MixBytes()

Aave v3.6 Security Audit Report

DECEMBER 09, 2025

Table of Contents

1. Introduction

1.1 Disclaimer

The audit makes no statements or warranties regarding the utility, safety, or security of the code, the suitability of the business model, investment advice, endorsement of the platform or its products, the regulatory regime for the business model, or any other claims about the fitness of the contracts for a particular purpose or their bug-free status.

1.2 Executive Summary

Aave is a decentralized protocol for liquidity provision, enabling users to earn interest on supplied assets or borrow by locking collateral.

Aave v3.6 decouples default reserve configuration from eMode settings: `lt` and `ltv` can now be configured independently per eMode (not constrained by the default reserve configuration), and borrowing ability is controlled per eMode via `borrowableBitmap` (independent of the reserve-level `borrowingEnabled` flag). It introduces an `ltvZeroBitmap` that applies `LTV = 0` rules per eMode, allowing assets to be collateral-only or borrowable-only inside specific modes.

Automatic collateral activation has been removed for aToken transfers, liquidations of aTokens, and isolated-collateral deposits, reducing gas costs for these operations. Supply and deposit operations still automatically enable collateral.

Event emission behavior now aligns with OpenZeppelin's ERC20 standard implementation, reducing gas costs and improving ecosystem consistency. Additionally, infinite allowance `(uint256.max)` is no longer consumed during transferFrom, matching OpenZeppelin's behavior. The release also adds `renounceAllowance()` on AToken and `renounceDelegation()` on VariableDebtToken, allowing delegatees to revoke excess approvals they have received.

The audit was conducted by 3 auditors over 4 days, involving a thorough manual code review of the scope and analysis via automated tools.

During the audit, the following attack vectors were checked:

- We verified that removing an asset from `eModeConfiguration.collateralEnabledBitmap` will also remove it from `eModeConfiguration.ltvzeroBitmap`.
- We confirmed that the update preserves all core protocol functionality and that existing user operations continue to work as expected.
- We verified that `ValidationLogic.getUserReserveLtv()` correctly returns the LTV value for all cases – eMode category LTV, base reserve LTV, or 0 when ltvzero is set in eMode – depending on the user's eMode status and asset configuration.

- We ensured that when switching eModes, the protocol enforces that all currently borrowed assets can be borrowed in the target eMode and that all enabled collaterals have non-zero LTV in the target eMode (not considered ltv0).
- We verified that the new zero-LTV assignment logic correctly applies and clears zero LTV for assets within eMode categories, reverting to base LTV when disabled and keeping all related bitmaps and validations consistent.

The codebase is well-written, follows development best practices, and aligns with the documentation describing new features and protocol changes. The project demonstrates high test coverage, including unit tests, integration tests, and invariant tests, ensuring system reliability and correctness.

1.3 Project Overview

Summary

Title	Description
Client Name	Aave DAO
Project Name	Aave v3.6
Code developed by	BGD Labs
Type	Solidity
Platform	EVM
Timeline	28.10.2025 - 09.12.2025

Scope of Audit

File	Link
src/contracts/protocol/libraries/logic/GenericLogic.sol	GenericLogic.sol
src/contracts/protocol/libraries/logic/LiquidationLogic.sol	LiquidationLogic.sol
src/contracts/protocol/libraries/logic/SupplyLogic.sol	SupplyLogic.sol

File	Link
src/contracts/protocol/libraries/logic/ValidationLogic.sol	ValidationLogic.sol
src/contracts/protocol/libraries/types/DataTypes.sol	DataTypes.sol
src/contracts/protocol/pool/Pool.sol	Pool.sol
src/contracts/protocol/pool/PoolConfigurator.sol	PoolConfigurator.sol
src/contracts/protocol/tokenization/AToken.sol	AToken.sol
src/contracts/protocol/tokenization/VariableDebtToken.sol	VariableDebtToken.sol
src/contracts/protocol/tokenization/base/DebtTokenBase.sol	DebtTokenBase.sol
src/contracts/protocol/tokenization/base/IncentivizedERC20.sol	IncentivizedERC20.sol

Versions Log

Date	Commit Hash	Note
28.10.2025	4143c2a59e0d901e4d05a8f4fb1791d998ce3b45	Initial Commit
07.11.2025	ce5d9ef3b8ad30eb7dbbd3e16a998a1b4de8e954	Re-audit Commit
09.12.2025	4df386b1c57381564a603fc78e0ad5663b567dae	Re-audit Commit

Mainnet Deployments

The deployment verification will be conducted later after the full deployment of the protocol into the mainnet.

