

CommitScope 2.4.0 | Руководство пользователя

01 / Что считать полной проверкой

Поймите конечный результат до запуска команд

Корпоративная проверка завершена, когда сотрудник проверил точный чистый коммит Git по утвержденной внешней политике; Semgrep, Gitleaks, Trivy, Claude Hunter и независимый Claude Verifier отработали; неизменный защищенный каталог передан назначенному ревьюеру; человек выполнил verify-review и отдельно зафиксировал решение. Hunter и Verifier обязательны. Демо только со сканерами, scan, ai или GitHub Action дают лишь частичные материалы.

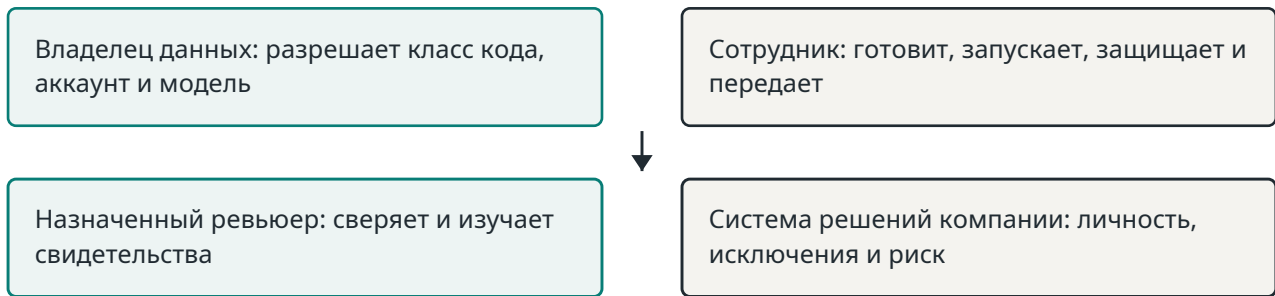


Рисунок 1. Роли и передача материалов; инструмент не принимает риск за компанию.

READY_FOR_HUMAN_REVIEW означает передачу ревьюеру, а не разрешение на слияние или доказательство безопасности приложения. Для реального репозитория нужны отдельное разрешение на передачу исходников и решение человека. В книге используется только синтетический пример.

02 / Установка и диагностика

Установите опубликованный тег на поддерживаемой машине

Цель: начать с проверенной версии v2.4.0, а не с изменяющейся ветки. CommitScope поддерживает macOS и Linux с glibc на x86_64/ARM64, Python 3.11-3.14; native Windows не поддерживается. Можно использовать WSL2 с поддерживаемым Linux. Нужны Git, pipx/Python, корневые сертификаты и разрешенный исходящий HTTPS. [G1]

```
pipx install \
  "git+https://github.com/akarazhev/commitscope.git@v2.4.0"
commitscope --version
commitscope preflight
commitscope bootstrap
commitscope doctor
```

Ожидаемо: версия 2.4.0, успешная проверка условий машины, установка закрепленных Semgrep/Gitleaks/Trivy в состояние CommitScope; doctor проверяет наличие и версии сканеров. Bootstrap скачивает исполняемые файлы и устанавливает зависимости Semgrep; БД Trivy загружается при scan/demo, а не bootstrap. Bootstrap не устанавливает и не запускает целевое приложение. Версии инструментов и БД отражаются в материалах запуска.

Если preflight отклоняет ОС, архитектуру или Python, перейдите на поддерживаемую машину. Если bootstrap или doctor не прошел, исправьте сеть, сертификаты, свободное место либо состояние сканеров и повторите диагностику. Не называйте неуспешный doctor полной проверкой и не пропускайте обязательный сканер.

