АУДИТ БЕЗОПАСНОСТИ

VolumeDynamicFeeHook

v2.4.0 · 13 апреля 2026 г.

ОБЪЕКТ	VolumeDynamicFeeHook (src/VolumeDynamicFeeHook.sol, 2 097 строк)
ВЕРСИЯ	v2.4.0
ДАТА	13 апреля 2026 г.
СТАТУС	Final
РЕПОЗИТОРИЙ	github.com/Axel-DeFi/VolumeDynamicFeeHook
АУДИТ	Выполнен · Critical: 0 / High: 0 / Medium: 0 / Low: 0
ГЛУБИНА АУДИТА	Deep
СРЕДА ИСПОЛНЕНИЯ	EVM L1 / L2, Solidity ^0.8.26
АДРЕС ХУКА	`0x2C3254Da...5044`
АДРЕС ПУЛА	`0x226d6297...974c`

1. СРЕДА АУДИТА

Компонент	Версия / идентификатор
Модель	claude-opus-4-6 · (reasoning: max)
Foundry (forge)	1.5.1
Solidity (solc)	0.8.26
Slither	0.11.5
Echidna	2.3.2
Halmos	0.3.3

Аудируемая ревизия: `53147aae` (ветка `main`, чистое рабочее дерево).

2. РЕЗЮМЕ

Проведён полный глубокий аудит безопасности контракта `VolumeDynamicFeeHook` версии v2.4.0 на ревизии `53147aae`. Выполнены все обязательные проверки глубокого аудита: ручной анализ кода (2 097 строк), полный набор тестов Foundry (201 тест, включая инвариантные кампании и фаззинг), статический анализ Slither с ручной сортировкой сигналов, проверка свойств Echidna (5 свойств), символьная верификация Halmos (16 свойств).

Подтверждённых уязвимостей уровней Critical, High, Medium и Low не обнаружено. Зафиксировано пять информационных наблюдений, относящихся к архитектурным решениям и внешним допущениям. Конфигурационно-зависимые риски описаны отдельно и относятся к области ответственности оператора.

Контракт демонстрирует зрелую архитектуру безопасности: строгую валидацию параметров в конструкторе и административных функциях, упакованное состояние без перекрытия полей, ограниченную поверхность внешних вызовов, детерминированный автомат состояний с подтверждениями и защитой удержания, 48-часовой таймлок для изменения процента комиссии хука, двухэтапную передачу владения и аварийный сброс с немедленным действием.

3. ОБЪЕКТ И ОБЛАСТЬ АУДИТА

Контракт `VolumeDynamicFeeHook` — это хук для пулов Uniswap v4 с динамической комиссией, который отслеживает объём торгов и автоматически переключает ставку комиссии для поставщиков ликвидности между тремя режимами:

- `FLOOR` (режим 0) — минимальная комиссия при низком объёме;
- `CASH` (режим 1) — средняя комиссия при нормальном объёме;
- `EXTREME` (режим 2) — максимальная комиссия при высоком объёме.

Переходы между режимами управляются конечным автоматом с требованием накопления подтверждающих периодов и механизмом защиты удержания (hold), который предотвращает преждевременный откат после повышения. Аварийный сброс (`emergencyReset`) позволяет владельцу принудительно перевести контракт в режим `FLOOR` или `CASH`, игнорируя hold-защиту.

Помимо переключения комиссий пула, контракт взимает собственную надбавку (Hook Fee) — процент от оценочной комиссии LP, начисляемый как ERC6909-требование в `PoolManager`. Владелец может забрать накопленные средства через функцию `claimHookFees`.

Область аудита включает:

- основную логику хука: обработка свопа, вычисление и начисление комиссий, отслеживание объёма;
- автомат переходов между режимами, включая «догоняющее» закрытие просроченных периодов, сброс при бездействии (Idle Reset) и сброс при низком объёме;
- административные функции: управление параметрами, таймлок, передача владения, приостановка и аварийный сброс;
- вывод средств: пошаговый вывод ERC6909-требований через учётную систему `PoolManager`;
- упаковку и распаковку состояния в единый слот `uint256`;
- смежные поверхности, способные повлиять на защищаемые свойства.