1.4 Security Assessment Methodology

Project Flow

Stage	Scope of Work
Interim audit	**Project Architecture Review:** · Review project documentation · Conduct a general code review · Perform reverse engineering to analyze the project's architecture based solely on the source code · Develop an independent perspective on the project's architecture · Identify any logical flaws in the design OBJECTIVE: UNDERSTAND THE OVERALL STRUCTURE OF THE PROJECT AND IDENTIFY POTENTIAL SECURITY RISKS.
	Code Review with a Hacker Mindset: · Each team member independently conducts a manual code review, focusing on identifying unique vulnerabilities. · Perform collaborative audits (pair auditing) of the most complex code sections, supervised by the Team Lead. · Develop Proof-of-Concepts (PoCs) and conduct fuzzing tests using tools like Foundry, Hardhat, and BOA to uncover intricate logical flaws. · Review test cases and in-code comments to identify potential weaknesses. OBJECTIVE: IDENTIFY AND ELIMINATE THE MAJORITY OF VULNERABILITIES, INCLUDING THOSE UNIQUE TO THE INDUSTRY.
	Code Review with a Nerd Mindset: · Conduct a manual code review using an internally maintained checklist, regularly updated with insights from past hacks, research, and client audits. · Utilize static analysis tools (e.g., Slither, Mythril) and vulnerability databases (e.g., Solodit) to uncover potential undetected attack vectors. OBJECTIVE: ENSURE COMPREHENSIVE COVERAGE OF ALL KNOWN ATTACK VECTORS DURING THE REVIEW PROCESS.

Stage	Scope of Work
	Consolidation of Auditors' Reports: · Cross-check findings among auditors · Discuss identified issues · Issue an interim audit report for client review OBJECTIVE: COMBINE INTERIM REPORTS FROM ALL AUDITORS INTO A SINGLE COMPREHENSIVE DOCUMENT.
Re-audit	**Bug Fixing & Re-Audit:** · The client addresses the identified issues and provides feedback · Auditors verify the fixes and update their statuses with supporting evidence · A re-audit report is generated and shared with the client OBJECTIVE: VALIDATE THE FIXES AND REASSESS THE CODE TO ENSURE ALL VULNERABILITIES ARE RESOLVED AND NO NEW VULNERABILITIES ARE ADDED.
Final audit	**Final Code Verification & Public Audit Report:** · Verify the final code version against recommendations and their statuses · Check deployed contracts for correct initialization parameters · Confirm that the deployed code matches the audited version · Issue a public audit report, published on our official GitHub repository · Announce the successful audit on our official X account OBJECTIVE: PERFORM A FINAL REVIEW AND ISSUE A PUBLIC REPORT DOCUMENTING THE AUDIT.

1.5 Risk Classification

Severity Level Matrix

Severity	Impact: High	Impact: Medium	Impact: Low
Likehood: High	Critical	High	Medium
Likehood: Medium	High	Medium	Low
Likehood: Low	Medium	Low	Low

Impact

- **High** – Theft from 0.5% OR partial/full blocking of funds (>0.5%) on the contract without the possibility of withdrawal OR loss of user funds (>1%) who interacted with the protocol.
- **Medium** – Contract lock that can only be fixed through a contract upgrade OR one-time theft of rewards or an amount up to 0.5% of the protocol's TVL OR funds lock with the possibility of withdrawal by an admin.
- **Low** – One-time contract lock that can be fixed by the administrator without a contract upgrade.

Likelihood

- **High** – The event has a 50–60% probability of occurring within a year and can be triggered by any actor (e.g., due to a likely market condition that the actor cannot influence).
- **Medium** – An unlikely event (10–20% probability of occurring) that can be triggered by a trusted actor.
- **Low** – A highly unlikely event that can only be triggered by the owner.

