



# Audit

Security Assessment

---

Date: **31. July 2024**

Client: **Axiome**

# Index

Introduction.....	3
Disclaimer.....	3
Project Overview.....	4
Overview.....	4
Social Media Information.....	4
Scope of Work.....	5
Imported Packages.....	6
Audit Information.....	7
Vulnerability & Risk Management Level.....	7
Auditing Strategy and Techniques Applied.....	7
Methodology.....	8
Audit Results .....	10
Critical Issues - [ 0 ].....	10
High Issues - [ 4 ].....	10
Medium Issues - [ 4 ].....	11
Low Issues - [ 1 ].....	12
Informational - [ 3 ].....	13

## Introduction

line1.io is a distinguished brand under the officially registered entity, FutureVisions Deutschland, based in Germany. Our expertise lies in Blockchain Security, providing comprehensive services including Smart Contract Audits and KYC verification for project teams. At line1.io, we rigorously evaluate smart contracts for security vulnerabilities, verify the alignment between the codebase and related whitepapers/documentation, and deliver detailed recommendations for improvement.

## Disclaimer


line1.io reports are neither an endorsement nor a disapproval of any specific project or team. These reports do not reflect the economic viability or value of any product or asset created by any team. line1.io does not test or audit the integration with external contracts or services.

**line1.io audits do not offer any warranty or guarantee regarding the absolute absence of bugs in the analyzed technology, nor do they provide any information about the proprietors of the technology. These audits should not be used as a basis for making investment decisions or for involvement in any particular project. The reports do not constitute investment advice and should not be relied upon as such.**

line1.io reports represent a comprehensive auditing process designed to assist our clients in enhancing the quality of their code while mitigating the significant risks associated with cryptographic tokens and blockchain technology. Given the high level of inherent risk in blockchain technology and cryptographic assets, line1.io emphasizes that each company and individual must conduct their own due diligence and maintain continuous security. line1.io does not claim any guarantee of the security or functionality of the technologies we analyze.

# Project Overview

## Overview

Project Name	Axiome
Website	<a href="https://axiome.pro">https://axiome.pro</a>
About the project	<p>Axiome is a DeFi ecosystem built on its own L1 solution, Axiome Chain, powered by the unique AXM coin.</p> <p>Our mission is to build a scalable ecosystem of modern DeFi solutions, allowing AXM holders to continuously increase their staking rewards.</p>
Chain	Own Blockchain – Axiome. Build with Cosmos SDK.
Language	

## Social Media Information

Telegram	<a href="https://t.me/axiomeen">https://t.me/axiomeen</a>
X / Twitter	<a href="https://twitter.com/axiome_pro">https://twitter.com/axiome_pro</a>
GitHub	<a href="https://github.com/axiome-pro">https://github.com/axiome-pro</a>
YouTube	<a href="https://www.youtube.com/@axiome_ru">https://www.youtube.com/@axiome_ru</a>

## Audit Summary

Version	Date	Changelog
V1.0	10th May 2024	Initial Report
V1.1	31th July 2024	Adjustments and Publication

**Note:** This audit report provides a detailed security analysis of the Go codebase used in the project, particularly focusing on potential vulnerabilities to external malicious interference with the program's functions. This analysis did not cover functional testing (or unit testing) of the program's logic. Therefore, we cannot assure complete logical correctness of the code, as we did not perform functional tests on it. This includes the internal calculations in the algorithms implemented in the codebase.

## Scope of Work

We aim to conduct a comprehensive security assessment of the provided repository, including a fork of the Cosmos SDK and custom modifications made by the Axiome team. This assessment aims to identify and mitigate potential security vulnerabilities, ensure code integrity, and verify the robustness of modifications to support secure, reliable, and efficient operations.

The repository consists of:

- The original Cosmos SDK source code
- Custom modifications and enhancements made to the SDK
- Integration points and dependencies with external systems or libraries

Repository	Commit
<a href="https://github.com/axiome-pro/axm-node">https://github.com/axiome-pro/axm-node</a>	645184e

## Imported Packages

Used code from other Frameworks. More are imported indirectly.

