GA GUARDIAN

GMX

Pro tiers

Security Assessment

November 18th, 2024

Summary

Audit Firm Guardian

Prepared By Daniel Gelfand, Owen Thurm

Client Firm GMX

Final Report Date November 18, 2024

Audit Summary

GMX engaged Guardian to review the security of its pro-tiers update to GMXV2. From the 30th of September to the 7th of October, 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

Table of Contents

Project Overview

Project Summary

Project Name	GMX
Language	Solidity
Codebase	https://github.com/gmx-io/gmx-synthetics
Commit(s)	Initial commit: 887b2b55e2fb5f1f4f53e00efcdc8df497fc41f5 Final commit: a3950ca6c54c8015df5aab477dcbb47bc6985729

Audit Summary

Delivery Date	November 18, 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	1	0	0	0	0	1
Medium	2	0	0	0	0	2
Low	11	0	0	7	0	4

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	validFromTime Risk Free Trades	Gaming	High	Resolved
M-01	Gas Multiplier Fee Key Errantly Removed	Validation	Medium	Resolved
M-02	Order Funds May Be Lost Upon Cancellation	Unexpected Behavior	Medium	Resolved
L-01	Integrations Broken By Deposit Gas Update	Warning	Low	Acknowledged
L-02	Nonzero validFrom Allowed For Market Orders	Validation	Low	Resolved
L-03	Dangerous maxFundingFactorPerSecond Configuration	Validation	Low	Resolved
L-04	validFromTime Orders Executed Before Their Date	Warning	Low	Acknowledged
L-05	Liquidation Fee Uses Round Down Division	Rounding	Low	Resolved
L-06	Liquidation Fee Avoided	Gaming	Low	Acknowledged
L-07	New Deposit Gas Key Configuration	Warning	Low	Acknowledged
L-08	Users Required To Send Extraneous Gas	Logical Error	Low	Acknowledged
L-09	Inaccurate Comment	Documentation	Low	Resolved
L-10	Users May Provide Less Gas Than Necessary	Configuration	Low	Acknowledged

Findings & Resolutions

ID	Title	Category	Severity	Status
L-11	Funding Configuration Invalidates Pending Funding	Unexpected Behavior	Low	Acknowledged

H-01 | validFromTime Risk Free Trades

Category	Severity	Location	Status
Gaming	High	Global	Resolved

Description

The validFromTime attribute does not allow non-market orders to be executed until their validFromTime. This allows for the gaming of limit orders resulting in risk free trades over short time periods.

Consider the following scenario for a LimitIncrease order:

```
 orderCreation  price goes below trigger       validFromTime
------|------------------|------|--------------------|----|-------------------------->
                      price goes above trigger         5 minutes after price goes above trigger
```

In this scenario the LimitIncrease order can be executed at the validFromTime with the price from when the trigger was satisfied, even though the current market price at the validFromTime can be noticeably above the trigger price.

This can all happen within a 5 minute timeframe, staying within the maxPriceAge. After the LimitIncrease order is executed with a stale price, the user can immediately create a MarketDecrease order to lock in their risk free profit.

If the price does not move in the way the user would like to achieve this risk free trade, they can simply cancel their limit order or update their validFromTime to try again during the next period.

Recommendation

Require that non-market orders be executed with prices only after the validFromTime.

Resolution

GMX Team: Resolved.

M-01 | Gas Multiplier Fee Key Errantly Removed

Category	Severity	Location	Status
Validation	Medium	Config.sol: 542	Resolved

Description

The `EXECUTION_GAS_FEE_MULTIPLIER_FACTOR` key was errantly removed from the `allowedLimitedBaseKeys`, preventing the limited config keeper from being able to configure this value. The value can still be configured by the normal config keeper however.

Recommendation

Add the `allowedLimitedBaseKeys[Keys.EXECUTION_GAS_FEE_MULTIPLIER_FACTOR] = true;` line back to the `_initAllowedLimitedBaseKeys` function and remove the duplicated `allowedLimitedBaseKeys[Keys.EXECUTION_GAS_FEE_PER_ORACLE_PRICE] = true;` line.

Resolution

GMX Team: Resolved.

M-02 | Order Funds May Be Lost Upon Cancellation

Category	Severity	Location	Status
Unexpected Behavior	Medium	Global	Resolved

Description

In the deployment plan for the new contract updates the old version of the GMX contracts will be live at the same time as the new version.

The keepers will be configured to execute using the new version of the contracts, however `StopIncrease` orders which are manually cancelled by users and integrations through the old version of the contracts will experience complete loss of funds.

This occurs because the new `StopIncrease` order type will not register as an increase order in the old version of the contracts.

Recommendation

Be aware of this risk and warn users and integrations of this potential loss.

Resolution

GMX Team: Resolved.

L-01 | Integrations Broken By Deposit Gas Update

Category	Severity	Location	Status
Warning	Low	GasUtils.sol	Acknowledged

Description

The `estimateExecuteDepositGasLimit` function has been updated so that single sided deposits use the same gas expenditure as double sided gas deposits.

Additionally the `depositGasLimitKey` has been changed to no longer accept a boolean indicating whether it is a single sided deposit.

These changes will likely cause issues with integrators if they are not informed of the updates.

Recommendation

Consider informing integrators of these changes.

Resolution

GMX Team: Acknowledged.

L-02 | Nonzero validFrom Allowed For Market Orders

Category	Severity	Location	Status
Validation	Low	OrderUtils.sol: 135	Resolved

Description

During order creation in the `OrderUtils.createOrder` function there is no validation preventing users from assigning a nonzero `validFrom` value for market orders.

However the `validFrom` field will have no effect on Market orders. This may result in user's assigning a `validFrom` field for a market order which will then not apply upon order execution.