03 / Аккаунт Claude Code и согласие

Используйте разрешенный личный или корпоративный аккаунт

Цель: будущие запросы Hunter и Verifier должны идти от разрешенного непривилегированного пользователя ОС и аккаунта. Установите официальный Claude Code 2.1.259 или новее под этим пользователем. Обертка ставит 2.1.278 при новой установке, но не обновляет имеющийся CLI; старую версию обновляйте по инструкции Anthropic. Поддержка платформ самим Claude Code шире, чем поддержка CommitScope. [G2]

```
claude --version
claude auth login
claude auth status
```

Ожидаемо: подходящая версия CLI и вход в аккаунт, разрешенный компанией. Уточните применимые условия обработки и хранения данных. У личного и коммерческого аккаунтов они различаются; личная подписка сама по себе не разрешает загружать исходники компании. Корпоративная проверка использует только --auth account, отклоняет переменные для API и иных провайдеров, без API fallback. [G3] [G4]

Если нет входа, квоты, доступа к модели или разрешения владельца, остановитесь до review. Повторно войдите нужным пользователем, устраните конфликтующие переменные, получите согласие владельца или выберите утвержденный точный ID модели. Не вставляйте токен в руководство и не переключайте провайдера незаметно.

04 / Учебный репозиторий

Создайте чистый учебный коммит с полным ID

Цель: потренироваться на включенном синтетическом примере IDOR, а не на коде компании. Команды предполагают, что /protected - утвержденный каталог владельца с правами 0700 вне целевого репозитория; замените его разрешенным абсолютным путем своей машины. Исходники v2.4.0 нужны только для учебного примера и политики. До разрешения владельца не используйте живой репозиторий.

```
git clone --branch v2.4.0 --depth 1 \
  https://github.com/akarazhev/commitscope.git \
  /protected/commitscope-training-source
install -d -m 700 /protected/training-target
install -d -m 700 /protected/reviews
cp /protected/commitscope-training-source/examples/ai-acceptance/idor/vulnerable/app.py \
  /protected/training-target/app.py
install -m 600 /protected/commitscope-training-source/examples/review-policy.json \
  /protected/review-policy.json

git -C /protected/training-target init -q
git -C /protected/training-target add app.py
git -C /protected/training-target \
  -c user.name=Training -c user.email=training@example.invalid \
  commit -m "Synthetic IDOR baseline"
git -C /protected/training-target status --short
git -C /protected/training-target rev-parse HEAD
```

Ожидаемо: пустой вывод status и один полный 40-символьный ID коммита в нижнем регистре. Проверьте и утвердите политику вне цели: область, угрозы, инвариант AUTH-001, порог и передачу кода. У учебного примера нет манифеста сторонних зависимостей, поэтому после проверки его live review использует явное объяснение --allow-empty-sca. Для реальной цели не скрывайте этим исключением отсутствующий lockfile.

Если status не пуст, зафиксируйте либо уберите все изменения, включая неотслеживаемые файлы. Политика с узнаваемыми формами секретов, например Bearer/Basic authorization values или URL с учетными данными, отклоняется до сканеров. Фильтр исходного пакета эвристический и не доказывает отсутствия неизвестных секретов. Исправьте исходник или политику и снова получите разрешение перед загрузкой.

05 / Тренировка и полный review

Различайте учебную проверку сканеров и полные материалы

Цель: различать упражнение со сканерами и полноценную локальную проверку. Сначала запустите demo без Claude в новом каталоге; это не корпоративная приемка. Результаты сканеров зависят от актуальных БД. Затем, только после явного разрешения владельца на передачу этого синтетического кода, выполните review по полному SHA из предыдущего шага и точному утвержденному ID модели.



Рисунок 2. Полный запуск: коммит, сканеры, независимые этапы модели, материалы и человек.

```
commitscope demo --out /protected/reviews/scanner-demo
```

Ожидаемо: DEMO_PASSED, если учебные проверки сканеров сошлись.

