LIMIT BREAK AMM

Security Audit — Personal Documentation

Guardian Defender Contest | $150,000 Prize Pool

Auditor	Gandalf The Builder
Contest Start	Feb 23, 2026
Contest End	Apr 9, 2026
Doc Updated	Feb 26, 2026
Total Hours	~45 Hours
Findings	**0 Confirmed (35+ Investigated)**

1. Overview & Context

This document is a complete personal audit record for the Limit Break AMM Guardian Defender contest. It covers all investigated areas, reasoning for discards, architectural insights learned, and skills developed during the engagement.

1.1 Contest Context

Limit Break AMM is a concentrated liquidity AMM with three pool types: Fixed (height-based), Dynamic (Uniswap V3-style tick-based), and SingleProvider (oracle/hook pricing). The protocol integrates an extensive hook system for token-level validation and fee customization, and a CLOB order book via transfer handlers.

The contest is a Guardian Defender format, meaning 5 professional auditors already completed a 3-month audit. The goal is to find what Guardian MISSED or has not yet FIXED not to re-discover known issues.

1.2 Why 0 Findings Is Not a Failure

Guardian's coverage was exceptionally thorough. Over 45 hours, 35+ candidates were investigated with proper reasoning. Every discard has a documented reason. This discipline not forcing false positives, not hallucinating bugs is the mark of a professional auditor.

- 0 false positives submitted
- Every candidate killed with verifiable on-chain reasoning
- Deep architectural understanding built over the process
- All core skill areas for future contests are now developed

2. Skills Acquired

This section documents concrete, transferable skills built during this contest. These apply directly to all future audit work.

2.1 Technical Skills

Skill	What Was Learned
Architectural Reading	Read complex multi-module codebases in hours, not days. Understand execution flow from entry point through all layers.
False Positive Elimination	Kill candidates quickly with verifiable reasoning. If 5 senior auditors didn't see it, and it looks obvious, it's almost certainly a false positive.
Hook System Design	Understand supportedFlags as explicit whitelist, hook execution ordering relative to pool type, and what hooks can/cannot affect.
Uniswap V3 Internals	Deep understanding of tick crossing, feeGrowthOutside initialization, modifyPosition ordering, and liquidity math. Applicable to all V3 forks.
Fee Accounting Patterns	Three separate key spaces for fee storage. Delta-based fee collection pattern. Protocol vs LP vs hook fee separation. Underflow protection via unchecked wrap detection.
Multi-hop Architecture	Shared swapCache across hops, partial fill guard design (CannotPartialFillAfterFirstHop), hop output chaining via amountIn = amountOut.
Assembly & Memory Safety	Read and validate inline assembly in hook execution paths. Identify memory-safe patterns vs dangerous patterns.

2.2 Methodology Skills

- Fail fast discipline: 90 minutes per candidate maximum, then PARK
- Guardian-aware auditing: check known issues before deep-diving

- Cross-module thinking: bugs that are invisible at function level but emerge across modules
- Economic attack mindset: beyond code correctness, into 'can this be abused economically?'
- Hypothesis-driven investigation: specific question first, then code lookup
- Reading audit reports professionally: extract scope, status, root cause, fix recommendation

2.3 Mindset Shifts

The most important insight from this contest:

> *Guardian defends with 'Is this function correct?' Attackers win with 'Is this system still correct after 7 valid steps?'*

The bugs that remain in well-audited codebases are almost never 'forgot a require'. They are emergent behaviors from the interaction of multiple individually-correct components.

3. Architecture Reference

Key architectural facts learned during this audit. Critical for any future work on this codebase or similar protocols.

