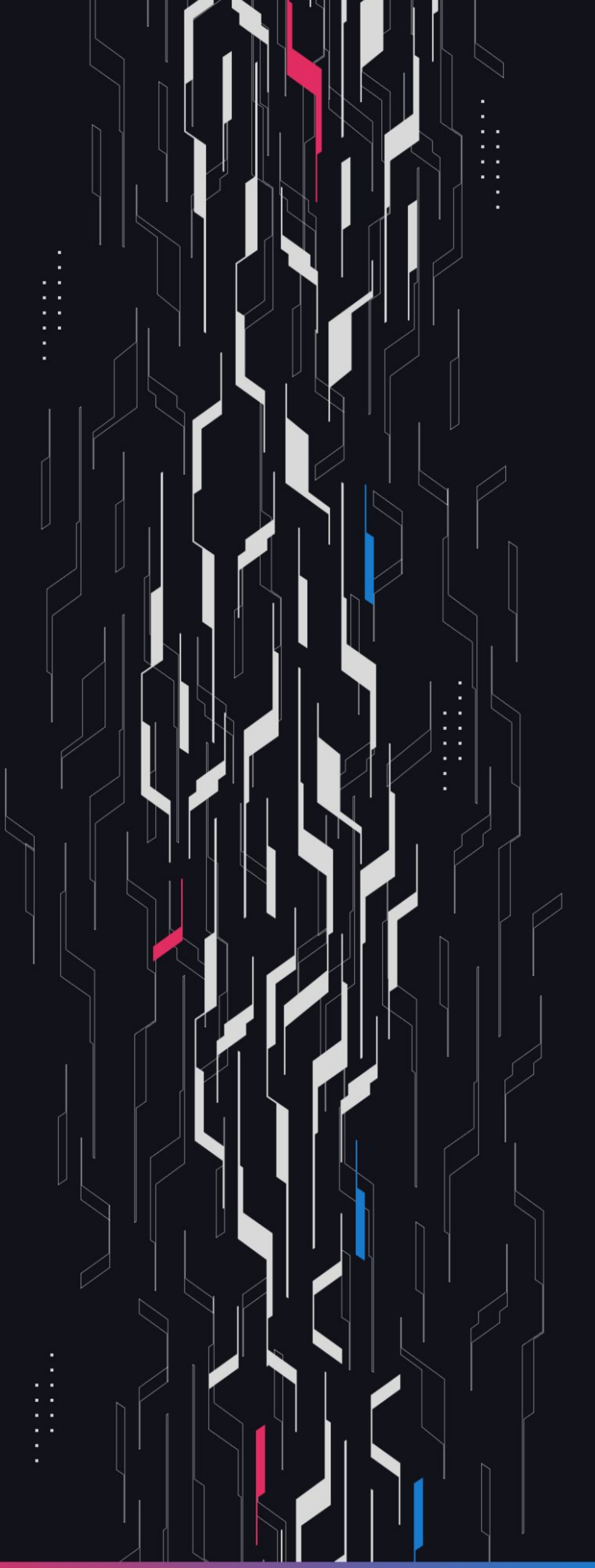
GUARDIAN

GMX

ConfigSyncer

Security Assessment

September 4th, 2024



Summary

Audit Firm Guardian

Prepared By Daniel Gelfand, Owen Thurm

Client Firm GMX

Final Report Date September 4, 2024

Audit Summary

GMX engaged Guardian to review the security of its ConfigSyncer contract which allows for a streamlined market parameter update process. From the 19th of August to the 26th of August, a team of 2 auditors reviewed the source code in scope. All findings have been recorded in the following report.

For a detailed understanding of risk severity, source code vulnerability, and potential attack vectors, refer to the complete audit report below.

🔗 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/gmx-io/gmx-synthetics/pull/31

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
Commit(s)	d596c9874b5e3dd95f9dd7c0851c74addfc36882

Audit Summary

Delivery Date	September 4, 2024
Audit Methodology	Static Analysis, Manual Review, Test Suite

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	0	0	0	0	0	0
● Medium	0	0	0	0	0	0
● Low	5	0	0	0	0	5

Audit Scope & Methodology

Vulnerability Classifications

Vulnerability Level	Classification
● Critical	Easily exploitable by anyone, causing loss/manipulation of assets or data.
● High	Arduously exploitable by a subset of addresses, causing loss/manipulation of assets or data.
● Medium	Inherent risk of future exploits that may or may not impact the smart contract execution.
● Low	Minor deviation from best practices.

