HACKEN

Blockchain Protocol Difference Evaluation Report

Customer: Makachain

Date: 20/11/2025

We express our gratitude to the Makachain team for the collaborative engagement that enabled the execution of this Blockchain Protocol Security Assessment.

This report evaluates the code changes between the following pairs of repositories:

- MeghaZeeve/avalanchego and the upstream avalanchego;
- MeghaZeeve/subnet-evm and the upstream subnet-evm.

It was confirmed that both repositories in scope are exact forks of the corresponding Avalanche codebases, without any modifications. Both avalanchego and subnet-evm are widely regarded as reliable and secure; however, it is important to note that this code was not assessed for correctness, reliability, or security within the scope of this review.

Document

Name	Blockchain Protocol Review and Security Analysis Report for Makachain
Audited By	Tanuj Soni, Nino Lipartiia
Approved By	Nino Lipartiia
Changelog	20/11/2025 - Final Report
Platform	Makachain
Language	Golang
Tags	Avalanche, subnet-evm

Review Scope

Repository_avalanche	https://github.com/MeghaZeeve/avalanchego
Commit	55652312c02869c5f4c202e73e90de9083420500
Repository_subnet-evm	https://github.com/MeghaZeeve/subnet-evm
Commit	36af3391857049d5e660d178316e4fd057cb50b7

Audit Summary

This report evaluates the `MeghaZeeve/avalanchego` and `MeghaZeeve/subnet-evm` forks against the official upstream `avalanchego` and `subnet-evm` repositories. At commits `55652312c` and `36af3391`, respectively, the forked repositories are byte-identical to their upstream counterparts, with no deviations detected.

A comparison was also performed against the versions stated as currently used by MakaChain:

- avalanchego v1.14.0 - https://github.com/ava-labs/avalanchego/tree/v1.14.0
- subnet-evm v0.8.0 - https://github.com/ava-labs/subnet-evm/tree/v0.8.0

It was observed that the most recent commits in the `MeghaZeeve` forks advance beyond these version tags. This indicates that the forks incorporate upstream changes newer than those associated with the referenced deployment versions, without introducing any custom modifications specific to MakaChain.

Table of Contents

System Overview

- **AvalancheGo** is the core node implementation for the Avalanche network, enabling users to run validator nodes, create custom blockchains (subnets), and participate in network consensus. It tracks validator stakes, manages network upgrades, and adheres to Avalanche's consensus protocol. Privileged accounts with validator permissions can participate in consensus and network governance.
- **Subnet-EVM** is an Ethereum Virtual Machine (EVM) implementation built as a fork of `go-ethereum/coreth`, designed for Avalanche subnets. It retains core EVM functionality while introducing subnet-specific modifications, custom precompiles, and network upgrade configurations to enhance compatibility and streamline operations on Avalanche subnets.

MakaChain uses these repositories to run a custom chain: AvalancheGo provides the base node implementation, while Subnet-EVM acts as the plugin VM. This setup supports custom stateful precompiles enabling features such as native token minting, contract allow-lists, custom reward distribution, and cross-chain message passing (e.g., Warp Messenger).

- This document presents a code comparison between MakaChain and the upstream Avalanche repositories. It is not a security assessment, and therefore should not be considered proof or a guarantee that either codebase is correct or secure.

Findings

Vulnerability Details

F-2025-14104 - MeghaZeeve/subnet-evm Is Ahead of subnet-evm v0.8.0 - Info

Description: When comparing `MeghaZeeve/subnet-evm` to the upstream `subnet-evm` v0.8.0 tag, it was observed that the MakaChain fork is ahead by one additional commit:

- Update avalanchego to v1.14.1-db-metrics-fix (ava-labs#1836)

This commit updates the AvalancheGo dependency to a newer patch version (`v1.14.1-db-metrics-fix`), incorporating an upstream fix related to database metrics handling.

Status: Fixed

Classification

Severity: Info

Recommendations

Remediation: N/A

F-2025-14105 - MeghaZeeve/avalanchego Is Ahead of avalanchego v1.14.0 - Info

Description:

When comparing `MeghaZeeve/avalanchego` to the upstream `avalanchego` v1.14.0 tag, it was observed that the MakaChain fork is ahead by 14 additional commits:

- Avoid unnecessary RST_STREAM and PING frames causing GOAWAY messages (ava-labs#4480)
- [tooling] Update to use go tool for mockgen and canoto (ava-labs#4479)
- Fix meterdb metrics for root db (ava-labs#4476)
- Reenable the upgrade tests (ava-labs#4477)
- Update canoto to v0.17.3 (ava-labs#4473)
- Remove gosec from test linting (ava-labs#4471)
- Add UpgradeTime to p2p handshake (ava-labs#4468)
- Cleanup the version package (ava-labs#4465)
- chore(customrawdb): unexport `customrawdb.NewSyncPerformedIterator` (ava-labs#4450)
- Fix ConvertSubnetToL1Tx flake (ava-labs#4462)
- Update darwin platform support from amd64 to arm64 (ava-labs#4461)
- Cache subnetIDs in cachedState (ava-labs#4402)
- fix(crypto/secp256k1): UnmarshalText (ava-labs#4419)
- Sort transactions by gas price in P-Chain mempool (ava-labs#3715)

These commits introduce various fixes and improvements, including:

- Networking stability improvements and reduced unnecessary RST/PING/GOAWAY frames
- Metrics and database fixes (meterdb, root DB handling)
- P-Chain mempool gas-price sorting
- Platform updates (Darwin arm64), tooling updates, and linting adjustments
- Bug fixes in crypto, rawdb, and transaction handling
- Enhancements to upgrade logic, cached state, and p2p handshake metadata

Overall, the fork incorporates a set of upstream maintenance, performance, and reliability updates beyond the v1.14.0 release.

