GUARDIAN AUDITS

SMART CONTRACT SECURITY AUDIT OF

Chainlink

Summary

Audit Firm Guardian

Prepared By Owen Thurm, Daniel Gelfand, 0xKato

Client Firm Chainlink

Final Report Date September 20, 2023

Audit Summary

Chainlink engaged Guardian to review the security of its real time pricing automation system for GMX V2. From the 4th of September to the 18th of September, a team of 3 auditors reviewed the source code in scope. All findings have been recorded in the following report.

Notice that the examined smart contracts are not resistant to internal exploit. 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	Chainlink
Language	Solidity
Codebase	https://github.com/Cyfrin/chainlink-gmx-automation
Commit(s)	6db7db623c6e04859e1dc653dcd0fd53830b73ff

Audit Summary

Delivery Date	September 20, 2023
Audit Methodology	Static Analysis, Manual Review, Test Suite, Contract Fuzzing

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	2	0	0	0	0	2
High	1	0	0	0	0	1
Medium	2	0	0	1	0	1
Low	8	0	0	4	0	4

Audit Scope & Methodology

ID	File	SHA-1 Checksum(s)
BASE	GMXAutomationBase.sol	2e76a9ca0183c020c96a21c2993e2ea96f529e8a
MKTA	MarketAutomation.sol	9c5238560f74c24e9034b254173673bace389ee8
DEPA	DepositAutomation.sol	7e81d1de9de415b69324513d202fed985674f7b6
WTDA	WithdrawalAutomation.sol	66834d526af3a2ee3e07904180ae8cc6d337c117
LGED	LibGMXEventLogDecoder.sol	dc8183ef3d9d7a3be4d26cdd34face94a52da8d3

Audit Scope & Methodology

Vulnerability Classifications

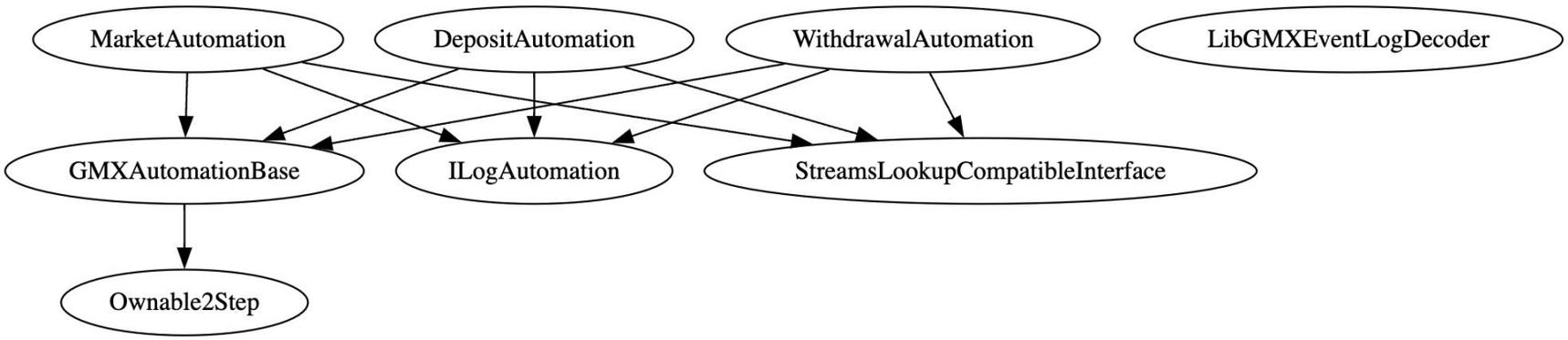
Vulnerability Level	Classification
🔴 Critical	Easily exploitable by anyone, causing loss/manipulation of assets or data.
🟠 High	Arduously exploitable by a subset of addresses, causing loss/manipulation of assets or data.
🟡 Medium	Inherent risk of future exploits that may or may not impact the smart contract 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.

Inheritance Graph



Invariants Assessed

During Guardian's review of Chainlink's real time pricing automation system, differential fuzz-testing with Foundry was performed on a key function.

Throughout the engagement the following invariant was assessed for a total of 1,000,000+ runs with a prepared Foundry fuzzing suite.

ID	Description	Definition	Run Count
BASE-1	_toHexString Matches OpenZeppelin's toHexString output with a length of 32 bytes	_toHexString(bytes32(uint)) == Strings.toHexString(uint, 32)	1,000,000+