4. ЗАЩИЩАЕМЫЕ СВОЙСТВА И УРОВНИ МАТЕРИАЛЬНОСТИ

Аудит оценивает контракт относительно следующих защищаемых свойств, упорядоченных по приоритету для заинтересованных сторон.

Приоритет	Защищаемое свойство	Материальность
1	Сохранность средств LP и защита от несанкционированного перемещения активов	`PRIMARY`
2	Целостность привилегий: контроль над комиссиями, режимами и состоянием принадлежит только авторизованным субъектам	`PRIMARY`
3	Целостность политики комиссий: переходы режимов, границы комиссий и правила обновления соответствуют заявленной модели	`PRIMARY`
4	Живучесть пула и безопасность восстановления	`PRIMARY`
5	Учётная целостность хука как системы	`SECONDARY`
6	Путь вывода дохода владельца	`ADMINISTRATIVE`
7	Ясность мониторинга, документации и операционная эргономика	`CONTEXTUAL`

Каждое наблюдение и риск в данном отчёте сопровождаются указанием домена воздействия и уровня материальности. Информационные наблюдения и административные риски не помещаются на один повествовательный уровень с рисками, затрагивающими сохранность средств LP или целостность привилегий.

5. МОДЕЛЬ НАРУШИТЕЛЯ И ГРАНИЦЫ ДОВЕРИЯ

Аудит исходит из трёх классов нарушителей и одного допущения о среде исполнения.

Класс нарушителя	Возможности
Внешний пользователь	Вызов доступных внешних функций, подача произвольных входных данных, повторные вызовы, попытки нарушить инварианты и вызвать отказ в обслуживании
MEV-нарушитель	Переупорядочивание транзакций, эксплуатация порядка исполнения и кратковременных состояний, попытки манипулирования комиссиями и принудительных переходов режимов
Привилегированный владелец	Исполнение всех предусмотренных привилегированных действий, изменение параметров в допустимых границах, возможное доведение системы до небезопасной конфигурации при недостаточных ограничениях

Допущение о враждебной среде: внешние контракты, токены, обратные вызовы и значения, поступающие извне, рассматриваются как потенциально враждебные.

Границы доверия контракта установлены следующим образом.

Зависимость	Уровень доверия	Обоснование
`PoolManager` (Uniswap v4)	Доверенный	Вызовы `afterSwap`, `updateDynamicLPFee`, `mint`, `burn`, `take` направляются только к `poolManager`, адрес которого задан в конструкторе и неизменяем
Владелец (`_owner`)	Доверенный с ограничениями	Может изменять параметры только в пределах жёстко заданных границ; изменение `hookFeePercent` требует 48-часового таймлока; не может извлечь средства LP; не может обойти ограничения `MAX_HOOK_FEE_PERCENT`, порогов и диапазонов подтверждений

Зависимость	Уровень доверия	Обоснование
Входные данные свопов	Недоверенные	Проверяются: ключ пула, объём (фильтр пылевого порога), результат свопа
`block.timestamp`	Допущение среды	Зависимость от метки времени блока для учёта периодов, таймлока и бездействия; манипулирование валидаторами не создаёт критического воздействия благодаря длительности периодов (часы-дни)
Токены пула	Доверенные с ограничениями	Предполагается стандартное поведение ERC20; нестандартные токены (с комиссией на перевод, перебазируемые) могут нарушить учёт объёма

