

Методология проверки безопасности | CommitScope 2.4.2

01 / Решение для компании

Какую задачу решаем

Компании нужна воспроизводимая проверка точного коммита Git: сотрудник запускает ее локально через Claude Code, передача исходников отдельно разрешена, затем защищенные материалы получает назначенный ревьюер. Полезный результат - не зеленый значок, а ограниченный пакет свидетельств, на основе которого человек принимает и фиксирует решение.

Рекомендация: использовать CommitScope 2.4.2 для локального сбора и проверки материалов. Semgrep, Gitleaks, Trivy, независимые Claude Hunter и Verifier, а также разбор человеком обязательны. READY_FOR_HUMAN_REVIEW означает передачу ревьюеру, а не одобрение.

Это рациональный выбор при согласованных ограничениях, а не измеренное утверждение о лучшем обнаружении, минимальной цене, формальном соответствии или полной безопасности. Он не заменяет моделирование угроз, тесты авторизации, динамическое тестирование, пентест и ответственное принятие риска. Обзор источников проверен 2026-09-23. [S1] [S3] [S4] [S9]

02 / Мировая практика

Дополняющие уровни, а не конкурирующие значки

NIST SSDF 1.1 - финальная высокоуровневая модель безопасной разработки; редакция 1.2 на дату обзора остается проектом. OWASP SAMM организует развитие практик по функциям Governance, Design, Implementation, Verification и Operations. OWASP ASVS 5.0.0 дает проверяемые требования к защитным механизмам приложения. Эти документы формируют программу и вопросы проверки; ни один не утверждает, что один инструмент закрывает все практики. [S1] [S2] [S3] [S4]



Рисунок 1. Уровни программы безопасности; CommitScope покрывает часть проверки и сохранения материалов.

Ревью кода и моделирование угроз выясняют, подходит ли защита бизнес-процессу. SAST, поиск секретов, SCA и проверка IaC находят повторяемые классы ошибок; динамические тесты проверяют поведение работающей системы. ИИ может предлагать и оспаривать гипотезы о путях в коде, но комментарии репозитория и внешние данные недоверенны. OWASP выделяет prompt injection как отдельный риск; Anthropic описывает ограничения и меры защиты, но не иммунитет. [S3] [S5] [S6] [S9]

03 / Сравнение подходов

Качественная оценка в наших условиях

Следующие выводы - оценка команды CommitScope для согласованного локального процесса с аккаунтом Claude. Это не баллы NIST, OWASP или результаты бенчмарка: покрытие и затраты зависят от репозитория, политики, правил сканеров, модели и квалификации ревьюера.

Таблица 1. Подходы: тип покрытия, повторяемость и передача исходников

Подход	Тип покрытия	Повторяемость	Конфиденциальность / передача
Только человек	Глубокий контекст, ограничен временем	Зависит от процесса	Может остаться локальной
Только сканеры	Шаблоны ошибок, зависимости, настройки	Высока при закрепленных правилах и БД	Обычно локально; БД обновляется по сети
Только ИИ	Контекстные гипотезы, переменная точность	Зависит от модели и запроса	Запрос с кодом получает провайдер
Центральный сервис	Зависит от стека и контролей	Возможна единая политика	Нужна разрешенная централизованная обработка
Локальная комбинация	Машинные проверки плюс проверка и оспаривание гипотез	Точный коммит и версии записаны	Ограниченный пакет по согласию

Таблица 2. Подходы: нагрузка, источники затрат и проверяемость материалов

Подход	Нагрузка	Источники затрат	Проверяемость
Только человек	Подготовка и повторные проверки	Время эксперта	Нужны дисциплинированные записи
Только сканеры	Правила, БД, исключения	Вычисления и разбор	Можно хранить сырой и нормализованный вывод
Только ИИ	Управление запросом, моделью, приватностью	Квота и проверка человеком	Слаба без записи входа и выхода
Центральный сервис	Идентичность, хостинг, права	Платформа и сопровождение	Может быть сильной при настройке
Локальная комбинация	Настройка сотрудника и передача	Сканеры, квота, время ревьюера	Манифест и защищенные материалы; без подписи

Комбинированный вариант выбран потому, что соответствует разрешенной модели запуска сотрудником и защищенной передачи, сохраняя дополняющие материалы. Сам по себе он не закрывает поведение в работе, злоупотребление бизнес-логикой, эксплуатацию в продакшене и все корпоративные политики. [S1] [S4] [S9]

