06 ACKNOWLEDGED

V3 transfers leave legacy vester caps with sender

CATEGORY	Informational
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L399-L448
REVIEW	Main Review

DESCRIPTION

RewardRouterV3.acceptTransfer moves tracker stake and designated RatioVester state but does not move the sender's legacy Vester cap as V2 did. A sender with an unused legacy cap can therefore retain it after transferring the staked position, while the fresh receiver gets no legacy cap. The migration separately plans to zero legacy caps after enabling V3 transfers. Until an account's cap is zeroed, it can still use that cap if it independently has esGMX and pair collateral. The transfer itself sweeps the sender's existing esGMX and pending issuance, and the fork PoC supplies post-transfer esGMX through test-controlled privileged issuance. This is a transfer behavior and migration timing detail for the team to document; the PoC does not establish duplicate redemption or third-party loss.

RECOMMENDATION

Recommendation (Informational): Be aware of and document that a V3 stake transfer leaves legacy vester caps with the sender, while caps in the new designated vesters move to the receiver. A retained legacy cap remains usable until the planned zeroing step if the sender has esGMX and collateral.

PROOF OF CONCEPT https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/rewardrouterv3-transfer-omits-legacy-vester-cap-closure-sell-dcc530f4.sol

RESOLUTION

Status: Acknowledged.

INFORMATIONAL **I-07** ACKNOWLEDGED

V3 staking events require indexer migration from legacy router topics

CATEGORY	Events
LOCATION	`packages/gmx-synthetics-private/contracts/staking/RewardRouterV3.sol:_stakeGmx/_unstakeGmx:L504-L556`
REVIEW	Main Review

DESCRIPTION

RewardRouterV2 emits `StakeGmx` and `UnstakeGmx` directly from the router. RewardRouterV3 instead calls `IncentiveEventUtils.emitStakeGmx` and `emitUnstakeGmx`, which emit `EventLog1` records through `EventEmitter` under the names `RewardRouterStakeGmx` and `RewardRouterUnstakeGmx`. An indexer that switches to the V3 router address but continues filtering for the legacy event topics will not observe those V3 staking operations.

This is an event-schema migration consideration. The linked PoC confirms the different event topics, but no currently deployed consumer that depends on the legacy topics at the V3 address was identified. On-chain staking balances and rewards are unaffected.

RECOMMENDATION

Document the V3 staking event schema in the migration guidance. Update affected indexers to read the `EventEmitter` records and their account, token, and amount fields. Add legacy event compatibility only if a specific consumer requires it.

PROOF OF CONCEPT `https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/rewardrouterv3-omits-legacy-stakegmx-unstakegmx-events-3dbb4d3f.sol`

RESOLUTION

Status: Acknowledged.

INFORMATIONAL I-08

ACKNOWLEDGED

Legacy bnGMX-only fee deposits can retain reward weight after V3 unstake

CATEGORY	Informational
LOCATION	packages/gmx-synthetics-private/contracts/staking/RewardRouterV3.sol:_restakeForAccount:L593-L625
REVIEW	Main Review

DESCRIPTION

Before cutover, an account can have bnGMX deposited directly in `feeGmxTracker`. When `_restakeBnGmx` runs, it withdraws that entire direct deposit and calls `_stakeBnGmx` to place wallet bnGMX into `extendedGmxTracker` up to the boost cap. But `_restakeForAccount` reaches this conversion only when restaking mode is enabled and the account also has a nonzero old `bonusGmxTracker` deposit in `feeGmxTracker`. A legacy bnGMX-only fee deposit therefore skips the conversion, even if the account is included in `batchRestakeForAccounts`.

The subsequent `_stakeBnGmx` call checks bnGMX already in `extendedGmxTracker` and the account's wallet balance; neither includes bnGMX still held directly by `feeGmxTracker`. V3's proportional burn on base unstake likewise checks only the extended-tracker bnGMX deposit. An unswept direct deposit can therefore remain after full base unstake and continue contributing to fee-tracker reward weight. GovToken voting power also reflects it only when `BaseAndBonusStakedAmount` is selected.

This is a migration edge case, not evidence of a material current reward loss. The linked PoC demonstrates the accounting with a constructed 3,000-unit legacy position; it does not establish a comparable live position. The migration readiness script permits a small direct fee-layer bnGMX custody residual by default.

RECOMMENDATION

Be aware that a legacy account with direct fee-layer bnGMX but no old bonus-tracker deposit can skip the restake conversion and retain fee reward weight after base unstake. Document this edge case in the cutover runbook and check direct fee-layer bnGMX residuals so operators can decide whether any account needs manual handling.