6. КАРТА ПОВЕРХНОСТИ АТАКИ

Поверхность	Затрагиваемое свойство	Воздействие	Тип нарушения	Материальность
`afterSwap` — точка входа обработки свопов	Политика комиссий, учёт объёма, hook fee	Внешний пользователь (через `PoolManager`)	Манипулирование объёмом, принудительные переходы	`PRIMARY`
`setModeFees` — установка ставок комиссий	Политика комиссий	Владелец	Некорректные ставки (ограничено валидацией `floor` < `cash` < `extreme`)	`PRIMARY`
`setControllerSettings` — пороги, подтверждения, hold	Политика комиссий, живучесть	Владелец	Невозможные или слишком лёгкие переходы	`PRIMARY`
`scheduleHookFeeChange` / `applyHookFeeChange` — таймлок	Сохранность средств, привилегии	Владелец	Обход таймлока, чрезмерная надбавка (`MAX_HOOK_FEE_PERCENT` = 10)	`PRIMARY`
`emergencyReset` — аварийный сброс (только в паузе)	Живучесть, политика комиссий	Владелец	Принудительный переход в обход hold	`PRIMARY`
`pause` / `unpause` — приостановка	Живучесть	Владелец	Блокировка обработки свопов (пул продолжает работать)	`PRIMARY`
`proposeNewOwner` / `acceptOwnership` — передача	Привилегии	Владелец, предложенный владелец	Компрометация привилегий при утрате ключа	`PRIMARY`
`claimHookFees` — вывод комиссий хука	Доход владельца	Владелец	Вывод не более накопленного; средства LP не затронуты	`ADMINISTRATIVE`
`_withdrawCurrencyClaim` — пошаговый вывод	Доход владельца	Владелец (через `claimHookFees`)	Цикл с ограничением `int128` — не эксплуатируется извне	`ADMINISTRATIVE`
`rescueToken` / `rescueETH` — спасение активов	Эргономика	Владелец	Доступ только к балансу контракта, не к ERC6909 в `PoolManager`	`CONTEXTUAL`

Поверхность	Затрагиваемое свойство	Воздействие	Тип нарушения	Материальность
Упакованное состояние `_state` (`uint256`, 9 полей, 248 бит)	Учётная целостность, переходы	Внутренний (`_packState` / `_unpackState`)	Перекрытие битовых полей, потеря данных	`PRIMARY`

7. МЕТОДОЛОГИЯ И ФАКТИЧЕСКОЕ ПОКРЫТИЕ

Аудит выполнен в пять этапов.

#	Этап	Метод	Покрытие
1	Ручной анализ кода	Последовательное чтение всех 2 097 строк контракта, анализ спецификации `docs/SPEC.md` (386 строк), инвариантных тестов (406 строк), фаззинг-тестов (241 строка), символьных тестов (815 строк), интеграционных тестов учёта claim-операций	Полный контракт, включая конструктор, обработку свопа, автомат переходов, административные функции, упаковку состояния, вычисление комиссий, вывод средств
2	Forge test suite	201 тест в 21 наборе: 26 конфигурационных и граничных тестов, 61 административный тест, 7 интеграционных тестов учёта, 28 инвариантных тестов (4 конфигурации × 7 инвариантов × 128 запусков × 8 192 вызова), 4 фаззинг-теста (256 запусков каждый), вспомогательные тесты библиотек	Все основные поверхности: инварианты режимов, границы комиссий, учёт hook fee, упаковка состояния, таймлок, hold-защита, конфигурация
3	Slither	Статический анализ с фокусом на основной контракт (`--exclude-dependencies`); ручная сортировка всех сигналов	42 сигнала средней серьёзности, 41 низкой, 4 информационных — все проанализированы и классифицированы
4	Echidna	5 свойств, 50 197 вызовов, режим тестирования свойств	Инварианты pack/unpack, границы `feeIdx`, переполнение EMA, насыщение объёма, корректность переходов режимов
5	Halmos	16 символьных свойств, 842 пути, без ограничения по таймауту решателя	Тождественность pack/unpack, границы всех упакованных полей, порядок комиссий, насыщение арифметики, аварийный сброс, hold-защита, подтверждения переходов, idle reset, clamp hold