These additional commits are inherited directly from the upstream `avalanchego` codebase and do not include any project-specific alterations or custom logic

Status: Fixed

Classification

Severity: Info

Recommendations

Remediation: N/A

Disclaimers

Hacken Disclaimer

The blockchain protocol given for audit has been analyzed based on best industry practices at the time of the writing of this report, with cybersecurity vulnerabilities and issues in the protocol source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The report contains no statements or warranties on the identification of all vulnerabilities and security of the code. The report covers the code submitted and reviewed, so it may not be relevant after any modifications. Do not consider this report as a final and sufficient assessment regarding the utility and safety of the code, bug-free status, or any other protocol statements.

While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of the blockchain protocol.

English is the original language of the report. The Consultant is not responsible for the correctness of the translated versions.

As part of Hacken’s ongoing quality assurance process, we may conduct re-audits of select projects. These re-audits are performed independently from the original audit and are intended solely for internal quality control and improvement. Updated reports resulting from such re-audits will be shared privately with the respective clients and may be published on the Hacken website only with their explicit consent.
The sole authoritative source for finalized and most up-to-date versions of all reports remains the Audits section at https://hacken.io/audits/.

Technical Disclaimer

Blockchain protocols are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the protocol can have vulnerabilities that can lead to hacks. Thus, the Consultant cannot guarantee the explicit security of the audited blockchain protocol.

Appendix 1. Severity Definitions

Severity	Description
Critical	Vulnerabilities that can lead to a complete breakdown of the blockchain network's security, privacy, integrity, or availability fall under this category. They can disrupt the consensus mechanism, enabling a malicious entity to take control of the majority of nodes or facilitate 51% attacks. In addition, issues that could lead to widespread crashing of nodes, leading to a complete breakdown or significant halt of the network, are also considered critical along with issues that can lead to a massive theft of assets. Immediate attention and mitigation are required.
High	High severity vulnerabilities are those that do not immediately risk the complete security or integrity of the network but can cause substantial harm. These are issues that could cause the crashing of several nodes, leading to temporary disruption of the network, or could manipulate the consensus mechanism to a certain extent, but not enough to execute a 51% attack. Partial breaches of privacy, unauthorized but limited access to sensitive information, and affecting the reliable execution of smart contracts also fall under this category.
Medium	Medium severity vulnerabilities could negatively affect the blockchain protocol but are usually not capable of causing catastrophic damage. These could include vulnerabilities that allow minor breaches of user privacy, can slow down transaction processing, or can lead to relatively small financial losses. It may be possible to exploit these vulnerabilities under specific circumstances, or they may require a high level of access to exploit effectively.
Low	Low severity vulnerabilities are minor flaws in the blockchain protocol that might not have a direct impact on security but could cause minor inefficiencies in transaction processing or slight delays in block propagation. They might include vulnerabilities that allow attackers to cause nuisance-level disruptions or are only exploitable under extremely rare and specific conditions. These vulnerabilities should be corrected but do not represent an immediate threat to the system.

Appendix 2. Scope

The scope of the project includes the following components from the provided repository:

Scope Details	
Repository_1	https://github.com/MeghaZeeve/avalanchego
Repository_2	https://github.com/MeghaZeeve/subnet-evm
Commit_1	55652312c02869c5f4c202e73e90de9083420500
Commit_2	36af3391857049d5e660d178316e4fd057cb50b7

Components in Scope

- https://github.com/MeghaZeeve/avalanchego
- https://github.com/MeghaZeeve/subnet-evm

The codebases in scope were compared to:

- avalanchego v1.14.0 - https://github.com/ava-labs/avalanchego/tree/v1.14.0
- subnet-evm v0.8.0 - https://github.com/ava-labs/subnet-evm/tree/v0.8.0

Appendix 3. Additional Valuables

Frameworks and Methodologies

This security assessment was conducted in alignment with recognised penetration testing standards, methodologies and guidelines, including the NIST SP 800-115 – Technical Guide to Information Security Testing and Assessment, and the Penetration Testing Execution Standard (PTES), These assets provide a structured foundation for planning, executing, and documenting technical evaluations such as vulnerability assessments, exploitation activities, and security code reviews. Hacken's internal penetration testing methodology extends these principles to Web2 and Web3 environments to ensure consistency, repeatability, and verifiable outcomes.