Action Required

- **Critical** – Must be fixed as soon as possible.
- **High** – Strongly advised to be fixed to minimize potential risks.
- **Medium** – Recommended to be fixed to enhance security and stability.
- **Low** – Recommended to be fixed to improve overall robustness and effectiveness.

Finding Status

- **Fixed** – The recommended fixes have been implemented in the project code and no longer impact its security.
- **Partially Fixed** – The recommended fixes have been partially implemented, reducing the impact of the finding, but it has not been fully resolved.
- **Acknowledged** – The recommended fixes have not yet been implemented, and the finding remains unresolved or does not require code changes.

1.6 Summary of Findings

Findings Count

Severity	Count
Critical	0
High	0
Medium	0
Low	2

Findings Statuses

ID	Finding	Severity	Status
L-1	Unused `toEModeCategory` Parameter in `Pool.finalizeTransfer()`	Low	Fixed
L-2	Documentation Inconsistencies	Low	Fixed

2. Findings Report

2.1 Critical

Not Found

2.2 High

Not Found

2.3 Medium

Not Found

2.4 Low

L-1	Unused `toEModeCategory` Parameter in `Pool.finalizeTransfer()`		
Severity	Low	**Status**	Fixed in `ce5d9ef3`

Description

In `Pool.finalizeTransfer()`, the parameter `toEModeCategory: _usersEModeCategory[to]` is passed to `SupplyLogic.executeFinalizeTransfer()` but is never used within the function or any of its internal calls.

Recommendation

We recommend removing the unused `toEModeCategory` parameter from `FinalizeTransferParams` struct and from the call in `Pool.finalizeTransfer()`.

Client's Commentary

Addressed in e4efedc0e3bd880a51521417e13046a2d8a0e4eb

L-2	Documentation Inconsistencies		
Severity	Low	**Status**	Fixed in ce5d9ef3

Description

Automatic collateral activation was removed from transfers and liquidations as documented in the v3.6 features. However, the `@notice` comment in `ValidationLogic.validateAutomaticUseAsCollateral()` still states that this function is used for supply, transfer, and liquidate, which is inaccurate. The function is currently only called in `SupplyLogic.executeSupply()`.

Recommendation

We recommend updating the `@notice` comment in `ValidationLogic.validateAutomaticUseAsCollateral()` to align with the documentation and accurately reflect the current usage of this function.

Client's Commentary

Addressed in 893c8caf851723b8391b5d645e137d69befc28a8

3. About MixBytes

MixBytes is a leading provider of smart contract audit and research services, helping blockchain projects enhance security and reliability. Since its inception, MixBytes has been committed to safeguarding the Web3 ecosystem by delivering rigorous security assessments and cutting-edge research tailored to DeFi projects.

Our team comprises highly skilled engineers, security experts, and blockchain researchers with deep expertise in formal verification, smart contract auditing, and protocol research. With proven experience in Web3, MixBytes combines in-depth technical knowledge with a proactive security-first approach.

Why MixBytes

- **Proven Track Record:** Trusted by top-tier blockchain projects like Lido, Aave, Curve, and others, MixBytes has successfully audited and secured billions in digital assets.
- **Technical Expertise:** Our auditors and researchers hold advanced degrees in cryptography, cybersecurity, and distributed systems.
- **Innovative Research:** Our team actively contributes to blockchain security research, sharing knowledge with the community.

Our Services

- **Smart Contract Audits:** A meticulous security assessment of DeFi protocols to prevent vulnerabilities before deployment.
- **Blockchain Research:** In-depth technical research and security modeling for Web3 projects.
- **Custom Security Solutions:** Tailored security frameworks for complex decentralized applications and blockchain ecosystems.

MixBytes is dedicated to securing the future of blockchain technology by delivering unparalleled security expertise and research-driven solutions. Whether you are launching a DeFi protocol or developing an innovative dApp, we are your trusted security partner.

Contact Information

 https://mixbytes.io/

 https://github.com/mixbytes/audits_public

 hello@mixbytes.io

 https://x.com/mixbytes