8. РЕЗУЛЬТАТЫ АВТОМАТИЗИРОВАННОГО АНАЛИЗА

Forge (сборка и полный набор тестов)

Сборка завершена без предупреждений (solc 0.8.26, 112 файлов). Все 201 тест пройдены, 0 провалено, 0 пропущено. Инвариантные кампании выполнены по четырём конфигурациям (два стабильных токена × два значения `tickSpacing`), каждая по 128 запусков × 8 192 вызова обработчика с 11 операциями — без нарушений и сбоев.

Вид теста	Количество	Результат
Модульные и конфигурационные	128	128 / 128 пройдено
Административные (включая 2 фаззинг-теста)	61	61 / 61 пройдено
Инвариантные (4 конфигурации × 7 свойств)	28	28 / 28 пройдено
Фаззинг-тесты длинных последовательностей	2	2 / 2 пройдено (256 запусков)
Интеграционные (учёт claim)	7	7 / 7 пройдено

Проверяемые инварианты включают: `feeIdx` всегда в допустимых пределах, строгий порядок модальных комиссий, все упакованные поля в пределах битовых масок, консистентность таймлока, соответствие активного hold настройкам режима, совпадение текущей комиссии LP с текущим режимом, корректность учёта hook fee на каждом шаге.

Slither (статический анализ)

Запущен на `src/VolumeDynamicFeeHook.sol` с исключением зависимостей. Результат: 0 сигналов высокой серьёзности, 42 средней, 41 низкой, 4 информационных.

Все 42 сигнала средней серьёзности проанализированы вручную и классифицированы как ложные срабатывания или допустимое поведение. Основные категории ложных срабатываний:

Категория	Кол-во	Причина ложного срабатывания
Сравнение `enum`-значений с константами	~15	Намеренное использование целочисленных констант `MODE_FLOOR`, `MODE_CASH`, `MODE_EXTREME` вместо `enum`-типа — архитектурное решение для совместимости с упаковкой состояния
Деление перед умножением	~10	Намеренное округление вниз в `_hookFeeAmount` — в пользу LP, документировано в спецификации
Предупреждения о повторном входе через `poolManager`	~8	Все внешние вызовы к `poolManager` происходят внутри управляемого контекста `unlock`; `poolManager` — доверенная зависимость
Прочие (неиспользуемые возвраты, затенение)	~9	Намеренные архитектурные решения, не несущие риска безопасности

Ни один сигнал не указывает на реальную уязвимость или неучтённый паттерн.

Echidna (тестирование свойств)

Все 5 свойств пройдены за 50 197 вызовов, контрпримеры не найдены.

Свойство	Результат
Тождественность упаковки и распаковки состояния	Пройдено
`feeIdx` всегда в допустимых пределах после произвольных переходов	Пройдено
EMA не превышает `uint96.max`	Пройдено
Объём насыщается на `uint64.max`	Пройдено
Переходы режимов соответствуют правилам автомата	Пройдено

Halmos (символьная верификация)

Все 16 символьных проверок пройдены, 842 символьных пути исследованы, контрпримеры не найдены.

Свойство	Пути	Результат
`check_packUnpackRoundtrip` — тождественность упаковки/распаковки	11	Пройдено
`check_feeIdxAlwaysBounded` — `feeIdx` ∈ {0, 1, 2}	320	Пройдено
`check_streakCountersNeverExceedBitWidth` — счётчики streak/hold в пределах битовых масок	320	Пройдено