Methodology

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
L-01	Duplicate allUpdateTypes Entries	Validation	● Low	Resolved
L-02	Incorrect previousValue Stored	Logical Error	● Low	Resolved
L-03	getLatestUpdateByType DoS	DoS	● Low	Resolved
L-04	Lacking Market Validation	Validation	● Low	Resolved
L-05	General Key Risk	Validation	● Low	Resolved

L-01 | Duplicate allUpdateTypes Entries

Category	Severity	Location	Status
Validation	● Low	RiskOracle.sol: 59	Resolved

Description

In the constructor for the RiskOracle contract on Arbitrum Sepolia https://sepolia.arbiscan.io/address/0x526d6789fCb503F2F898f45912A7a24fe9dd48e4#code, a list of initialUpdateTypes may be passed where there are duplicate updateType strings.

In this case the updateType string will be pushed to the allUpdateTypes list multiple times. This can lead to unexpected behavior for any systems relying on the allUpdateTypes list as it will have duplicate entries.

Recommendation

Consider validating that no duplicate entries have been made for the updateTypes in the RiskOracle constructor.

Resolution

GMX Team: Resolved.

L-02 | Incorrect previousValue Stored

Category	Severity	Location	Status
Logical Error	● Low	RiskOracle.sol: 151	Resolved

Description

When creating the newUpdate object to be stored the previousValue is declared as the direct previous update no matter what market or updateType the previous update acted upon.

This is incorrect or at least misleading as the previousValue serves little to no value if it merely references the previous update of any market/parameter.

Instead the previousValue should pertain to the previous value of the market and updateType combination.

Recommendation

Query the last update for the particular market and parameter type via the latestUpdateIdByMarketAndType mapping and store this value as the previousValue.

Resolution

GMX Team: Resolved.

L-03 | getLatestUpdateByType DoS

Category	Severity	Location	Status
DoS	● Low	RiskOracle.sol: 170	Resolved

Description

The getLatestUpdateByType function loops backwards through the updateHistory to find the most recent matching entry for the updateType.

This can lead to an out of gas DoS for integrating contracts if a particular updateType has been performed before many other subsequent updates.

Recommendation

Be aware of this risk and avoid using the getLatestUpdateByType function in a Smart Contract. Otherwise consider refactoring the RiskOracle such that it stores the latestUpdateByType in a mapping so it can be easily queried.

Resolution

GMX Team: Resolved.

L-04 | Lacking Market Validation

Category	Severity	Location	Status
Validation	● Low	Global	Resolved

Description

Throughout the GMX contracts a common pattern for configuration keys is to include the market which is being targeted in the additional key data.

In the RiskOracle and ConfigSyncer it is assumed that the market which an update is supplied for is the one which provided additionalData includes, however there is no validation to enforce this.

Recommendation

Be aware of this risk and put in place validations off-chain such that trusted parties will not commit updates for markets which are not included in the additionalData provided.

Otherwise consider implementing validations at the contract level which do not allow market configurations which do not agree with the market address provided in the additionalData.

Though this is likely not realistic to validate on-chain due to the many existing arbitrary key datas + new keys which will be introduced in the future.

Resolution

GMX Team: The issue was resolved in commit e9807e2.

L-05 | General Key Risk

Category	Severity	Location	Status
Validation	● Low	Config.sol	Resolved

Description

The following keys are not validated in the _validateRange function and belong to uint values which can potentially DoS or otherwise cause loss or harm to GMX V2 users.

- MAX_SWAP_PATH_LENGTH
- MIN_POSITION_SIZE_USD
- MAX_POSITION_IMPACT_FACTOR_FOR_LIQUIDATIONS
- MAX_ORACLE_PRICE_AGE
- MAX_ORACLE_TIMESTAMP_RANGE
- ORACLE_TIMESTAMP_ADJUSTMENT
- MAX_ORACLE_REF_PRICE_DEVIATION_FACTOR
- REQUEST_EXPIRATION_TIME
- MIN_COLLATERAL_FACTOR_FOR_OPEN_INTEREST_MULTIPLIER
- POSITION_IMPACT_FACTOR
- MAX_POSITION_IMPACT_FACTOR
- SWAP_IMPACT_FACTOR
- MAX_AUTO_CANCEL_ORDERS
- RESERVE_FACTOR
- OPEN_INTEREST_RESERVE_FACTOR
- MIN_FUNDING_FACTOR_PER_SECOND
- THRESHOLD_FOR_STABLE_FUNDING
- THRESHOLD_FOR_DECREASE_FUNDING
- PRICE_FEED_HEARTBEAT_DURATION
- All gas related keys

Recommendation

Carefully consider the risk of each of these keys being configured by the RiskOracle without any range validation. Where appropriate add the corresponding validations that the configured values are within an expected range.

Resolution

GMX Team: Resolved.

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 Audits

Founded in 2022 by DeFi experts, Guardian Audits is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian Audits 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