



# SOLIDProof

*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC  
Development | Marketing**

MADE IN GERMANY

## LogX

# AUDIT

SECURITY ASSESSMENT

**03. August, 2023**

FOR

## LOGX



**SolidProof\_io**



**@solidproof\_io**

Introduction	3
Disclaimer	3
Project Overview	4
Summary	4
Social Medias	4
Audit Summary	5
File Overview	6
Imported packages	8
Components	9
Exposed Functions	9
Capabilities	10
Inheritance Graph	11
Audit Information	12
Vulnerability & Risk Level	12
Auditing Strategy and Techniques Applied	13
Methodology	13
Overall Security	14
Medium or higher issuesUpgradeability	14
Ownership	15
Ownership Privileges	16
Minting tokens	16
Burning tokens	17
Blacklist addresses	18
Fees and Tax	19
Lock User Funds	20
Centralization Privileges	21
Audit Results	23

## Introduction

[SolidProof.io](#) is a brand of the officially registered company MAKE Network GmbH, based in Germany. We're mainly focused on Blockchain Security such as Smart Contract Audits and KYC verification for project teams.

Solidproof.io assess potential security issues in the smart contracts implementations, review for potential inconsistencies between the code base and the whitepaper/documentation, and provide suggestions for improvement.

## Disclaimer

[SolidProof.io](#) reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc'...)

**SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.**

SolidProof.io Reports represent an extensive auditing 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. SolidProof's position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of the security or functionality of the technology we agree to analyze.

# Project Overview

## Summary

Project Name	LogX
Website	<a href="http://www.logx.trade">www.logx.trade</a>
About the project	LogX is a Decentralised Exchange for trading perpetuals with aggregated liquidity. LogX Aggregator combines liquidity on various platforms to give users the best trades - low fees, high liquidity, and most importantly, you don't have to maintain positions across various DEXes.
Chain	Arbitrum
Language	Solidity
Codebase Link	<b>MuxProxyFactory -</b> 0x7b4f5cA5b32419deC15C32884F273E288027E623 <b>GmxProxyFactory -</b> 0x575814E9838021e4BfA712524215c5B3Fb678783 <b>MuxTransparentUpgradeableProxy -</b> 0x68c574f7134f9Ef32b689C1BB923D21C20FA544F <b>GmxTransparentUpgradeableProxy -</b> 0x72788cdec71e8a9ca66d1d0da3a9dfbdbd6b44e8 <b>GmxAdapter -</b> 0x2ef139b7fea432e7e303e70457235ca4b01fa6a9 <b>Mux Adapter -</b> 0x78B2eEBeeE4C533f617cf2B9d48b9c065eF99826
Unit Tests	Provided

## Social Medias

Telegram	<a href="https://t.me/logx_announcements">https://t.me/logx_announcements</a>
Twitter	<a href="https://twitter.com/logx_trade?s=21&amp;t=6nKHmInqn5ROkRIKjZCTmA">https://twitter.com/logx_trade?s=21&amp;t=6nKHmInqn5ROkRIKjZCTmA</a>
Facebook	N/A
Instagram	N/A
Github	N/A
Reddit	N/A
Medium	N/A
Discord	<a href="https://discord.gg/logx">https://discord.gg/logx</a>
Youtube	N/A
TikTok	N/A
LinkedIn	N/A

## Audit Summary

Version	Delivery Date	Changelog
v1.0	03. August 2023	<ul style="list-style-type: none"> <li>• Layout Project</li> <li>• Automated- /Manual-Security Testing</li> <li>• Summary</li> </ul>

**Note** - The following audit report presents a comprehensive security analysis of the smart contract utilized in the project. This analysis did not include functional testing (or unit testing) of the contract/s logic. We cannot guarantee 100% logical correctness of the contract as it was not functionally tested by us.





## File Overview

The project provided us with the files that should be tested in the security assessment. This audit covered the following files listed below with an SHA-1 Hash.

