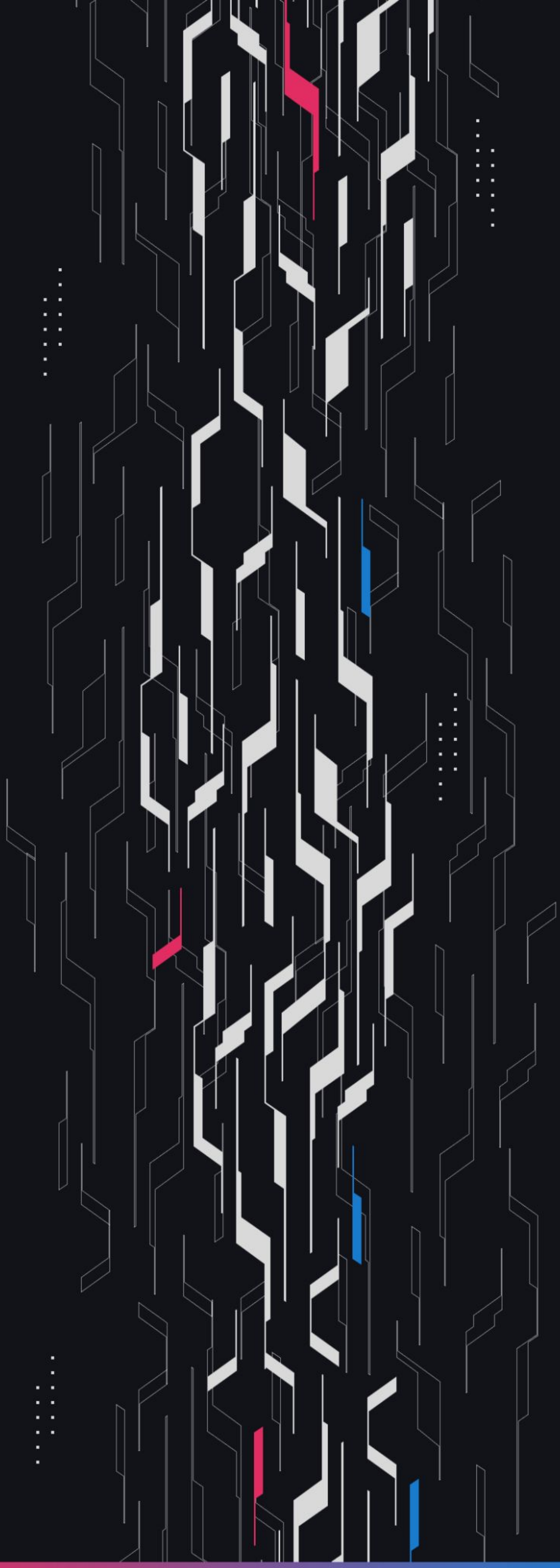
GUARDIAN

GMX

GMX Crosschain V2.2 7

Security Assessment

July 28th, 2025



Summary

Audit Firm Guardian

Prepared By Owen Thurm, Daniel Gelfand

Client Firm GMX

Final Report Date July 28, 2025

Audit Summary

GMX engaged Guardian to review the security of their GMX Crosschain architecture. From the 21st of July to the 25th of July, a team of 2 auditors reviewed the source code in scope. All findings have been recorded in the following report.

Confidence Ranking

Given the lack of critical issues detected and minimal code changes following the main review, Guardian assigns a Confidence Ranking of 5 to the protocol. Guardian advises the protocol to consider periodic review with future changes. For detailed understanding of the Guardian Confidence Ranking, please see the rubric on the following page.

🔗 Blockchain network: **Arbitrum, Avalanche**

✅ Verify the authenticity of this report on Guardian’s GitHub: https://github.com/guardianaudits

📊 Code coverage & PoC test suite: https://github.com/GuardianOrg/gmx-syntheticsgmxcrosschain3-fuzz