Findings & Resolutions

ID	Title	Category	Severity	Status
GLOBAL-1	Orders Requiring More Than 5 Million Gas Cannot Execute	Insufficient Gas	Critical	Resolved
WTDA-1	Withdrawals With Swaps Are Incompatible	Logical Error	Critical	Resolved
GLOBAL-2	Anyone May performUpkeep	Access Control	High	Resolved
MKTA-1	performUpkeep Can Be Used To Execute Any orderType	Validation	Medium	Acknowledged
GLOBAL-3	FeedIds Must Be Set For Every Token	Configuration	Medium	Resolved
MKTA-2	Typo	Typo	Low	Resolved
GLOBAL-4	Accept Ether Optimization	Optimization	Low	Acknowledged
GLOBAL-5	Lack Of cannotExecute Modifier For checkLog	Validation	Low	Acknowledged
GLOBAL-6	Inaccurate Comment	Documentation	Low	Resolved
GLOBAL-7	Key Contract Addresses May Not Be Updated	Upgradeability	Low	Acknowledged
GLOBAL-8	Block Number Provided As Time	Documentation	Low	Acknowledged
BASE-1	Missing NatSpec	Documentation	Low	Resolved
BASE-2	Typo	Typo	Low	Resolved

GLOBAL-1 | Orders Requiring More Than 5 Million Gas Cannot Execute

Category	Severity	Location	Status
Insufficient Gas	● Critical	Global	Resolved

Description

The Chainlink automation keepers are configured to send 5,000,000 gas with each execution, however GMX deposits, withdrawals, and orders may often require more than 5,000,000 gas to execute.

In the general.ts configuration file the `increaseOrderGasLimit` and `decreaseOrderGasLimit` are configured to `4_000_000`, additionally the `minAdditionalGasForExecution` is configured to `1_000_000`. Therefore any increase or decrease order including a callback with a nonzero `callbackGasLimit` cannot be executed by the Chainlink automation keepers.

Additionally the `singleSwapGasLimit` is configured to `1_000_000`, therefore deposits and withdrawals with 4 or more total swaps in the `longTokenSwapPath` and `shortTokenSwapPath` are unable to be executed by the Chainlink automation keepers.

Recommendation

Increase the configured gas amount to as high as 15,000,000 to ensure that even the most expensive of actions can be executed on GMX V2.

Resolution

Chainlink Team: The configuration was adjusted to 13,000,000 gas after discussing with the GMX team.

WTDA-1 | Withdrawals With Swaps Are Incompatible

Category	Severity	Location	Status
Logical Error	● Critical	WithdrawalAutomation.sol: 66	Resolved

Description

The `longTokenSwapPath` and the `shortTokenSwapPath` of a withdrawal cannot be decoded. This is because `WithdrawalEventUtils.emitWithdrawalCreated` does not emit the swap paths on a Withdrawal.

Consequently, the withdrawal automation becomes unusable whenever a user requires a swap post-withdrawal.

Recommendation

Add the `longTokenSwapPath` and `shortTokenSwapPath` to the `WithdrawalCreated` event. Afterwards, modify `WithdrawalAutomation` to decode these paths and `_addPropsToMapping` to set the necessary feeds.

Resolution

GMX Team: The necessary event data was added in commit 3122a9.

Chainlink Team: The `longTokenSwapPath` and `shortTokenSwapPath` logic was added to the `WithdrawalAutomation` in commit fc35f03.

GLOBAL-2 | Anyone May performUpkeep

Category	Severity	Location	Status
Access Control	High	Global	Resolved

Description

There is no access control for the performUpkeep function, therefore any arbitrary address may execute an order. On a network with a public mempool, such as Avalanche, any arbitrary user may observe the Chainlink keeper's transaction and copy the performData to execute their own deposit, withdrawal, or order.

A malicious actor can therefore front-run the execution of other user's orders to manipulate price impact such that the actor stands to gain a profit at the user's detriment.

Recommendation

Add access controls to the performUpkeep function in the MarketAutomation, DepositAutomation, and WithdrawalAutomation contracts.

Once access controls are added to the performUpkeep function, it should be noted that the permissioned caller still holds the ability to decide execution ordering, controlling the price impact experienced by orders.

Resolution

Chainlink Team: The suggested onlyForwarder access control was added in commit 0a7edc2.

MKTA-1 | performUpkeep Can Be Used To Execute Any orderType

Category	Severity	Location	Status
Validation	Medium	MarketAutomation.sol: 128	Acknowledged

Description

Using the `performUpkeep` function, the Chainlink keeper may execute any order, even if the order is not a `MarketIncrease`, `MarketDecrease`, or `MarketSwap`.

However the `MarketAutomation` contract is explicitly designed to execute only market orders, therefore the scope of which orders can be executed with the `performUpkeep` function should be limited by validating the `orderType` from the `dataStore`.

Recommendation

Validate that the order being executed is indeed a `MarketIncrease`, `MarketDecrease`, or `MarketSwap`.

Resolution

Chainlink Team: Acknowledged.

GLOBAL-3 | FeedIds Must Be Set For Every Token

Category	Severity	Location	Status
Configuration	Medium	Global	Resolved

Description

Deposits, withdrawals and orders may use swap paths including any token supported on the GMX V2 platform.

Therefore every supported token must be assigned a valid functioning `feedId`, otherwise deposits, withdrawals and orders may be unable to be executed using the Chainlink keeper or may become cancelled unexpectedly.

Recommendation

Ensure that every token on the GMX V2 platform is supported by a valid `feedId`.