File Name	SHA-1 Hash
src/muxProxyFactory/MuxStorage.sol	10589bb0d4cbae618049ceb6564442c5e09eeb18
src/muxProxyFactory/MuxProxyFactory.sol	145142b9b56213b0342444e7fe1393dd57d64036
src/muxProxyFactory/MuxProxyBeacon.sol	fc38b5803df911c78a1a87846656d328f1bc4054
src/muxProxyFactory/MuxProxyConfig.sol	074dc4b662379bb8fe13fbb3d13996641732eda3
src/interfaces/IGmxProxyFactory.sol	850f4b56b65c06137a7432090f0c00d5b4a289d9
src/interfaces/ ITransparentUpgradeableProxy.sol	1aabca3b1d39bc4e43760f6e13e721317945073e
src/interfaces/IMuxGetter.sol	281d4b873e5e78b0118767197edbdaaa0011013a
src/interfaces/IGmxPositionRouter.sol	422e9d4496ee3306ac4ce7180cc6e1458a158e97
src/interfaces/IMuxProxyFactory.sol	efb960486638886223bfef716d434f0b24ead37a
src/interfaces/IMuxOrderBook.sol	726aa329ab453ec0f91256022aa57638de2079d2
src/interfaces/IGmxRouter.sol	71fa16317a8e19f46d4736849a17b45448bf5341
src/interfaces/IGmxOrderBook.sol	fde6574d475d30e4116e98677d38a6805ac06c26
src/interfaces/IWETH.sol	22bfd6a6c722d022091f1ce8dddeb7a3860c8d93
src/interfaces/IMuxAggregator.sol	86d639a6fd9f72f7a74544510ca1360324502b7e
src/interfaces/IGmxVault.sol	45c46108c6d482af1517972679ecb47ff1f64b36



src/interfaces/IGmxAggregator.sol	08a95658234cef76c6c77c5a86ffe819f59bfa01
src/transparentUpgradeableProxy/TransparentUpgradeableProxy.sol	5fd07dfacbe068460e501f3bf4832bc084f4b40f
src/gmxProxyFactory/GmxProxyFactory.sol	b7b7ccc5aa7fa2e5f37f7e14853ea47af9327191
src/gmxProxyFactory/GmxProxyBeacon.sol	984bdf5986f2521e50ee86e127fb4c27c9e50c0c
src/gmxProxyFactory/GmxProxyConfig.sol	05297cb82c8a06686e0192ba41d1ad38b3fd4da6
src/gmxProxyFactory/GmxStorage.sol	bf453e1a88f259e4d2c5b35a4aa8fbded44200f6
src/components/ImplementationGuard.sol	a4b07fba4808fd60a006c18f52d702bc97bbdf60
src/aggregators/lib/LibMath.sol	5556a51bdebae1a1fdea4374789ff694b291506a
src/aggregators/lib/LibUtils.sol	989a70622d1c54a5ca26b91a11bed6d2ca10cbd3
src/aggregators/mux/Types.sol	dd0aab8d24cd82172eaa410887a0793f8bc638d9
src/aggregators/mux/lib/LibMux.sol	c624396717067907a2873b2537808a797f4f1554
src/aggregators/mux/MuxAdapter.sol	6a490cb0ce7c7cee4f54d760ee7a262d2812d345
src/aggregators/mux/Storage.sol	ba82878e1153d386a35ddd597b5c11d845a19dc3
src/aggregators/mux/Config.sol	d0173f451745e5b64719b11c13e6a1e4fa8c8f53
src/aggregators/mux/Position.sol	a3c47e548bcfe297dcde788f3286a040bbb6bb9d
src/aggregators/gmx/Types.sol	584fdfcd3ecf7f05f89c4099c42deb6905f624a7
src/aggregators/gmx/lib/LibGmx.sol	088a0fd4298d5d1ee1362fb9a3ee7e4ed625aff0
src/aggregators/gmx/Storage.sol	6c53d9a1d67f2a104343ddc6daa212356cdcf715



src/aggregators/gmx/Config.sol	a35e944bed1acae7bea75c0a496ddc2256124ede
src/aggregators/gmx/GmxAdapter.sol	a149b8554006ad36bca0eb26bed602019b735b2d
src/aggregators/gmx/Position.sol	6e5e102d424496b30892b7bcc395c0cf05c8a6ed

*Please note: Files with a different hash value than in this table have been modified after the security check, either intentionally or unintentionally. A different hash value may (but need not) be an indication of a changed state or potential vulnerability that was not the subject of this scan.*

## Imported packages

*Used code from other Frameworks/Smart Contracts (direct imports).*

N/A - Direct Imports are not available in the project. All the imports are used as files

**Note for Investors:** We only audited contracts mentioned in the scope above. All contracts related to the project apart from that are not a part of the audit, and we cannot comment on its security and are not responsible for it in any way.