04 / Почему такая реализация

Критерии согласованного процесса

Таблица 3. Обязательные условия и выбранное решение

Условие компании	Решение CommitScope	Оставшийся контроль
Нужен аккаунт Claude Code	Hunter и Verifier только через аккаунт, без API fallback	Утвердить аккаунт и условия данных
Локальный запуск сотрудником	Одна команда commitscope review	Обучить и уполномочить
Точно проверяемый код	Чистый полный ID коммита и неизменяемый снимок	Управление областью и изменениями
Разрешенная передача кода	Явный --allow-code-upload и ограниченный отобранный пакет	Владелец разрешает передачу
Защищенная передача	Новый закрытый каталог вне цели	Правила хранения и доступа
Назначенный ревьюер	verify-review, нормализованные и сырые материалы	Личность, решение, исключения
Поддерживаемые машины	macOS и glibc Linux x86_64/ARM64	Учет и обновления хостов

Одному человеку трудно регулярно повторять поиск машинных шаблонов в каждом коммите. Одни сканеры не отвечают на все вопросы достижимости и бизнес-инвариантов. Один ИИ не дает детерминированного базового покрытия и может принять инструкции из недоверенного кода. Централизованный сервис может понадобиться позже, но меняет требования к хранению, идентичности и сопровождению. Ни один подход не хуже всегда; локальная комбинация подходит именно этим ограничениям.

Сохраняем внешние контроли: модель угроз и требования на основе ASVS; динамические тесты и разрешенное воспроизведение; защиту репозитория; личность ревьюера и принятие риска; разрешение передачи; хранение и реагирование. CommitScope собирает материалы, но не получает полномочия владельцев.

05 / Архитектура и доверие

Что остается локально и что пересекает границу

Сотрудник выбирает чистый точный коммит и утвержденную политику вне цели. CommitScope экспортирует снимок, выполняет закрепленные Semgrep, Gitleaks и Trivy и сохраняет результаты в закрытом каталоге. Затем ищет распознаваемые секреты и формирует ограниченный пакет исходников. Обновление сканеров и БД - отдельная сетевая зависимость. Успешный фильтр не гарантирует отсутствие неизвестных секретов. Версия 2.4.2 добавляет пять ограниченных правил Semgrep для Java: прямую конкатенацию SQL в JDBC, запуск процессов, десериализацию и отключенные проверки TLS. Это не исчерпывающее покрытие Java.

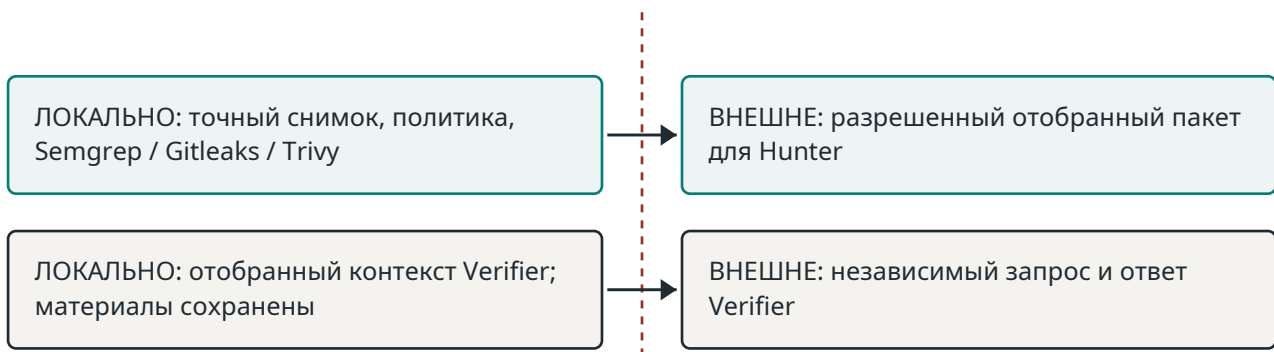


Рисунок 2. Пунктир: хост сотрудника и сервис аккаунта Anthropic.

Hunter и Verifier - отдельные вызовы Claude Code через аккаунт. Сотрудник обязан разрешить отправку отобранного пакета и проверить условия, настройки приватности и сроки хранения нужного аккаунта: персональные и корпоративные условия различаются. Репозиторий и ответы сканеров/модели недоверенны. Anthropic описывает защитные меры и использование данных, но полномочие на передачу устанавливает организация. [S5] [S6] [S7] [S8]

Недостаточное покрытие сканеров, сбой входа, недоступная модель, тайм-аут, неверный формат ответа или неполные Hunter/Verifier дают INCOMPLETE. Запуск сохраняет защищенные материалы; манифест записывается последним и не подписан. verify-review проверяет согласованность, а не личность автора или неизменность хранения. Решение остается за человеком.