- ◉ cosmossdk.io/api v0.7.2
- ◉ cosmossdk.io/client/v2 v2.0.0-beta.1
- ◉ cosmossdk.io/collections v0.4.0
- ◉ cosmossdk.io/core v0.11.0
- ◉ cosmossdk.io/depinject v1.0.0-alpha.4
- ◉ cosmossdk.io/errors v1.0.1
- ◉ cosmossdk.io/log v1.3.0
- ◉ cosmossdk.io/math v1.2.0
- ◉ cosmossdk.io/store v1.0.2
- ◉ cosmossdk.io/tools/confix v0.1.1
- ◉ cosmossdk.io/x/upgrade v0.1.1
- ◉ github.com/bits-and-blooms/bitset v1.8.0
- ◉ github.com/cockroachdb/errors v1.11.1
- ◉ github.com/cometbft/cometbft v0.38.5
- ◉ github.com/cosmos/cosmos-db v1.0.0
- ◉ github.com/cosmos/cosmos-proto v1.0.0-beta.3
- ◉ github.com/cosmos/cosmos-sdk v0.50.3
- ◉ github.com/cosmos/go-bip39 v1.0.0
- ◉ github.com/cosmos/gogoproto v1.4.11
- ◉ github.com/gogo/protobuf v1.3.2
- ◉ github.com/golang/mock v1.6.0
- ◉ github.com/golang/protobuf v1.5.3
- ◉ github.com/gorilla/mux v1.8.0
- ◉ github.com/grpc-ecosystem/grpc-gateway v1.16.0
- ◉ github.com/hashicorp/go-metrics v0.5.2
- ◉ github.com/pkg/errors v0.9.1
- ◉ github.com/spf13/cobra v1.8.0
- ◉ github.com/spf13/pflag v1.0.5
- ◉ github.com/spf13/viper v1.18.2
- ◉ github.com/stretchr/testify v1.8.4
- ◉ golang.org/x/exp v0.0.0-20231006140011-7918f672742d
- ◉ google.golang.org/genproto/googleapis/api v0.0.0-20231120223509-83a465c0220f
- ◉ google.golang.org/grpc v1.60.1
- ◉ google.golang.org/protobuf v1.32.0
- ◉ gopkg.in/yaml.v3 v3.0.1
- ◉ gotest.tools/v3 v3.5.1

**Note for Investors:** We have only audited the Go dependencies listed in the above scope. We have not reviewed any additional dependencies related to the project that are not included in our audit scope, and we cannot comment on their security. We are not responsible for any security issues arising from these unreviewed dependencies.

# Audit Information

## Vulnerability & Risk Management 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	Description	Required Action
CRITICAL	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.
HIGH	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.
MEDIUM	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.
LOW	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.
INFORMATIONAL	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 audit of the Cosmos SDK Layer 1 blockchain repository, meticulous attention was dedicated to identifying security vulnerabilities, ensuring code quality, and verifying adherence to both specifications and best practices. Our audit was conducted by a seasoned team of penetration testers and blockchain developers with extensive experience in the Cosmos ecosystem and smart contract security.

Our team conducted a thorough line-by-line review of the code, ensuring that no detail was overlooked. Every identified issue was meticulously documented to provide a comprehensive overview of potential vulnerabilities and areas for improvement. Each file within the repository was subjected to a thorough manual examination, maintaining a high level of scrutiny and precision.

While automated tools were employed, their usage was strategically limited to augment the efficiency and effectiveness of the manual review process rather than replace it. This combination of rigorous manual inspection and strategic use of automated tools allows us to deliver an in-depth and reliable assessment, ensuring the highest standards of security and code quality for our clients.



## Methodology

The auditing process follows a structured series of steps to ensure thorough examination and assessment:

- **Specification Review:** Our team meticulously reviewed all relevant documentation, specifications, and guidelines provided initially. This step ensures a deep understanding of the blockchain's architecture, scope, and intended functionalities.
- **Manual Code Inspection:** The source code was examined line by line in an intensive review aimed at uncovering potential security vulnerabilities that could be exploited maliciously. This careful inspection ensures no detail is overlooked.
- **Specification Conformance:** The code was rigorously compared against the provided specifications to confirm its fidelity in performing as described. This ensures that the implementation accurately reflects the intended design and requirements.
- **Test Coverage Analysis:** We analyzed the extent of test cases' coverage over the codebase. This involved determining how much code was executed during these tests to identify untested paths and ensure comprehensive test coverage.
- **Symbolic Execution:** This technique was applied to analyze how different inputs affect the code execution paths. It helped understand the conditions under which various parts of the program would execute, revealing potential vulnerabilities and ensuring robust security.

During this audit, certain aspects, such as unused constants, functions, exported functions, global variables, and parameters, were excluded from the scope of our review. While these elements were not directly evaluated, we suggest revisiting these areas in future audits to ensure that they do not introduce security concerns or affect the overall quality of the codebase.