Additionally, implement validation in the `checkLog` function such that if a `feedId` is not configured for a token that is necessary for the action, the action is not executed by the Chainlink keeper and is instead executed by the default keeper.

Resolution

Chainlink Team: The recommended validation was added in commit a21b0b9.

MKTA-2 | Typo

Category	Severity	Location	Status
Typo	Low	MarketAutomation.sol: 51	Resolved

Description

The comment on line 51 reads `a feed lookup lookup` where the word `lookup` is repeated twice.

Recommendation

Remove the second instance of `lookup`.

Resolution

Chainlink Team: The recommendation was implemented in commit ffed912.

GLOBAL-4 | Accept Ether Optimization

Category	Severity	Location	Status
Optimization	● Low	Global	Acknowledged

Description

The `DepositAutomation`, `MarketAutomation`, and `WithdrawalAutomation` contracts have no `receive` function and therefore cannot receive gas remuneration from GMX V2 in native tokens.

GMX V2 handles this by re-wrapping the native tokens and sending them to the keeper. However the keeper must pay for this additional re-wrapping gas expenditure on every transaction.

Recommendation

To avoid this additional gas expenditure on every transaction, implement a `receive` function, with a way to retrieve the accumulated native tokens.

Resolution

Chainlink Team: Acknowledged.

GLOBAL-5 | Lack Of cannotExecute Modifier For checkLog

Category	Severity	Location	Status
Validation	● Low	Global	Acknowledged

Description

The `ILogAutomation` contract documentation mentions that a `cannotExecute` modifier should be added to the `checkLog` function to ensure it can never be errantly called in a transaction, only simulated.

However the `checkLog` functions in the `DepositAutomation`, `MarketAutomation`, and `WithdrawalAutomation` contracts lack any `cannotExecute` or similar modifier.

Recommendation

Consider whether a `cannotExecute` modifier should be added.

If so, implement a `cannotExecute` modifier in the `GMXAutomationBase` contract and add it to the `checkLog` functions in the `DepositAutomation`, `MarketAutomation`, and `WithdrawalAutomation` contracts.

Resolution

Chainlink Team: Acknowledged.

GLOBAL-6 | Inaccurate Comment

Category	Severity	Location	Status
Documentation	● Low	Global	Resolved

Description

In the `DepositAutomation` and `WithdrawalAutomation` contracts the comment on line 53 states `// Decode Event Log 1`, however both `"DepositCreated"` and `"WithdrawalCreated"` are emitted with Event Log 2.

Recommendation

Update the comment to `// Decode Event Log 2`.

Resolution

Chainlink Team: The recommendation was implemented in commit ffed912.

GLOBAL-7 | Key Contract Addresses May Not Be Updated

Category	Severity	Location	Status
Upgradeability	● Low	Global	Acknowledged

Description

In the `DepositAutomation`, `WithdrawalAutomation`, and `MarketAutomation` contracts the `i_depositHandler`, `i_withdrawalHandler`, and `i_orderHandler` variables are declared as `immutable`. Additionally, in the `GMXAutomationBase` contract the `i_dataStore` and `i_reader` variables are declared `immutable`.

However it is possible that the respective contracts are upgraded or replaced, meaning that the existing Chainlink Automation contract is no longer functional. Therefore a new Automation contract would need to be deployed and whitelisted as a valid keeper in the event of a handler contract upgrade.

Recommendation

Consider if the `owner` address should be able to update these contract addresses in the event of an upgrade.

Resolution

Chainlink Team: Redeployment of an Automation contract is preferred on the Chainlink operational side.

GLOBAL-8 | Block Number Provided As Time

Category	Severity	Location	Status
Documentation	● Low	Global	Acknowledged

Description

The `StreamsLookup` error labels the second to last parameter as `time`, it is unclear whether this is intended to be a timestamp.

However a block number is provided for the `time` in the `StreamsLookup` error in the `checkLog` function in each of the `DepositAutomation`, `WithdrawalAutomation`, and `MarketAutomation` contracts.

Recommendation

Verify whether the time parameter is intended to be a block number or timestamp, consider documenting what the time parameter represents in the `StreamsLookupCompatibleInterface` interface.

Resolution

Chainlink Team: This is expected.

BASE-1 | Missing NatSpec

Category	Severity	Location	Status
Documentation	Low	GMXAutomationBase.sol: 81	Resolved

Description

In the NatSpec for the `_flushMapping` function, the addresses returned value is omitted.

Recommendation

Add the addresses value to the NatSpec for the `_flushMapping` function.

Resolution

Chainlink Team: The recommendation was implemented in commit ffed912.

BASE-2 | Typo

Category	Severity	Location	Status
Typo	Low	GMXAutomationBase.sol: 101	Resolved

Description

In the `_toHexString` function, the comment on line 101 states `// Fixed buffer size for hexadecimal convertion`, however `conversion` is misspelled as `convertion`.

Recommendation

Replace `convertion` with `conversion`.

Resolution

Chainlink Team: The recommendation was implemented in commit ffed912.

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