06 / От гипотезы до исправления

Материалы должны выдержать опровержение и перепроверку



Рисунок 3. Жизненный цикл находки; новый коммит требует полного нового запуска.

Синтетический пример: уязвимый обработчик заказа принимает ID и получает запись, не проверяя owner = текущий пользователь. Hunter предлагает путь IDOR; Verifier ищет аутентификацию, достижимый обработчик и возможную проверку владельца ниже по цепочке. Ревьюер подтверждает или отклоняет гипотезу по коду и разрешенным тестам. Исправление добавляет поиск с проверкой владельца; новый коммит проверяется полностью. Пример иллюстрирует метод, а не измеренную точность модели.

Исчезновение находки после исправления не доказывает устранение. Сравните два точных коммита, проверьте инвариант в разрешенной среде, изучите пропуски и контрдоводы, зафиксируйте решение ревьюера для каждого запуска.

07 / Контролируемое внедрение

Предлагаемый порядок и материалы пилота



Рисунок 4. Предлагаемые этапы внедрения с решениями людей.

Таблица 4. Что измерять в пилоте, а не заявлять как результат

Показатель	Способ сбора	Владелец решения
Завершено / INCOMPLETE	Состояния и причины сбоев	Владелец инструмента
Нагрузка ревьюера	Время от передачи до решения	Руководитель ревью
Подтвержденные / отклоненные гипотезы	Журнал решений и причин	Руководитель ИБ
Исключения передачи кода	Разрешения и инциденты	Владелец данных
Дрейф правил и модели	Повтор контролируемых примеров	Владелец инструмента

Исключение оформляется вне инструмента: владелец, область, срок и компенсирующая проверка. После изменений БД сканеров, закрепленных инструментов, доступности модели, политики или условий передачи нужна перепроверка. Нельзя незаметно превратить сбой обязательного этапа в успешное ревью.

08 / Ограничения и источники

Что именно обосновано

Обзор практик проверен 2026-09-23. NIST SSDF 1.1 - финальная версия; редакция 1.2 остается проектом, поэтому здесь она не представлена обязательным окончательным стандартом. Страницы OWASP и Anthropic могут меняться; при обновлении политики их надо сверять снова. [S1] [S2]

Остаются риски неподдерживаемых языков и манифестов, ложных и пропущенных находок сканеров, ошибок модели и prompt injection, неизвестных секретов в пакете, изменений БД уязвимостей, неподписанных материалов, подмены хранилища и человеческого суждения. Фиксируйте область и пропуски. Чистый запуск не сертифицирует репозиторий и не разрешает слияние. [S5] [S6]

Пошаговые действия сотрудника и ревьюера описаны в парном руководстве пользователя CommitScope. Корпоративный путь - исходный защищенный запуск плюс отдельное решение человека; отчеты сканеров CI дают лишь частичные материалы.

Источники и дата проверки

[S1] NIST SP 800-218, SSDF 1.1 (финальная версия, 2022). <https://csrc.nist.gov/pubs/sp/800/218/final> (2026-09-23)

[S2] Публикации NIST SSDF: SP 800-218 Rev. 1 / SSDF 1.2 (проект). <https://csrc.nist.gov/projects/ssdf/publications> (2026-09-23)

[S3] OWASP SAMM: модель и функции. <https://owaspsamm.org/about/> (2026-09-23)

[S4] OWASP Application Security Verification Standard 5.0.0. <https://owasp.org/projects/asvs> (2026-09-23)

[S5] OWASP LLM01:2025 Prompt Injection. <https://genai.owasp.org/llmrisk/llm01-prompt-injection/> (2026-09-23)

[S6] Anthropic Claude Code: безопасность. <https://code.claude.com/docs/en/security> (2026-09-23)

[S7] Anthropic Claude Code: использование данных. <https://code.claude.com/docs/en/data-usage> (2026-09-23)

[S8] Anthropic Claude Code: аутентификация. <https://code.claude.com/docs/en/iam> (2026-09-23)

[S9] OWASP Web Security Testing Guide 4.2. <https://wstg.owasp.org/v4.2/> (2026-09-23)