3.1 Execution Flow

```
LimitBreakAMM.sol (entry point)
  ├── AMMModule.sol (swap + fee logic)
  ├── ModuleAdmin.sol (config, roles)
  ├── ModuleLiquidity.sol (LP operations — thin wrapper)
  ├── ModuleFeeCollection.sol (fee collection)
  └── Pool Types:
      ├── FixedPoolType → FixedHelper.sol (880 lines)
      ├── DynamicPoolType → DynamicHelper.sol (V3 clone)
      └── SingleProviderPoolType → SingleProviderHelper.sol

Hooks: AMMStandardHook.sol
Registry: CreatorHookSettingsRegistry.sol
Handlers: CLOBTransferHandler.sol, PermitTransferHandler.sol
```

3.2 Critical Architectural Facts

Fact	Detail
Pool Type Before Hook	In ALL liquidity operations, pool type contract executes BEFORE hooks. addLiquidity() runs first, then _executeAddLiquidityHooks(). This is by design.
Partial Fill Guard	LBAMM__CannotPartialFillAfterFirstHop at AMMModule.sol line 1565. Kills any partial fill on hop N+1. Shared swapCache is intentional, not a bug.
Hook Flag System	_supportedHookFlags is an EXPLICIT WHITELIST. AMMStandardHook does NOT support REMOVE_LIQUIDITY or COLLECT_FEES flags. LP can always exit.
Fee Storage Keys	Three isolated key spaces: LIQUIDITY_OWED (general), hash(hook, tokenFor, tokenFee) (hook-managed), TOKEN_MANAGED_HOOK_FEE (token-managed). Cannot cross-claim.
Global Reentrancy Lock	Single ENTERED lock — not per-flag. Cross-function reentrancy is impossible by design. Flag interactions are additive, not conflicting.
CLOB Architecture	CLOB is a transfer handler via _directSwap path — architecturally separate from AMM pool multi-hop. No intersection possible.
DynamicPoolType	Uniswap V3 clone. computeSwap uses atomic commit — all state settles before return. afterSwap hook always reads final settled state.

4. Complete Area Investigation Status

Area	Status	Notes
SecureProxy.sol (197 lines)	**SKIPPED**	Permissionless pause by design. Storage slots no collision. 0 candidates.
CLOBTransferHandler.sol (332 lines)	**SKIPPED**	H-01 Acknowledged (known). CEI correct. Arbitrary hook injection blocked by empty orderBook revert.
AMMStandardHook.sol	**SKIPPED**	FULLY EXHAUSTED. 5 candidates investigated. _supportedHookFlags confirmed explicit whitelist.
PermitTransferHandler.sol (260 lines)	**SKIPPED**	3 candidates. Nonce atomicity, feeOnTop limitAmount, EIP-712 domain all confirmed safe.
AMMModule.sol (swap/fee/hook)	**SKIPPED**	Multi-hop RV2-M-08 guard confirmed. tmpSwapCache false alarm. Fee order intended design.
ModuleLiquidity.sol	**SKIPPED**	Thin wrapper. Hook chain confirmed: token0 -> token1 -> position -> pool.
ModuleFeeCollection.sol	**SKIPPED**	Underflow wrap detection solid. Storage keys isolated. collectHookFeesByHook asymmetry intended.

Area	Status	Notes
ModuleAdmin.sol (97 lines)	SKIPPED	All fees have MAX_BPS bound. callerHasRole consistent. Hook replacement intended design.
FeeHelper.sol (83 lines)	SKIPPED	Unchecked subtraction safe. protocolFeeBPS max = MAX_BPS. 0 candidates.
DynamicPoolType.sol	SKIPPED	snapPrice M-07 self-inflicted. Swap path atomic commit. Hook timing clean. L-05 Acknowledged.
DynamicHelper.sol	SKIPPED	modifyPosition = V3 identical. _crossTick separate from _updateTick (confirmed). computeSwap clean.
FixedPoolType.sol entry (173 lines)	SKIPPED	poolFeeBPS asymmetry guarded by ZeroValueSwap. onlyAMM consistent.
FixedHelper.sol addLiquidity path	SKIPPED	getFeeGrowthInside uses delta pattern. feeGrowthOutside=0 becomes baseline, not overclaim.
SingleProviderPoolType.sol	SKIPPED	Fee on partial fill correct. RV2-M-02 variation = duplicate. 0 confirmed findings.
SingleProviderHelper.sol	SKIPPED	swapByOutput sets feeAmount + protocolFee correctly. No fee leak.
tm-core-lib (Signatures, Tstorish)	SKIPPED	ECDSA + EIP-1271 correct. low-s check present. reentrancy guard preserves ENTERED bit.
CreatorHookSettingsRegistry.sol	SKIPPED	Whitelist management only. No liquidity ops. 0 candidates.
PoolDecoder.sol	SKIPPED	Bit shifting clean. No collision. _createPool: no bypass, no re-init bug.
Swap Settlement + Reentrancy Arch	SKIPPED	limitAmount cannot be bypassed. Balance equality intentional. Global ENTERED lock.
Malicious Hook Operator Scenarios	SKIPPED	hookFee: LP protects self. LP trap: REMOVE flag not in supportedFlags. Pricing bounds: governance only.
CLOB + Multi-hop Interaction	SKIPPED	Architecturally separate. _directSwap vs _poolSwapByInput. No intersection possible.
RV2-M-01 Decimal Mismatch PoC	SKIPPED	Guardian already describe unit mismatch explicitly. Angka konkret = illustration, not new finding.
FixedHelper.sol (880 lines, swap)	SKIPPED	PERMANENT BLACKLIST. Guardian covered C-01, H-02, H-03, all RV2-M. RV2-M-08 has Guardian PoC gist.