# Metrics

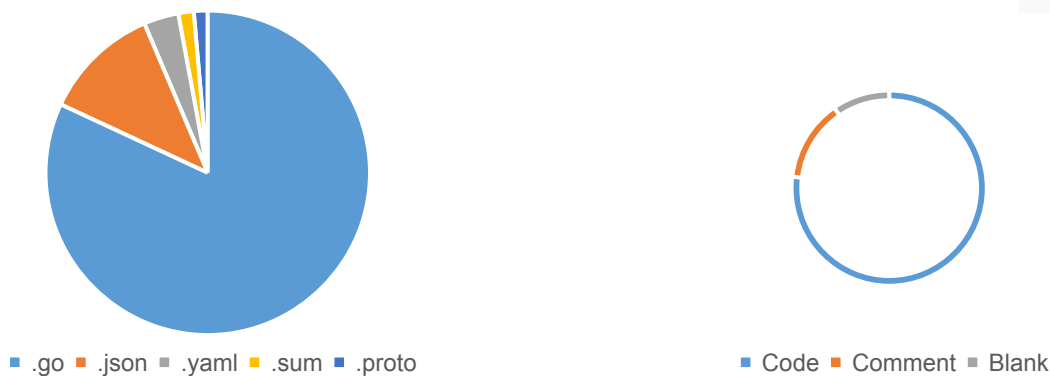
## Codebase Composition Overview

This report section provides an in-depth analysis of the project's codebase, which is primarily composed of Go (81%). The codebase is well-structured, with extensive comments and documentation, particularly within the .proto files. This level of documentation is crucial for maintaining clarity and understanding across the development team and for any future developers who may work on the project.

The code distribution reveals that 77% of the total lines are dedicated to the actual code, while 14% consist of comments, and 10% are blank lines. This distribution highlights a balanced approach to coding, emphasizing not only functionality but also the importance of clear documentation. The substantial proportion of comments demonstrates a commitment to making the code understandable and maintainable. These comments provide insights into the code's logic, functionality, and any potential edge cases that developers need to be aware of.

Additionally, the presence of 10% blank lines indicates that the codebase follows good coding practices, such as separating logical blocks of code for better readability and maintainability. This practice helps in reducing complexity and making the code more approachable for review and debugging.

Overall, the structure of the codebase, with its emphasis on thorough documentation and clear separation of code, comments, and blank lines, suggests a high level of professionalism and foresight in its development. This meticulous approach not only facilitates easier maintenance and updates but also ensures that the project can be effectively scaled and adapted as needed in the future.



Extension	Total Code	Total Comment	Total Blank	Percent
.go	138.085	24.417	17.097	81
.json	19.675	0	0	11
.yaml	5.853	19	1126	3.4
.sum	2.591	25	0	1.5
.proto	2.333	745	570	1.4

# Audit Results

## Critical Issues - [ 0 ]

No Critical Issues

## High Issues - [ 4 ]

#H-1 Constant Condition		
Severity	Location / Line	Status
High	x/staking/client/cli/tx.go:516 x/staking/keeper/delegation.go:1096 x/staking/keeper/query_utils.go:44	Open
Description	Condition is always false	
Advisory	Check the logic and adjust your code.	

#H-2 Use of deprecated Methods		
Severity	Location / Line	Status
High	app/params/config.go:52 x/staking/keeper/keeper.go:136 x/staking/keeper/keeper.go:148	Open
Description	These methods are supported for backward compatibility only and will be removed in a future release.	
Advisory	Docs: <a href="https://pkg.go.dev/github.com/cosmos/cosmos-sdk/types@v0.50.3#Config.SetFullFundraiserPath">https://pkg.go.dev/github.com/cosmos/cosmos-sdk/types@v0.50.3#Config.SetFullFundraiserPath</a> Docs: <a href="https://pkg.go.dev/github.com/cosmos/cosmos-sdk/types@v0.50.3#IntProto">https://pkg.go.dev/github.com/cosmos/cosmos-sdk/types@v0.50.3#IntProto</a>	

#H-3 Error may be not nil		
Severity	Location / Line	Status
High	x/staking/keeper/points.go:46 x/staking/types/ref_hooks.go:44	Open
Description	Operations on emissionRate and timeElapsed without error checks and unchecked error from PayUpFees function.	
Advisory	Ensure the robustness of MullInt64, TruncateInt, and Uint64 or introduce error handling if these functions can fail. Explicitly check and handle any errors returned by PayUpFees.	