## External/Public functions

External/public functions are functions that can be called from outside of a contract, i.e., they can be accessed by other contracts or external accounts on the blockchain. These functions are specified using the function declaration's external or public visibility modifier.

## State variables

State variables are variables that are stored on the blockchain as part of the contract's state. They are declared at the contract level and can be accessed and modified by any function within the contract. State variables can be defined with a visibility modifier, such as public, private, or internal, which determines the access level of the variable.

## Components

 <b>Contracts</b>	 <b>Libraries</b>	 <b>Interfaces</b>	 <b>Abstract</b>
18	4	12	0


## Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

 <b>Public</b>	 <b>Payable</b>
208	27



External	Internal	Private	Pure	View
204	231	6	30	134

## StateVariables

<b>Total</b>	 <b>Public</b>
48	0



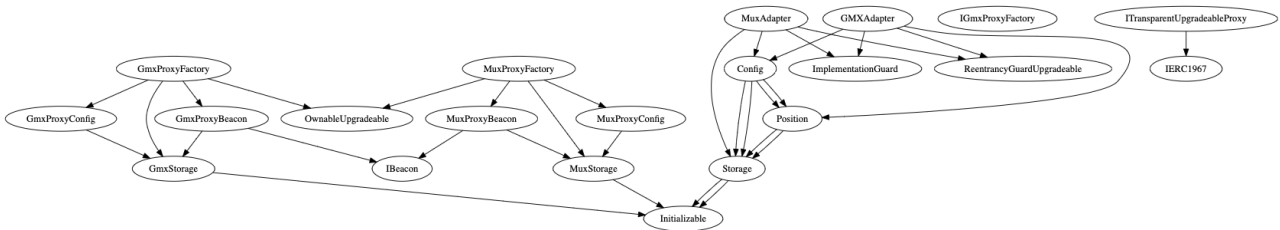
## Capabilities

Solidity Versions observed	 Uses Hash Functions	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts
^0.8.19 0.8.19	Yes	Yes	Yes	----



## Inheritance Graph

An inheritance graph is a graphical representation of the inheritance hierarchy among contracts. In object-oriented programming, inheritance is a mechanism that allows one class (or contract, in the case of Solidity) to inherit properties and methods from another class. It shows the relationships between different contracts and how they are related to each other through inheritance.



## Audit Information

### Vulnerability & Risk Level

Risk represents the probability that a certain source threat will exploit vulnerability and the impact of that event on the organization or system. The risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
<b>Critical</b>	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
<b>High</b>	7 - 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
<b>Medium</b>	4 - 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
<b>Low</b>	2 - 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
<b>Informational</b>	0 - 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

## Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to check the repository for security-related issues, code quality, and compliance with specifications and best practices. To this end, our team of experienced pen-testers and smart contract developers reviewed the code line by line and documented any issues discovered.

We check every file manually. We use automated tools only so that they help us achieve faster and better results.

## Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
  - a. Reviewing the specifications, sources, and instructions provided to SolidProof to ensure we understand the size, scope, and functionality of the smart contract.
  - b. Manual review of the code, i.e., reading the source code line by line to identify potential vulnerabilities.
  - c. Comparison to the specification, i.e., verifying that the code does what is described in the specifications, sources, and instructions provided to SolidProof.
2. Testing and automated analysis that includes the following:
  - a. Test coverage analysis determines whether test cases cover code and how much code is executed when those test cases are executed.
  - b. Symbolic execution, which is analysing a program to determine what inputs cause each part of a program to execute.
3. Review best practices, i.e., review smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on best practices, recommendations, and research from industry and academia.
4. Concrete, itemized and actionable recommendations to help you secure your smart contracts.

# Overall Security

## Medium or higher issuesUpgradeability

**Contract is an upgradeable**

**✗ Deployer can update the contract with new functionalities**

Description

The deployer can replace the old contract with a new one with new features. Be aware of this, because the owner can add new features that may have a negative impact on your investments.

Example

We assume that you have funds in the contract and it has been audited by any security audit firm. Now the audit has passed. After that, the deployer can upgrade the contract to allow him to transfer the funds you purchased without any approval from you. This has the consequence that your funds can be taken by the creator.