PROOF OF CONCEPT https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/direct-legacy-fee-layer-bngmx-bypasses-boost-cap-and-survive-0cac866f.sol

I-08 Legacy bnGMX-only fee deposits can retain reward weight after V3 unstake

CONTINUED

RESOLUTION

Status: Acknowledged.

INFORMATIONAL I-09

ACKNOWLEDGED

Vester collateral can block legacy restaking

CATEGORY	Informational
LOCATION	https://github.com/GuardianOrg/mirror-gmx-io-gmx-synthetics-private-7432a9686e2c2135064422ef5bf25c36/blob/f5f939a6861f46001fdf4bc4bc783b4df4afdf68/contracts/staking/RewardRouterV3.sol#L593-L610
REVIEW	Main Review

DESCRIPTION

When inRestakingMode is enabled for an account with an unswept legacy bonusInFee deposit, _restakeForAccount temporarily recalls feeGmxTracker receipts from the legacy gmxVester only. A designated RatioVester can also hold that account's receipts as pair collateral while the tracker still records the full deposit. _restakeBonusGmxTracker may then try to burn more receipts than the account holds, so compound, unstake, and a batch containing the account can revert. Withdrawing the RatioVester collateral restores the receipts but ends the vesting session. This is a conditional migration concern because the launch plan requires legacy sweep populations cleared and V3 restaking mode off.

RECOMMENDATION

Be aware of and document this behavior: unswept legacy bonus deposits combined with designated RatioVester collateral can block restaking-related actions if V3 restaking mode is enabled. Confirm the migration readiness gate before enabling that mode, and document the collateral-withdrawal recovery tradeoff.

PROOF OF CONCEPT https://github.com/GuardianOrg/SentryRepo-RewardRouterV3---RewardRouterUtils--Round-1-01a0f2c981347bf9a1cba1123bd62ff4/blob/refinery/poc-01a0f2ca-b353-7ba3-96b1-90ae75f41c39/test/refinery-poc/vester-pair-collateral-blocks-legacy-bonusinfee-sweeps-until-2121c08a.sol

RESOLUTION

Status: Acknowledged.

LOW **L-01** ACKNOWLEDGED

Separate setters revive expired unstake requests

CATEGORY	Business logic errors
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/config/TimelockConfig.sol#L534-L584 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterUtils.sol#L168-L180
REVIEW	Remediation Review

DESCRIPTION

`signalSetUnstakeDelay` checks the proposed delay against the current expiry and schedules only the delay setter. To move a configured 90-day delay and 100-day expiry to 180 and 200 days through these wrappers, governance must execute the expiry increase first. Requests retain their original timestamp and are checked against the current settings, so an expired request can become executable before the longer delay takes effect.

For example, Alice signals an unstake of 100 GMX on day 0. On day 151:

Settings	Alice's request
90-day delay, 100-day expiry	Expired since day 100
90-day delay, 200-day expiry, after the first setter	Executable; Alice can withdraw 100 GMX
180-day delay, 200-day expiry, intended final settings	Cannot execute until day 180

This requires a previously shortened expiry; V3's initial expiry is 100 years.

RECOMMENDATION

Be aware of and document this behavior: increasing expiry before delay can reopen expired requests between governance transactions. If the new delay must apply before withdrawals resume, execute both setters in one atomic timelock batch, with expiry first.

RESOLUTION

Status: Acknowledged.

GMX Team: Acknowledged. The expiry stays at its 100-year default and no change to it is planned, so the precondition does not arise. The behaviour is documented in our ops runbook, including that the wrappers schedule each setter separately and validate it against the live values, so the suggested atomic batch is not available through them.

LOW **L-02** ACKNOWLEDGED

Legacy pair locks block new vesting deposits

CATEGORY	Business logic errors
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterUtils.sol#L255-L275 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterV3.sol#L377-L380
REVIEW	Remediation Review

DESCRIPTION

RewardRouterV3.vesterDeposit runs syncFeeGmxTrackerBalance before every deposit into either new Vester. The sync reverts when the account's wallet sbfGMX plus its old and new Vester pairAmounts exceeds feeGmxTracker.stakedAmounts by more than the wallet holds. Equivalently, the locked pair ledger alone exceeds the fee tracker stake ledger. Historical VesterCap operations left old pairAmounts in place after releasing sbfGMX receipts, so an account can meet this condition without making another legacy deposit.