5. All Candidates Investigated

Complete record of every hypothesis investigated with verdict and reasoning.

Candidate	Verdict	Kill Reason
tmpSwapCache amountOut sync	DISCARD	Too obvious — 5 auditors + test suite would catch instantly. Stack depth workaround pattern.
Fee order asymmetry input/output	DISCARD	Intended design. No clear extraction path identified.
RV2-M-08 multi-hop variation	DISCARD	LBAMM__CannotPartialFillAfterFirstHop guard at line 1565 kills all partial fill mid-route.
Price bounds inversion _enforcePoolCreation	DISCARD	bounds0 and bounds1 both use same sqrtPriceX96 — consistent with swap path behavior. Intended.
M-07 fix incomplete (missing hook flag)	DISCARD	Same entity controls both pricingBounds (registry) and ADD_LIQUIDITY_HOOK_FLAG (setTokenSettings). Self-inflicted.
collectFees hook missing	DISCARD	_executeLiquidityCollectFeesHooks confirmed at line 329. Hook is called correctly.
validateRemoveLiquidity always revert	DISCARD	TOKEN_SETTINGS_REMOVE_LIQUIDITY_HOOK_FLAG NOT in _supportedHookFlags (line 54-57). Dead code, unreachable.
Nonce before signature (Permit)	DISCARD	EVM transaction atomicity. State fully rolled back on revert.
feeOnTop uncommitted in permit sig	DISCARD	limitAmount protects user economic intent. Triple conjunction required for any risk.
Arbitrary permitProcessor	DISCARD	EIP-712 domain binding. Signature invalid for wrong processor address.
snapPrice M-07 fix incomplete	DISCARD	Same entity controls hook flag and pricingBounds. Misconfiguration = self-inflicted.
collectHookFeesByHook (no nonReentrant)	DISCARD	Caller restriction (msg.sender == hook) + underflow check sufficient. Self-drain only possible.
Underflow check pattern in fee transfer	DISCARD	Unchecked wrap-around detection: amountBefore < underflowCheck correctly detects underflow.
Fee leak on SingleProvider partial fill	DISCARD	swapByOutput sets swapCache.feeAmount and swapCache.protocolFee correctly before return.
RV2-M-02 variation SingleProvider	DISCARD	sqrtPriceCurrentX96 always re-fetched from hook in both paths. Partial fill no re-query = design. Duplicate.
poolFeeBPS asymmetry (> vs >= MAX_BPS)	DISCARD	FixedPool__ZeroValueSwap guard + FeeAmountExceedsInputAmount guard catch the edge case.
DynamicHelper modifyPosition ordering	DISCARD	_updateTick (liquidity) vs _crossTick (swap) are different functions. feeGrowthOutside init = V3 design.
FixedHelper feeGrowthInside initialization	DISCARD	feeGrowthInside snapshot at deposit time = baseline. Fee = delta only. Cannot overclaim history.
LP whitelist trap scenario	DISCARD	REMOVE_LIQUIDITY_HOOK_FLAG absent from _supportedHookFlags. setTokenSettings would revert if attempted.
hookFee extraction via validateCollectFees	DISCARD	LP sets maxHookFee0/maxHookFee1 themselves in liquidityParams. Hook cannot exceed user-set limit.

