FYEO

Flare

Security Review Update: 2025-02-11

Reviewer: thomas@gofyeo.com

Diff audit: Upgrade of Flare codebase to Avalanche v1.10.0

The current Flare codebase is based on Avalanche v1.9.0. In the process of catching up with the latest Avalanche code (v1.12.x), the next intermediate version will be based on v1.10.0, https://github.com/ava-labs/avalanchego/releases/tag/v1.10.0.

This audit report is a diff audit of the changes made to make this integration possible.

New security issues, 0

After the development team implemented the latest updates and fixes to integrate a new version of Avalanche into Flare, FYEO conducted a review of the modifications. The primary goal of this evaluation was to ensure the continued robustness of the app's security features, safeguarding user data and maintaining the overall integrity of the applications.

Analysis Methodology

To analyze the changes introduced in the updated codebase, FYEO followed a structured methodology that involved creating diffs between the supplied repository and the original repository. This approach allowed us to systematically review modifications and assess their impact on the overall code structure and functionality.

Process

1. **Generating Diffs** We began by comparing the supplied repository against the original codebase to identify all modifications. This was done using standard version control tools, generating `.diff` files that captured the changes at the line and function level for each affected file. These diffs provided insight into the additions, deletions, and modifications made to the code.
2. **Reviewing Diff Files** Once the diffs were created, FYEO systematically examined each `.diff` file to understand the scope and nature of the changes. The analysis focused on:
    - **Structural modifications** such as new functions, removed code, or refactored logic.
    - **Functionality enhancements** including new features, updated validation logic, and configuration changes.
    - **Code dependencies** to evaluate how modifications affected interactions within the larger codebase.

   - **Potential impact on the overall system** by determining whether the changes introduced significant alterations to core components.
3. **Assessing Full Repository Impact** After reviewing individual diffs, we examined their collective impact on the full repository. This involved:
   - Mapping changes to core modules and dependencies.
   - Evaluating how modifications influenced network interactions, transaction processing, and configuration management.
   - Identifying potential areas requiring further review or testing to ensure stability and compatibility.

Update Analysis

This update introduces a range of modifications across various files, enhancing network configuration, delegation logic, reward mechanisms, and overall compatibility with different blockchain networks such as Flare, Songbird, and Coston. The changes aim to improve flexibility, maintainability, and efficiency in handling network operations.

Key Changes

- A significant addition to the codebase is the `NetworkValue` struct in `utils/network_value.go`, which provides a structured way to manage network-related values. This enhancement improves data organization and accessibility, facilitating more effective network operations.
- Support for additional networks has been expanded, with new configurations introduced for Flare, Songbird, and test networks such as Coston and Costwo. These updates ensure that the system remains adaptable to evolving network requirements and can accommodate various blockchain environments seamlessly.
- The staking and delegation mechanisms have been refined, particularly in `staker_tx_verification.go`, where the delegation logic has been adjusted to dynamically manage stake limits based on time. This modification allows for more precise handling of staking rules and enhances the flexibility of the delegation process.
- Network upgrade schedules have also been modified in `version/constants.go`, incorporating changes that improve compatibility across different versions. This ensures smoother transitions between network phases and enhances overall system stability.
- Another important addition is the introduction of application prefix management in `version/flare_version.go`. New functions have been implemented to handle application prefixes based on the network ID, improving consistency in network identification and application behavior.
- The reward calculation logic has undergone a notable change in `reward/calculator.go`, where the function now always returns zero rewards. This adjustment alters the reward mechanism, potentially affecting staking incentives and overall economic dynamics within the network.

- in `secp256k1fx/fx.go` Ethereum-style signature verification has been introduced. This enhancement improves interoperability with Ethereum-based tools and platforms, making the system more versatile in handling cryptographic verification processes.
- Changes have also been made to various genesis files. New genesis configurations for Flare, Songbird, and Coston have been added in `genesis_flare.go`, `genesis_songbird.go`, and `genesis_coston.go`, respectively.
- The removal of hardcoded IP addresses and node IDs from `genesis/beacons.go` contributes to improved maintainability and reduces dependency on static configurations.
- Validator and network state management have also seen improvements with the introduction of new structures for tracking validator uptime in `validator_uptimes.go`. The refactoring of network inflation settings in `inflation_settings.go` enhances the modularity and maintainability of economic parameters within the network.
- Finally, configuration management has been enhanced in `genesis/config.go`, where improvements to network parameter handling and JSON parsing contribute to a more reliable system.

Conclusion and security posture

These updates enhance network support, refine staking and delegation mechanisms, introduce new network configurations, and improve application-specific logic. The modifications help ensure a more adaptable, efficient, and maintainable blockchain environment.

Commit Hashes Reference

For transparency and reference, the security review was conducted on the specific commit hashes for both the repositories. The commit hashes for the reviewed versions are as follows:

https://github.com/ava-labs/avalanchego/compare/535456298046b5c2fbcb95ce36702422b6980c66...mboben:avalanchego:flare-merge-1_10_0

https://github.com/ava-labs/coreth/compare/dce18eb9ca551d4a650c9f203c58679e1ddad3e3...mboben:coreth:flare-merge-0_12_0_rc2

Conclusion

The changes implemented in the updated codebase are structured in a secure manner and contribute to improving the overall security of the platform. The modifications enhance validation logic, network configurations, and transaction processing, strengthening the integrity of the system.

The only notable exception is the removal of the staking incentive, which may impact participation in the staking mechanism. However, this change aligns with the Flare model, where staking incentives are intentionally removed as part of the network's design. Therefore, in this specific context, the absence of staking incentives does not pose a security concern.