Recommendation

Consider whether the `validFrom` field should be validated to be exactly zero when a market order is created to avoid any unexpected behavior.

Resolution

GMX Team: Resolved.

L-03 | Dangerous maxFundingFactorPerSecond Configuration

Category	Severity	Location	Status
Validation	Low	Config.sol	Resolved

Description

The limited config keeper is now allowed to configure the `MAX_FUNDING_FACTOR_PER_SECOND` key as it is assigned to `true` in the `allowedLimitedBaseKeys` mapping.

This can be dangerous if the limited config keeper assigns the max funding fee to less than the minimum funding fee which will lead to unexpected results for the funding calculations of a market.

Recommendation

Consider adding validation to the `_validateRange` function that validates that the max funding factor per second for a market is above the min funding factor per second whenever either the min or max funding factor per seconds are updated.

Resolution

GMX Team: Resolved.

L-04 | validFromTime Orders Executed Before Their Date

Category	Severity	Location	Status
Warning	Low	Global	Acknowledged

Description

In the deployment plan for the new contract updates the old version of the GMX contracts will be live at the same time as the new version.

There is a risk that any orders executed through the old contracts would ignore the new `validFromTime` attribute and execute orders before that specified time.

However the keepers will be configured to execute using the new contracts. This finding simply serves as a warning to document this risk.

Recommendation

Be aware of this risk and ensure that all order executions occur through the new contracts.

Resolution

GMX Team: Acknowledged.

L-05 | Liquidation Fee Uses Round Down Division

Category	Severity	Location	Status
Rounding	Low	PositionPricingUtils.sol: 575	Resolved

Description

Throughout the GMX V2 codebase roundup division is used where it is in the protocol's favor and against the favor of the user. However when computing the liquidation fee to charge the amount is computed using round down division.

Recommendation

Use round up division to compute the `liquidationFees.liquidationFeeAmount`.

Resolution

GMX Team: Resolved.

L-06 | Liquidation Fee Avoided

Category	Severity	Location	Status
Gaming	Low	Global	Acknowledged

Description

A liquidation fee is now charged by the protocol to monetize liquidations, however this fee can be avoided by setting a stop loss order above the liquidation price to close a position right before the liquidation point is reached.

Recommendation

Be aware of this gaming and consider if any actions should be done to penalize users who utilize this strategy.

Resolution

GMX Team: Acknowledged.

L-07 | New Deposit Gas Key Configuration

Category	Severity	Location	Status
Warning	Low	Config.sol	Acknowledged

Description

The deposit gas key has been changed and must be configured again through the config contract.

Recommendation

This finding merely serves as a reminder for this. Be sure to populate the new deposit key value in the `dataStore` through the config contract upon deployment.

Resolution

GMX Team: Acknowledged.

L-08 | Users Required To Send Extraneous Gas

Category	Severity	Location	Status
Logical Error	Low	OrderUtils.sol: 176	Acknowledged

Description

In the manual user initiated order cancellation flow the `minHandleExecutionErrorGas` is validated in the `cancelOrder` function.

Thus users are required to provide this gas amount even though they are not handling an execution error. This may cause unintended gas estimation problems and reverts.

Recommendation

Consider only requiring that this minimum execution gas is provided while executing an order/autocancelling and not while a user is manually cancelling their order.

The user still will not be able to provide less than the required callback gas as the `afterOrderCancellation` function will validate the remaining gas for the callback.

Resolution

GMX Team: Acknowledged.

L-09 | Inaccurate Comment

Category	Severity	Location	Status
Documentation	Low	Order.sol: 24	Resolved

Description

In the comment for the `MarketDecrease` order type it is now mentioned that the order will be frozen if the acceptable price is not reached. However market orders cannot be frozen, instead they will be cancelled.

Recommendation

Revert the comment to say that the order will be cancelled.

Resolution

GMX Team: Resolved.

L-10 | Users May Provide Less Gas Than Necessary

Category	Severity	Location	Status
Configuration	Low	Global	Acknowledged

Description

The deposit gas limit key is no longer dependent on the amount of distinct tokens being deposited, therefore users and integrations creating deposits through the old contracts will have a different deposit gas execution fee to pay than what is expected by the keepers.

For instance, the new deposit gas requirement is likely to be higher than the single token deposit configuration. In this case users and integrations submitting orders through the old contracts will be required to send less gas than the new contracts would require.

This may be unexpected for keepers and cause them to run a lower margin or even a deficit on some of these orders.

Recommendation

Be aware of this potentially unexpected behavior. If necessary, configure the old single token and double token gas key values to be the same as the new single gas key value to match the gas requirements across both versions.

Resolution

GMX Team: Acknowledged.

L-11 | Funding Configuration Invalidates Pending Funding

Category	Severity	Location	Status
Unexpected Behavior	Low	Config.sol	Acknowledged

Description

The limited config keeper is now allowed to configure the `MAX_FUNDING_FACTOR_PER_SECOND` key as it is assigned to `true` in the `allowedLimitedBaseKeys` mapping.

However the adjustment of the max funding factor per second will in many cases change the funding amount which was pending for the market.

This is because capping the maximum to a higher or lower value will change the resulting `fundingFactorPerSecond` which will apply over the past `[lastMarketUpdate, block.timestamp]` period.

Integrations with GMX V2 often rely on the current funding values to measure the value of open positions on GMX. Thus this measurement may become retroactively invalidated once the max funding factor per second is assigned to a different value.

This can happen if the limited config keeper is performing maliciously but also if they are performing honest updates to the max funding factor per second assignment for a market.

Additionally this applies to the maximum and minimum funding factor per second configurations which can be made by the normal config keeper.

Recommendation

Consider updating the funding in a market before changing the maximum or minimum thresholds for the funding factor per second.

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 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