Guardian Confidence Ranking

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. Low risk of latent critical issues.	0 High/Critical findings and few Low/Medium severity findings.
4: High Confidence	Code is clean, well-structured, and adheres to best practices. Only Low or Medium-severity issues were discovered. Design patterns are sound, and test coverage is reasonable. Small changes, such as modifying rounding logic, may introduce new vulnerabilities and should be carefully reviewed. **Recommendation:** Suitable for deployment after remediations; consider periodic review with changes.	0 High/Critical findings. Varied Low/Medium severity findings.
3: Moderate Confidence	Medium-severity and occasional High-severity issues found. Code is functional, but there are concerning areas (e.g., weak modularity, risky patterns). No critical design flaws, though some patterns could lead to issues in edge cases. **Recommendation:** Address issues thoroughly and consider a targeted follow-up audit depending on code changes.	1 High finding and ≥ 3 Medium. Varied Low severity findings.
2: Low Confidence	Code shows frequent emergence of Critical/High vulnerabilities (~2/week). Audit revealed recurring anti-patterns, weak test coverage, or unclear logic. These characteristics suggest a high likelihood of latent issues. **Recommendation:** Post-audit development and a second audit cycle are strongly advised.	2-4 High/Critical findings per engagement week.
1: Very Low Confidence	Code has systemic issues. Multiple High/Critical findings (≥5/week), poor security posture, and design flaws that introduce compounding risks. Safety cannot be assured. **Recommendation:** Halt deployment and seek a comprehensive re-audit after substantial refactoring.	≥5 High/Critical findings and overall systemic flaws.

Table of Contents

Project Information

Smart Contract Risk Assessment

Addendum

Project Overview

Project Summary

Project Name	GMX
Language	Solidity
Codebase	https://github.com/gmx-io/gmx-synthetics/tree/main/contracts
Commit(s)	Initial commit: c738aeba644aa1400de1279b28aa65d08a729afe Final commit: 57de8c6ab71d1af42d3a198c782f1a0fc0a0c669

Audit Summary

Delivery Date	July 28, 2025
Audit Methodology	Static Analysis, Manual Review, Test Suite, Contract Fuzzing

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	1	0	0	0	0	1
● Medium	0	0	0	0	0	0
● Low	8	0	0	5	0	3
● Info	0	0	0	0	0	0

Audit Scope & Methodology

Vulnerability Classifications

Severity	Impact: *High*	Impact: *Medium*	Impact: *Low*
Likelihood: *High*	● Critical	● High	● Medium
Likelihood: *Medium*	● High	● Medium	● Low
Likelihood: *Low*	● Medium	● Low	● Low

Impact

High Significant loss of assets in the protocol, significant harm to a group of users, or a core functionality of the protocol is disrupted.

Medium A small amount of funds can be lost or ancillary functionality of the protocol is affected. The user or protocol may experience reduced or delayed receipt of intended funds.

Low Can lead to any unexpected behavior with some of the protocol's functionalities that is notable but does not meet the criteria for a higher severity.

Likelihood

High The attack is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount gained or the disruption to the protocol.

Medium An attack vector that is only possible in uncommon cases or requires a large amount of capital to exercise relative to the amount gained or the disruption to the protocol.

Low Unlikely to ever occur in production.

Audit Scope & Methodology

Methodology

Guardian is the ultimate standard for Smart Contract security. An engagement with Guardian entails the following:

- Two competing teams of Guardian security researchers performing an independent review.
- A dedicated fuzzing engineer to construct a comprehensive stateful fuzzing suite for the project.
- An engagement lead security researcher coordinating the 2 teams, performing their own analysis, relaying findings to the client, and orchestrating the testing/verification efforts.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts. Comprehensive written tests as a part of a code coverage testing suite.
- Contract fuzzing for increased attack resilience.

Findings & Resolutions

ID	Title	Category	Severity	Status
H-01	Secondary Amount Is Not Bridged Out	Logical Error	● High	Resolved
L-01	Some Tokens Incompatible With Edge Oracle	Warning	● Low	Acknowledged
L-02	Max Data List Length Warning	Configuration	● Low	Acknowledged
L-03	Lacking From And To Validations	Validation	● Low	Resolved
L-04	Misleading Account Emitted	Events	● Low	Resolved
L-05	Terms Should Be Set Before Depositing	Warning	● Low	Acknowledged
L-06	Lacking Signature Deadline Or Nonce	Validation	● Low	Acknowledged
L-07	Lacking Domain Separator	Best Practices	● Low	Resolved
L-08	Min Amount Is Altered	Unexpected Behaviour	● Low	Acknowledged

H-01 | Secondary Amount Is Not Bridged Out

Category	Severity	Location	Status
Logical Error	● High	BridgeOutFromControllerUtils.sol: 142	Resolved

Description

In the bridgeOutFromController function for tokens and secondaryTokens the _bridgeOutParams.amount is assigned as only the params.amount in the first case, ignoring the secondaryAmount entirely.

This misses the secondary token amount that should be bridged out in the event that both output tokens are the same token.

Recommendation

Assign the _bridgeOutParams.amount as params.amount + params.secondaryAmount in the first case.

Resolution

GMX Team: Resolved.

L-01 | Some Tokens Incompatible With Edge Oracle

Category	Severity	Location	Status
Warning	● Low	EdgeDataStreamProvider.sol	Acknowledged

Description

Depending on the token decimals and corresponding edge oracle decimals there may be some tokens which cannot be used with the edge oracle due to the token decimals multiplier calculations.