Comment

All contracts are upgradeable by nature.

## Ownership

**The ownership is not renounced**

**✗ The owner is not renounce**

Description

The owner has not renounced the ownership that means that the owner retains control over the contract's operations, including the ability to execute functions that may impact the contract's users or stakeholders. This can lead to several potential issues, including:

- Centralizations
- The owner has significant control over contract's operations

Comment

N/A

**Note** - If the contract is not deployed then we would consider the ownership to be not renounced. Moreover, if there are no ownership functionalities then the ownership is automatically considered renounced.



## Ownership Privileges

*These functions can be dangerous. Please note that abuse can lead to financial loss. We have a guide where you can learn more about these Functions.*

### Minting tokens

*Minting tokens refer to the process of creating new tokens in a cryptocurrency or blockchain network. This process is typically performed by the project's owner or designated authority, who has the ability to add new tokens to the network's total supply.*

#### Contract owner cannot mint new tokens

 **The owner cannot mint new tokens**

Description	The owner is not able to mint new tokens once the contract is deployed.
-------------	---


Comment	N/A
---------	-----





## Burning tokens

*Burning tokens is the process of permanently destroying a certain number of tokens, reducing the total supply of a cryptocurrency or token. This is usually done to increase the value of the remaining tokens, as the reduced supply can create scarcity and potentially drive up demand.*

Contract owner cannot burn tokens		 <b>The owner cannot burn tokens</b>
Description	The owner is not able burn tokens without any allowances.	
Comment	N/A	



## Blacklist addresses

*Blacklisting addresses in smart contracts is the process of adding a certain address to a blacklist, effectively preventing them from accessing or participating in certain functionalities or transactions within the contract. This can be useful in preventing fraudulent or malicious activities, such as hacking attempts or money laundering.*

**Contract owner cannot blacklist addresses**



**The owner cannot blacklist addresses**

Description

The owner is not able blacklist addresses to lock funds.

Comment

N/A





## Fees and Tax

*In some smart contracts, the owner or creator of the contract can set fees for certain actions or operations within the contract. These fees can be used to cover the cost of running the contract, such as paying for gas fees or compensating the contract's owner for their time and effort in developing and maintaining the contract.*

**Contract owner cannot set fees more than 25%**



**The owner cannot levy unfair taxes**

Description	The owner is not able to set the fees above 25%
Comment	N/A



## Lock User Funds

*In a smart contract, locking refers to the process of restricting access to certain tokens or assets for a specified period of time. When tokens or assets are locked in a smart contract, they cannot be transferred or used until the lock-up period has expired or certain conditions have been met.*

### Owner cannot lock the contract



**The owner cannot lock the contract**

Description

The owner is not able to lock the contract by any functions or updating any variables.

Comment

N/A

## Centralization Privileges

*Centralization can arise when one or more parties have privileged access or control over the contract's functionality, data, or decision-making. This can occur, for example, if the contract is controlled by a single entity or if certain participants have special permissions or abilities that others do not.*

In the project, there are authorities that have access to the following functions:

File	Privileges
1. GmxAdapter.sol	<ul style="list-style-type: none"> <li>❖ <b>onlyTraderorFactory</b> <ul style="list-style-type: none"> <li>- Open and Close Positions</li> <li>- Update and cancel orders</li> </ul> </li> </ul>
2. MuxAdapter.sol	<ul style="list-style-type: none"> <li>❖ <b>onlyTraderorFactory</b> <ul style="list-style-type: none"> <li>- Place an opening position request on MUX</li> <li>- Cancel Orders</li> </ul> </li> </ul>
3. GmxProxyFactory.sol	<ul style="list-style-type: none"> <li>❖ <b>onlyOwner</b> <ul style="list-style-type: none"> <li>- Upgrade the implementation</li> <li>- Set Exchange Config</li> <li>- Set maintainer address</li> <li>- Remove Proxy</li> <li>- Withdraw</li> </ul> </li> </ul>
4. MuxProxyFactory.sol	<ul style="list-style-type: none"> <li>❖ <b>onlyOwner</b> <ul style="list-style-type: none"> <li>- Upgrade the implementation</li> <li>- Set Exchange Config</li> <li>- Set maintainer address</li> <li>- Remove Proxy</li> <li>- Withdraw</li> </ul> </li> </ul>

## Recommendations

To avoid potential hacking risks, it is advisable for the client to manage the private key of the privileged account with care. Additionally, we recommend enhancing the security practices of centralized privileges or roles in the protocol through a decentralized mechanism or smart-contract-based accounts, such as multi-signature wallets.