For example, an Arbitrum account has 93.7328 sbfGMX in its wallet, 156.7095 in the old Vester pair ledger, and 93.7328 in the fee tracker stake ledger. The 156.7095 excess exceeds its 93.7328 wallet balance, so any V3 vesting deposit reverts. The user must withdraw the old Vester position to clear its pair ledger, either early or after it finishes vesting. An early withdrawal returns unvested esGMX but ends its existing vesting session; redepositing that amount starts a new vesting period.

RECOMMENDATION

Be aware of and document this behavior: affected users need a clear old-Vester withdrawal path before they can deposit into the new vesters. Confirm that the old Vester can return its recorded pair receipts during the cutover, and explain the vesting-time effect of withdrawing early.

RESOLUTION

Status: Acknowledged.

GMX Team: Acknowledged, intended. The revert is the guard added for the unbacked sbfGMX, so pair receipts the legacy vester still records cannot back a new position. An affected account withdraws its legacy position first, or stakes enough GMX to cover it, which our integrator guide now states together with the cost of an early legacy withdrawal. On the cutover question, the legacy vester holds less than its recorded pair amounts by exactly the unbacked excess, which returns to it as affected accounts deposit through V3 or are synced, and the sync stays available until the gap is closed.

LOW L-03

ACKNOWLEDGED

Router replacement can revive invalidated staking-account transfers

CATEGORY	Business logic errors
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterV3.sol#L104 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterV3.sol#L209-L235 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterV3.sol#L514-L534 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/data/Keys.sol#L577-L584
REVIEW	Remediation Review

DESCRIPTION

Pending transfer receivers and captured vester versions persist in sender-keyed DataStore entries, but vesterConfigVersion is local to each router deployment. Vester-list changes invalidate pending signals only by incrementing the local counter; they do not remove those entries. A replacement can reproduce an earlier version integer and accept a signal already invalidated on its predecessor. The revised regression uses real vester addition, deactivation and drained removal to restore the predecessor's original vester set, checks matching final transfer policies, and demonstrates that the predecessor rejects the stale signal while the replacement accepts it.

Impact: Following vester lifecycle changes and a router replacement with a colliding counter, a previously designated receiver can transfer the sender's staking balances, wallet esGMX/bnGMX, pending unstake requests and vesting entitlements without a fresh signal. The regression asserts loss of the sender's assets and subsequent GMX withdrawal by the receiver. This requires a stale, unconsumed signal and the described migration; it does not permit arbitrary-account theft or additional voting-token inflation.

RECOMMENDATION

Be aware that a replacement RewardRouterV3 starts its own vesterConfigVersion at zero, while a pending transfer remains in the shared DataStore. Document this during a router migration. Before granting the replacement its transfer permissions, compare its version with any stored pendingTransferVesterVersion values, and have affected senders signal again or clear those entries. Adding and removing designated vesters on the predecessor does not by itself clear a signal.

RESOLUTION

Status: Acknowledged.

L-03 Router replacement can revive invalidated staking-account transfers CONTINUED

GMX Team: Acknowledged, deferred. The router deployed with this release is the first to store these entries, so nothing can be revived at launch. A replacement router will not accept transfer signals stored by its predecessor; this is tracked for the next router change and documented in our ops runbook meanwhile.

LOW **L-04** ACKNOWLEDGED

Residual balances pin a deactivated vester

CATEGORY	Business logic errors
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/incentive/RatioVester.sol#L276-L283 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/staking/RewardRouterUtils.sol#L233-L240
REVIEW	Remediation Review

DESCRIPTION

`removeDesignatedVester` is the only removal path from `designatedVesters`, and it accepts a vester only when `isDrainedForRemoval` is true. After deactivation, that still requires `totalSupply == 0`, `pairSupply == 0`, and `totalConverted == totalGmxClaimed`. A bound issuer also needs `totalIssuedAmount == totalClaimedAmount`.

No vester function clears another account's escrow, pair collateral, or unpaid GMX. Router withdraw and claim run only for `msg.sender`. `claimForAccount` reverts while the account is frozen, and `handleRewards` skips that claim. `withdrawToken` can transfer only GMX above `totalSupply + totalConverted - totalGmxClaimed`, so unpaid vested GMX stays reserved. A guardian can lift the freeze, after which the account itself can claim. Unclaimed issuer allocations can be reduced to zero with `reduceIssuedAmount`. Escrow, pair collateral, and unpaid vested GMX cannot.