Свойство	Пути	Результат
`check_feeOrderingPreserved` — строгий порядок `floorFee` < `cashFee` < `extremeFee`	4	Пройдено
`check_emaUpdateSaturatesAt96bit` — насыщение EMA на `uint96.max`	8	Пройдено
`check_volumeAdditionSaturatesAt64bit` — насыщение объёма на `uint64.max`	4	Пройдено
`check_hookFeeNonNegativeAndClamped` — hook fee ≥ 0 и ≤ `int128.max`	9	Пройдено
`check_emergencyFloorPreemptsHold` — аварийный сброс игнорирует hold	10	Пройдено
`check_modeFeeSelectorConsistency` — каждый режим возвращает свою ставку	7	Пройдено
`check_holdBlocksOrdinaryCashToFloor` — hold блокирует CASH → FLOOR	13	Пройдено
`check_holdBlocksOrdinaryExtremeToCash` — hold блокирует EXTREME → CASH	10	Пройдено
`check_extremeCanReachFloorOnlyViaEmergency` — EXTREME → FLOOR только через аварийный сброс	85	Пройдено
`check_idleResetClearsRuntimeState` — idle reset обнуляет все счётчики	4	Пройдено
`check_cashToFloorNeedsFullConfirms` — CASH → FLOOR требует полный набор подтверждений	12	Пройдено
`check_extremeToCashNeedsFullConfirms` — EXTREME → CASH требует полный набор подтверждений	10	Пройдено
`check_controllerSettingsClampActiveHold` — ограничение hold при смене настроек	10	Пройдено

Символьная верификация покрывает все критические арифметические свойства (переполнение, насыщение, границы), инварианты автомата переходов (hold-защита, подтверждения, аварийный сброс) и корректность упаковки состояния.

9. ПРОБЕЛЫ ПОКРЫТИЯ И ВОПРОСЫ, ТРЕБУЮЩИЕ ПРОВЕРКИ

Область	Описание	Влияние на выводы аудита
Нестандартные токены	Контракт предполагает стандартное поведение ERC20. Тесты используют стандартные моковые токены. Поведение с токенами, взимающими комиссию на перевод (fee-on-transfer) или перебазируемыми токенами (rebase), не верифицировано.	Не является уязвимостью контракта — это допущение при развёртывании. Пул привязывается к конкретной паре при создании.
Экономическое моделирование MEV-сценариев	Фаззинг и инвариантные тесты проверяют корректность переходов и учёта, но не моделируют стратегического MEV-нарушителя, оптимизирующего прибыль через последовательности свопов.	Защищаемые свойства контракта не нарушаются при MEV: границы комиссий и переходы соблюдаются. Экономическая оптимальность параметров — ответственность оператора.
Поведение при длительном бездействии пула	Сценарий, когда `idleResetSeconds` истекает через множество периодов с последующим резким всплеском активности, покрыт на уровне автомата (idle reset обнуляет состояние), но не как полный экономический сценарий.	Автомат детерминирован; idle reset корректно обнуляет все счётчики (подтверждено символьно).

Все инструменты из обязательного чеклиста глубокого аудита выполнены без пропусков и сбоев. Блокирующих факторов не выявлено.

10. ТАБЛИЦА НАХОДОК

Подтверждённых уязвимостей уровней Critical, High, Medium и Low не обнаружено.

11. КАРТОЧКИ ПОДТВЕРЖДЁННЫХ НАХОДОК

Подтверждённых уязвимостей не обнаружено. Раздел пуст.

12. ИНФОРМАЦИОННЫЕ НАБЛЮДЕНИЯ

ID	Название	Домен воздействия	Материальность
I-01	Округление вниз при вычислении hook fee	`FEE_POLICY_INTEGRITY`	`CONTEXTUAL`
I-02	Семантика «догоняющего» закрытия просроченных периодов	`FEE_POLICY_INTEGRITY`	`CONTEXTUAL`
I-03	Внешнее допущение о безопасности ключа владельца	`PRIVILEGE_INTEGRITY`	`CONTEXTUAL`
I-04	Остаточный экономический риск манипулирования объёмами	`FEE_POLICY_INTEGRITY`	`CONTEXTUAL`
I-05	Пошаговый вывод средств с ограничением `int128`	`OWNER_REVENUE_PATH`	`ADMINISTRATIVE`