#H-4 Improper Input Validation		
Severity	Location / Line	Status
High	Dependency: cosmos-sdk	Open
Description	The "ValidateVoteExtensions" helper function in Cosmos SDK may allow incorrect voting power assumptions. This issue affects github.com/cosmos/cosmos-sdk versions 0.50.0-alpha.0 through 0.50.4.	
Advisory	Upgrade to 0.50.6. More information under: <a href="https://github.com/advisories/GHSA-95rx-m9m5-m94v">https://github.com/advisories/GHSA-95rx-m9m5-m94v</a> Cosmos SDK 0.50.5 Changelog: <a href="https://github.com/cosmos/cosmos-sdk/releases/tag/v0.50.5">https://github.com/cosmos/cosmos-sdk/releases/tag/v0.50.5</a>	

## Medium Issues - [ 4 ]

#M-1 Loop with Unreachable Exit Condition		
Severity	Location / Line	Status
Medium	Dependency: google.golang.org/protobuf	Open
Description	In the package google.golang.org/protobuf versions prior to 1.33.0, the "protojson.Unmarshal" function can enter an infinite loop when unmarshaling certain forms of invalid JSON. This condition can occur when unmarshaling into a message which contains a "google.protobuf.Any" value, or when the "UnmarshalOptions.DiscardUnknown" option is set.	
Advisory	Upgrade to 1.33.0 <a href="https://pkg.go.dev/vuln/GO-2024-2611">https://pkg.go.dev/vuln/GO-2024-2611</a> <a href="https://github.com/advisories/GHSA-8r3f-844c-mc37">https://github.com/advisories/GHSA-8r3f-844c-mc37</a>	

#M-2 Improper Restriction of Excessive Authentication Attempts		
Severity	Location / Line	Status
Medium	Dependency: JumpServer	Open
Description	JumpServer is an open-source bastion host. When users enable MFA and use a public key for authentication, the Koko SSH server does not verify the corresponding SSH private key. An attacker could exploit a vulnerability by utilizing a disclosed public key to attempt brute-force authentication against the SSH service. This issue has been patched in versions 3.6.5 and 3.5.6. There are no known workarounds for this issue.	
Advisory	Upgrade to 0.21.0 <a href="https://github.com/jumpserver/jumpserver/security/advisories/GHSA-jv3c-27cv-w8jv">https://github.com/jumpserver/jumpserver/security/advisories/GHSA-jv3c-27cv-w8jv</a>	

#M-3 Potential nil dereference		
Severity	Location / Line	Status
Medium	x/distribution/keeper/delegation.go:120	Open
Description	<p>The rewards receiver might be nil in the call to Add, causing a runtime panic.</p> <p>The Add method for DecCoins operates under the invariant that coins are sorted by denominations and will never return coins where one has a non-positive amount. It ensures that IsValid will always return true for the result</p> <p>Docs: <a href="https://pkg.go.dev/github.com/cosmos/cosmos-sdk/types@v0.50.3#DecCoins.Add">https://pkg.go.dev/github.com/cosmos/cosmos-sdk/types@v0.50.3#DecCoins.Add</a></p>	
Advisory	Check if rewards is nil and initialize it if necessary before calling Add.	

#M-4 Insecure Default Initialization of Resource		
Severity	Location / Line	Status
Medium	Dependency: cometbft	Open
Description	<p>A default configuration in CometBFT is small for common use cases and may prevent the slashing mechanism from working in specific cases. The default values for EvidenceParams.MaxAgeNumBlocks and EvidenceParams.MaxAgeDuration consensus parameters may not be sufficient for common use cases to provide coverage for the entire unbonding period for a chain (Staking.UnbondingTime). Suppose the conditions of both parameters are exceeded. In that case, evidence may be prematurely expired and considered no longer valid, potentially allowing for unpunished Byzantine behavior if the evidence is discovered outside that window.</p>	
Advisory	Upgrade to 1.0.0-alpha.2 after throughtout testing.	

## Low Issues - [ 1 ]

#L-1 Types not defined		
Severity	Location / Line	Status
Low	X/vote/types/types.go:185 X/vote/keeper/keeper.go:519	Open
Description	Types *Poll_CanValidate and *Poll_MinStatus are not defined.	
Advisory	Check and adjust the code.	

## Informational - [ 3 ]

#I-1 New Codec Recommendation		
Severity	Location / Line	Status
Informational	/	Open
Description	We have identified instances of NewAminoCodec usage within the codebase. To adhere to modern best practices and maintain future compatibility, we recommend transitioning to NewLegacyAmino. Implementing this change will enhance the maintainability, performance, and security of the codebase.	
Advisory	/	

#I-2 Unhandled Errors		
Severity	Location / Line	Status
Informational	/	Open
Description	During the audit, multiple instances of unhandled errors were observed. Addressing these unhandled errors is essential for improving the system's reliability and robustness. Proper error handling ensures that the system can gracefully manage unexpected conditions without compromising operational continuity or security.	
Advisory	/	

#I-3 Update of vulnerable dependencies		
Severity	Location / Line	Status
Informational	/	Open
Description	Our analysis revealed 15 possible vulnerabilities across 15 dependencies. Of these, four were highlighted as high and medium and have been detailed in the issues table for urgent resolution. These dependencies could introduce security risks and should be promptly addressed to protect the system's security and stability.	
Advisory	/	