For example:

- Token has 18 decimals
- Expo is -14
- `floatMultiplier` = 30 - 18 - 14 = -2

A resulting negative multiplier reverts with the `InvalidEdgeDataStreamExpo` error.

Recommendation

Consider if tokens which have a net negative float multiplier due to high decimals or a high exponent should be supported by using a division of `10^(-floatMultiplier)`. Otherwise be aware of this incompatibility for exotic tokens.

Resolution

GMX Team: Acknowledged.

L-02 | Max Data List Length Warning

Category	Severity	Location	Status
Configuration	● Low	Global	Acknowledged

Description

With the addition of the minimum output amount configuration for the user, the maximum datalist length should be increased as necessary to be able to include a minimum amount out value for both the primary output token and secondary output token if there is one for the action.

Recommendation

Be sure that the largest required data list length can be supported by the validation.

Resolution

GMX Team: Acknowledged.

L-03 | Lacking From And To Validations

Category	Severity	Location	Status
Validation	● Low	ClaimHandler.sol	Resolved

Description

In the transferClaim function there is no validation preventing the from and to addresses from being the same account. This may lead to unexpected issues and should be explicitly prevented to avoid mistakes.

Recommendation

Consider adding validation to ensure that the from and to addresses of each TransferClaimParam are unique.

Resolution

GMX Team: Resolved.

L-04 | Misleading Account Emitted

Category	Severity	Location	Status
Events	● Low	ClaimHandler.sol	Resolved

Description

In the `emitClaimFundsClaimed` invocation in the `claimFunds` function the receiver is emitted as the account, however the `msg.sender` is the account that claimed funds.

This may be misleading to consumers of the `emitClaimFundsClaimed` emission.

Recommendation

Consider using the `msg.sender` as the account field for the `emitClaimFundsClaimed` invocation.

Resolution

GMX Team: Resolved.

L-05 | Terms Should Be Set Before Depositing

Category	Severity	Location	Status
Warning	● Low	ClaimHandler.sol	Acknowledged

Description

The setTerms function should be called for a given distributionId that requires terms before the depositFunds function is used for that distributionId.

This is because it is possible for an actor to withdraw their distribution immediately after the depositFunds function is used, without signing any terms if they are not configured at that point.

Recommendation

Be sure to call the setTerms function before the depositFunds function for any distributions which require terms.

Resolution

GMX Team: Acknowledged.

L-06 | Lacking Signature Deadline Or Nonce

Category	Severity	Location	Status
Validation	● Low	ClaimHandler.sol	Acknowledged

Description

The validateTermsSignature validation does not include any deadline for when the signature made by the user should become invalid, or Nonce which makes signature uses unique.

This may be unexpected especially if a user makes an initial claim for a distributionId before their allocation for that distributionId is increased and they are able to claim again.

Recommendation

Consider adding mechanisms that more strictly define the use of a user’s signature such as a nonce and deadline.

Resolution

GMX Team: Acknowledged.

L-07 | Lacking Domain Separator

Category	Severity	Location	Status
Best Practices	● Low	ClaimHandler.sol	Resolved

Description

There is no domain separator included in the validateTermsSignature signature validation. This means a signature of terms could be potentially used across multiple contracts or even on other chains if a ClaimHandler were to be deployed on multiple networks.

Recommendation

Consider adding a domain separator to make signatures specific to the context in which they are meant to be used.

Resolution

GMX Team: Resolved.

L-08 | Min Amount Is Altered

Category	Severity	Location	Status
Unexpected Behaviour	● Low	LayerZeroProvider.sol	Acknowledged

Description

In the prepareSend function the sendParam.minAmountLD is re-assigned to receipt.amountReceivedLD.

Therefore if for any reason the amount received by the user on the stargate.send invocation would be less than the receipt.amountReceivedLD but more than the sendParam.minAmountLD then the send would fail even though the minimum amount specified by the user could have been met.

Recommendation

This case should not be possible since the quoteOFT call should always return the exact result of the send.

However consider if the minAmountLD should be left as the user's specified sendParam.minAmountLD so that this case cannot arise with any underlying stargate pool or OFT implementation.

Resolution

GMX Team: Acknowledged.

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.

About Guardian

Founded in 2022 by DeFi experts, Guardian is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian upholds best-in-class security while achieving our mission to relentlessly secure DeFi.

To learn more, visit https://guardianaudits.com

To view our audit portfolio, visit https://github.com/guardianaudits

To book an audit, message https://t.me/guardianaudits