I-01. Округление вниз при вычислении hook fee. Функция `_hookFeeAmount` (строка 1641) выполняет два последовательных целочисленных деления: сначала вычисляется оценка комиссии LP как `absUnspecified * appliedFeeBips / FEE_SCALE`, затем от неё берётся доля владельца как `lpFeeAmount * hookFeePct / 100`. Каждое деление отбрасывает дробную часть, что систематически занижает начисляемую надбавку на несколько wei в пользу трейдера и LP. Это намеренное архитектурное решение, документированное в спецификации: хук не должен завышать свою долю. Защищаемые свойства не нарушены.

I-02. Семантика «догоняющего» закрытия просроченных периодов. Когда с последнего свопа прошло более одного периода, функция `_closePeriodsIfNeeded` последовательно закрывает все просроченные периоды. При этом первый закрываемый период получает весь объём текущего свопа, а последующие закрываются с нулевым объёмом. Цикл ограничен механизмом сброса при бездействии: если пауза превышает `idleResetSeconds`, состояние обнуляется в одну операцию без итерирования. При максимальном отношении `idleResetSeconds / periodSeconds` (ограничено константой `MAX_IDLE_PERIODS` = 23) цикл совершает не более 23 итераций. Это документированный архитектурный компромисс (SPEC.md, раздел «Overdue Catch-Up»), принятый ради простоты и детерминированности. Защищаемые свойства не нарушены.

I-03. Внешнее допущение о безопасности ключа владельца. Владелец может изменять все конфигурируемые параметры в пределах жёстко заданных границ, приостанавливать обработку свопов, выполнять аварийный сброс и выводить накопленные комиссии хука. Все привилегированные действия ограничены: `hookFeePercent` ≤ 10 с 48-часовым таймлоком, модальные комиссии проходят перекрёстную валидацию, параметры контроллера ограничены максимальными константами. При компрометации ключа нарушитель может установить максимально невыгодные (но ограниченные) параметры и вывести только накопленные комиссии хука — доступа к средствам LP нет. Двухэтапная передача владения исключает случайную потерю контроля. Безопасность ключа является внешней ответственностью.

I-04. Остаточный экономический риск манипулирования объёмами. Объём торгов отслеживается по стабильной стороне пула без внешнего ценового оракула. Нарушитель с достаточным капиталом может раздуть объём через wash trading (встречные свопы) для ускорения перехода в режим `EXTREME`. Воздействие ограничено конструкцией: границы комиссий задаются владельцем, переходы требуют накопления подтверждений в течение нескольких периодов, hold-защита предотвращает немедленный откат, а пылевой фильтр (`dustSwapThreshold`) отсекает малые свопы. Стоимость атаки

включает комиссии LP на каждый сфабрикованный своп, что делает её экономически невыгодной для большинства реалистичных конфигураций. Это задокументированный остаточный риск (SPEC.md).

I-05. Пошаговый вывод средств с ограничением `int128`. Функция `_withdrawCurrencyClaim` (строка 1540) выводит ERC6909-требования из `PoolManager` порциями размером не более `int128.max`, поскольку учётная система `PoolManager` оперирует знаковыми 128-битными числами. Для реалистичных объёмов накопленных комиссий цикл совершает ровно одну итерацию. Теоретический максимум итераций определяется как `ceil(amount / int128.max)`, что для `uint256.max` составляет 2 итерации. Функция доступна только через `claimHookFees`, вызываемую исключительно владельцем. Эксплуатация извне невозможна.

13. КОНФИГУРАЦИОННО-ЗАВИСИМЫЕ РИСКИ

Перечисленные ниже условия не являются уязвимостями контракта. Они описывают области конфигурации, в которых оператор может создать неоптимальное или нежелательное поведение в пределах допустимых значений параметров.