Here are some suggestions of what the client can do:

- Consider using multi-signature wallets: Multi-signature wallets require multiple parties to sign off on a transaction before it can be executed, providing an extra layer of security e.g. Gnosis Safe
- Use of a timelock at least with a latency of e.g. 48-72 hours for awareness of privileged operations



- Introduce a DAO/Governance/Voting module to increase transparency and user involvement
- Consider Renouncing the ownership so that the owner cannot modify any state variables of the contract anymore. Make sure to set up everything before renouncing.



# Audit Results

## #1 | Wrong Access Control

File	Severity	Location	Status
GmxProxyFactory	Medium	L240	Open

**Description** - The function can be called by anyone which is very dangerous as all users will be able to cancel timeout orders. Moreover, the comment above the code implies that the method must be callable by the “maintainer” but the function has no checks to verify it.

**Remediation** - Make sure to set the correct access control for the function, either using a modifier or a require check.

## #2 | Missing Events

File	Severity	Location	Status
GmxAdapter	Low	L93, 176, 317, 350	Open

**Description** - Make sure to emit events for all the critical parameter changes in the contract to ensure the transparency and trackability of all the state variable changes in the contract.

## #3 | Missing Events

File	Severity	Location	Status
MuxAdapter	Low	L87, 147	Open

**Description** - Make sure to emit events for all the critical parameter changes in the contract to ensure the transparency and trackability of all the state variable changes in the contract.

## #4 | Missing Events

File	Severity	Location	Status
MuxProxyFactory	Low	L190,	Open

**Description** - Make sure to emit events for all the critical parameter changes in the contract to ensure the transparency and trackability of all the state variable changes in the contract.

## #5 | Missing Events

File	Severity	Location	Status
GmxProxyFactory	Low	L120,	Open

**Description** - Make sure to emit events for all the critical parameter changes in the contract to ensure the transparency and trackability of all the state variable changes in the contract.

## #6 | Missng “isContract” check

File	Severity	Location	Status
GmxAdapter	Low	L280	Open

**Description** - The contract doesn't have any checks to verify whether the “\_account.account” is an EOA or not, so without the receive function this call will be reverted

**Remediation** - We recommend putting a check to verify that the caller of the function must be an EOA

## #7 | Missng “zero Value” check

File	Severity	Location	Status
GmxAdapter	Low	L351	Open

**Description** - We recommend putting a check if the length of the keys is zero or not.

## #8 | Missng “receive” function

File	Severity	Location	Status
MuxAdapter	Low	L158	Open

**Description** - The payable keyword will not work without the “receive” function.

**Remediation** - We recommend putting receive function in the contract to address the issue.



## #9 | NatSpec documentation missing

File	Severity	Location	Status
All	Informational	—	Open

**Description** - If you started to comment on your code, also comment on all other functions, variables etc.

## #10 | Contract doesn't import npm packages from source (like OpenZeppelin etc.)

File	Severity	Location	Status
All	Informational	N/A	Open

**Description** - We recommend importing all packages from npm directly without flattening the contract. Functions could be modified or can be susceptible to vulnerabilities.

**Recommendation for the Project Developers** - As we have observed that the contracts are upgradeable in nature we strongly advise the team to make sure to disable the initializers in all the contracts once they are initialized otherwise there will be a risk of losing the ownership of the contracts.

Moreover, it has also come to our attention that some router contracts still have "TODO" comments and we would strongly advise the team to address them properly. If not done so then it may lead to unwanted errors and those errors would not be considered as part of the audit.



## Legend for the Issue Status

Attribute or Symbol	Meaning
<b>Open</b>	The issue is not fixed by the project team.
<b>Fixed</b>	The issue is fixed by the project team.
<b>Acknowledged(ACK)</b>	The issue has been acknowledged or declared as part of business logic.





**Blockchain Security | Smart Contract Audits | KYC  
Development | Marketing**

MADE IN GERMANY