An account that never withdraws and claims, or a frozen account with settled unpaid GMX, leaves the deactivated vester on the list. One wei of settled unpaid GMX is enough: `totalConverted` stays above `totalGmxClaimed`, and removal reverts with `DesignatedVesterNotDrained`. Later vesters are appended around it. Account transfers, GMX claims, and vesting deposits keep calling into every listed vester. `validateSender` calls `validateTransfersOpen` on each one, and that call reverts when the vester's transfer feature is disabled. Disabling transfers on a vester that cannot be removed rejects every `signalTransfer` and `acceptTransfer`.

RECOMMENDATION

Be aware of and document this behavior: residual escrow or unpaid vested GMX keeps a deactivated vester on designatedVesters, and disabling that vester's transfer feature blocks every account transfer until the residual is cleared by the account.

RESOLUTION

Status: Acknowledged.

L-04 Residual balances pin a deactivated vester CONTINUED

GMX Team: Acknowledged, intended. Removal requires a drained vester so that no account's escrow, pair or unpaid GMX becomes unreachable, since the router only serves designated vesters. A transfer pause on any designated vester pausing all account transfers is also intended, so vesting state always moves together. Both are documented in our ops runbook.

INFORMATIONAL **I-01** ACKNOWLEDGED

Issuer vester binding is not checked

CATEGORY	Validation
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/incentive/EsGmxIssuer.sol#L64-L68
REVIEW	Remediation Review

DESCRIPTION

`setVester` can be called once. It rejects the zero address and a second assignment, then stores `_vester` without reading that contract. `confirmIssuerBinding` on a vester succeeds only when this issuer's `vester` equals that vester. A mismatched assignment makes that check fail permanently, leaves `isIssuerBindingConfirmed` false, and makes every deposit revert with `IncentiveIssuerBindingNotConfirmed`. The issuer has to be redeployed, which drops its `issuedAmounts` ledger.

The season vester is constructed with this issuer. The orphan vester is constructed with the zero address, so this binding does not apply to it. Users can still claim issued esGMX from the issuer. They cannot vest those tokens on the real season vester until the binding is confirmed.

RECOMMENDATION

Require `IRatioVester(_vester).issuer()` to equal this issuer before storing the binding.

RESOLUTION

Status: Acknowledged.

GMX Team: Acknowledged, deferred. For this release the binding is set and confirmed right after deployment, before anything is funded or issued, and our deployment notes require checking the vester's issuer before setVester. The check in setVester will be in place before the next season's issuer is deployed.

INFORMATIONAL **I-02** ACKNOWLEDGED

Vester views and events disagree

CATEGORY	Unexpected Behavior
LOCATION	https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/incentive/RatioVester.sol#L221-L238 https://github.com/gmx-io/gmx-synthetics-private/blob/7fbd664f3cbf89c748c0e86f42319aacfbf05259/contracts/incentive/RatioVester.sol#L401-L454
REVIEW	Remediation Review

DESCRIPTION

`transferVestingState` emits `getVestingCap` as `cap`. An uncapped vester returns `type(uint256).max` there, and the same function skips the cap move, so the event reports a max cap with zero `usage`. The vester deployment in this repo sets `isUncapped` to false.

`claimable` still counts settled but unpaid GMX for a frozen account (`cumulativeClaimAmounts - claimedAmounts + unpaidClaimAmounts`). Pending conversion is zero while the account is frozen, and `_claim` reverts with `VesterAccountFrozen`.

`setProvisioningComplete` stores `provisioningComplete` without an event. Transfers stay closed until that flag is true.

A frozen withdrawal returns the account's esGMX and pair collateral. `_settle` returns immediately while the account is frozen, and `_withdraw` then sets `trancheStartIndex` to the tranche length. Vesting time since the last settle is not recorded as claimable GMX, and those tranches stay closed after an unfreeze. The comment that withheld conversion lands on unfreeze applies while the tranches remain open.

RECOMMENDATION

Be aware of and document this behavior: a frozen `claimable` balance cannot be claimed, a frozen withdrawal returns esGMX and drops vesting time since the last settle, an uncapped transfer event reports a max cap that was not moved, and provisioning completion is not evented.

RESOLUTION

Status: Acknowledged.

GMX Team: Acknowledged. No tokens are lost in any of the cases: both vesters are deployed capped, provisioningComplete is a public flag set once, and a withdrawal while frozen returns the unconverted esGMX and pair. The frozen-account behaviour is documented in our integrator guide.

§ 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/gmx/rewardrouterv3-and-rewardrouterutils-2026-11-26.