Candidate	Verdict	Kill Reason
`Pricing bounds economic manipulation`	**DISCARD**	Governance-level operator control. Can freeze volatility but cannot extract LP funds or corrupt accounting.
`CLOB + multi-hop interaction`	**DISCARD**	_directSwap and _poolSwapByInput are architecturally separate. No intersection possible.
`_directSwap afterSwap hook ordering`	**DISCARD**	swapAmount = pre-fee amount but hook only uses for price validation. Output fees applied after is by design.
`RV2-M-01 decimal mismatch concrete PoC`	**DISCARD**	Guardian RV2-M-01 already explicitly describes unit mismatch. Concrete numbers = illustration only. Duplicate.
`DynamicPoolType tick crossing timing`	**DISCARD**	computeSwap atomic commit — all state settles before return. afterSwap reads final state only.

6. Known Issues — Do Not Submit

Complete list of all Guardian findings. Submitting any of these will result in immediate duplicate classification.

6.1 Guardian Main Report

- C-01: Zero-Amount Cross Underflow Liquidity — Resolved
- H-01: Missing Hook In CLOB closeOrder — Acknowledged
- H-02: increaseHeight Leaves Zero Remaining — Resolved
- H-03: Split Rounding Shifts Excess Output — Resolved
- M-01 through M-10: Resolved / Acknowledged / Partially Resolved
- L-01 through L-06: Resolved / Acknowledged
- I-01 through I-07: Resolved / Acknowledged

6.2 Remediation V1

- M-01: Zero Ratio Component Bricks Swaps — Resolved
- L-01 through L-08: Resolved / Acknowledged

6.3 Remediation V2 (Critical)

- M-01: Flashloan Cross-Token Fee Wrong Units — Acknowledged, no new angle
- M-02: Hook Pricing Breaks on Partial Fills — Acknowledged, DUPLICATE confirmed
- M-03: Floor Math Arbitrage — Resolved
- M-04: swapByOutput Reverts Valid State — Resolved
- M-05: Unbacked Output Becomes Unfunded Dust — Acknowledged
- M-06: Forced Top Up Revert — Resolved

- M-07: Floor Rounding Stalls Height — Acknowledged
- **M-08: SwapByInput DoS in Valid States — PENDING. Guardian has working PoC gist. Do NOT submit anything from this area.**
- L-01 through L-05: Resolved / Acknowledged

6.4 Permanent Blacklist Areas

- **FixedHelper.sol _splitAmountsAndFeesByHeight — ALL paths blacklisted**
- **FixedHelper.sol floor math and height traversal logic — blacklisted**

7. Contest Rules & Methodology

Rule	Detail
Daily Budget	3-4 hours maximum per day
Fail Fast	90 minutes per candidate without clear progress → PARK and move on
Submit Rule	1 working PoC is worth more than 5 half-finished investigations
Variation Rule	Guardian issue + new condition or new state = VALID submission
Duplicate Rule	Exact same root cause + same fix = DISCARD, do not submit
Obvious Test	If it looks obvious and 5 senior auditors missed it — it's almost certainly a false positive

8. Final Status & Next Steps

8.1 Contest Status

Coverage Status	~98% of codebase exhausted
Confirmed Findings	**0 (35+ candidates investigated and discarded with reasoning)**
Hours Invested	~45 hours across multiple sessions
Contest Remaining	~43 days (ends Apr 9, 2026)

8.2 Honest Assessment

This was a Guardian Defender contest, the hardest contest format that exists. 5 professional auditors for 3 months. 0 confirmed findings after 45 hours by a first contest auditor is not a failure. It is a realistic outcome and demonstrates proper discipline.

The value gained from this contest is not a payout, it is the architectural knowledge, methodology, and mindset that will apply directly to every future audit engagement.

End of Document — Limit Break AMM Audit Record