ID	Условие	Воздействие	Рекомендуемая граница
CS-01	Слишком малое значение `periodSeconds`	Увеличивает число итераций «догоняющего» закрытия при бездействии, повышает расход газа первого свопа после паузы	Использовать значения от 3 600 секунд (1 час) и выше
CS-02	Близкие значения модальных комиссий (`cashFee` ≈ `extremeFee`)	Сужает полезный диапазон динамической стратегии, делая переходы экономически малозначимыми	Обеспечить значимый разрыв между уровнями
CS-03	Малое значение `idleResetSeconds`	Частый сброс накопленного состояния контроллера при естественных паузах в торговле	Установить не менее нескольких периодов бездействия
CS-04	Максимальное значение `hookFeePercent` (10) при высоких модальных комиссиях	Надбавка до 10% от оценочной комиссии LP в режиме `EXTREME` может быть значимой для трейдеров	Калибровать `hookFeePercent` с учётом абсолютных ставок комиссий

Все перечисленные параметры валидируются контрактом при установке. Комбинации, приводящие к невозможным переходам или нарушению порядка комиссий, отклоняются на уровне кода.

14. ОСТАТОЧНЫЕ РИСКИ

Риск	Описание	Ограничивающий фактор
Компрометация ключа владельца	Нарушитель получает доступ ко всем привилегированным действиям в пределах жёстких ограничений контракта	Максимальный ущерб ограничен: доступа к средствам LP нет, `hookFeePercent` ≤ 10 с таймлоком, вывод только накопленных комиссий хука
Манипулирование объёмами (wash trading)	Искусственное раздувание объёма для ускорения перехода в высокий режим	Стоимость атаки (комиссии LP), требование подтверждающих периодов, hold-защита, пылевой фильтр
Корректность `PoolManager` v4	Контракт доверяет вызовам и учётной системе `PoolManager`	Внешнее допущение; `PoolManager` — аудированный протокольный компонент Uniswap v4

Риск	Описание	Ограничивающий фактор
Нестандартные токены	Токены с комиссией на перевод или перебазированием могут нарушить точность учёта объёма	Допущение при развёртывании; пара токенов фиксируется в конструкторе
Точность метки времени блока	Валидаторы могут незначительно смещать `block.timestamp`	Периоды измеряются в часах-днях; отклонение в секунды не влияет на защищаемые свойства

15. ИТОГОВЫЙ ВЕРДИКТ

Контракт `VolumeDynamicFeeHook` версии v2.4.0 на ревизии `53147aae` прошёл полный глубокий аудит безопасности со всеми обязательными проверками.

Подтверждённых уязвимостей уровней Critical, High, Medium и Low не обнаружено. Пять информационных наблюдений относятся к документированным архитектурным решениям и внешним допущениям, ни одно из которых не нарушает защищаемые свойства.

Контракт демонстрирует последовательную оборонительную архитектуру: жёсткая валидация всех параметров при установке и в конструкторе, детерминированный автомат переходов с требованием подтверждений, упакованное состояние без перекрытия полей (подтверждено символьно), ограниченные привилегии владельца с таймлоком, двухэтапная передача владения, корректный порядок операций при выводе средств (сначала обновление внутреннего учёта, затем внешний вызов), насыщающая арифметика для EMA и объёма.

Набор тестов покрывает все критические поверхности: 201 тест Foundry (включая 28 инвариантных кампаний с 4 194 304 вызовами обработчика суммарно), 5 свойств Echidna (50 197 вызовов), 16 символьных проверок Halmos (842 пути). Статический анализ Slither не выявил ни одного реального сигнала безопасности.

Остаточные риски ограничены внешними допущениями (безопасность ключа владельца, корректность `PoolManager`, стандартность токенов) и экономическим резидуальным воздействием (wash trading), которое ограничено конструкцией контракта и стоимостью атаки.