SCANNERS_VERIFIED_AI_NOT_RUN не означает завершённую корпоративную проверку. Если demo не прошёл, исследуйте версии инструментов и БД, не подгоняйте ожидаемый ответ. Результат demo отделен от каталога review.

```
commitscope review \
--repo /protected/training-target \
--ref 0123456789abcdef0123456789abcdef01234567 \
--policy /protected/review-policy.json \
--out /protected/reviews/run-id \
--auth account --allow-code-upload \
--model APPROVED_EXACT_MODEL_ID \
--allow-empty-sca "Synthetic stdlib-only fixture; reviewed imports and metadata"
```

Это шаблон команды, а не выполненный live run. Замените пример SHA реальным полным ID из g04, APPROVED_EXACT_MODEL_ID - утвержденным точным ID, run-id - новым именем. --allow-code-upload явно разрешает передачу отобранного пакета исходников Anthropic по условиям выбранного аккаунта. Чтение или сборка этого руководства не вызывает Claude. [G4]

Ожидаемо: одно из трех состояний g06 и защищенный каталог запуска. При ошибке входа, квоты, модели, тайм-ауте, неверном ответе или пропуске обязательного этапа состояние INCOMPLETE. Сохраните каталог, устраните причину и создайте новый запуск; не переиспользуйте путь вывода и не переходите на API-ключ.

06 / Состояния и находки

Преобразуйте код выхода в действие ревьюера



Рисунок 3. Материалы сохраняются в любом состоянии; только полные запуски разбирают по существу.

Таблица 1. Состояние, смысл материалов и действие

Код / состояние	Смысл	Действие
0 / READY_FOR_HUMAN_REVIEW	Этапы завершены; нет нормализованных находок на пороге политики	Ревьюер все равно изучает и решает
1 / FINDINGS_REQUIRE_TRIAGE	Этапы завершены; находки достигли порога	Ревьюер подтверждает, отклоняет или приоритизирует
2 / INCOMPLETE	Не выполнено условие, покрытие или согласованность	Сохранить материалы, исправить и повторить

Сначала прочтите верхнеуровневый report.md, затем report.json с нормализованными полями находок и report.sarif для совместимых инструментов. От каждой находки перейдите в evidence/ и к точному участку кода; изучите сильнейший контраргумент и пропуски. Сырой вывод сканеров,

пакеты исходников, конверты модели и диагностика в `private/` конфиденциальны. Учебная IDOR-находка - пример гипотезы, а не доказательство качества модели на реальном коде.

Иллюстративный синтетический пример, а не наблюдавшийся ответ модели: `app.py:9-11` отбрасывает `requesting_tenant` и возвращает `ORDERS[order_id]`. Инвариант `AUTH-001` разрешает вызывающему доступ только к данным своего арендатора. Проверяемое свидетельство: `tenant-a` может запросить `order-2`, принадлежащий `tenant-b`. Контраргумент: вышестоящий маршрут может проверять владельца, поэтому до подтверждения изучите реальную цепочку вызовов. Решение человека: назначенный ревьюер отдельно фиксирует подтверждение, отклонение или нехватку доказательств в защищенной системе решений.

Если для находки нет достижимого пути, нарушенного инварианта или убедительных материалов, запишите неопределенность; не удаляйте находку молча. Если нет покрытия, классифицируйте запуск как `INCOMPLETE`. `READY_FOR_HUMAN_REVIEW` никогда не разрешает слияние.

07 / Защита и передача

Передайте ревьюеру неизменный каталог

Корень: `report.md`, `report.json`, `report.sarif`

`evidence/`: нормализованные трассы и данные о покрытии

`private/`: сырые сканы, код, ответы модели, диагностика

Манифест и `reviewer-decision-template.json`: целостность и шаблон решения

Рисунок 4. Состав каталога запуска; все содержимое конфиденциально.

Цель: сохранить исходный каталог 0700 и файлы 0600, затем передать его без изменений назначенному ревьюеру по утвержденному защищенному каналу компании. Никогда не коммитьте и не публикуйте `private/`. Нормализованные отчеты и `evidence/` можно передавать лишь по правилам компании. Ревьюер должен получить и проверить весь исходный запуск, не только выдержку из Markdown.

```
commitscope verify-review --run /protected/reviews/run-id
```

Ожидаемо: `verify-review` сверяет структуру, типы файлов и права, хеши, коммит/снимок, идентичность политики и ресурсов, покрытие сканеров и Hunter/Verifier, согласованность решения. Для полного корректного запуска возвращается его класс 0 или 1; измененные, отсутствующие или неполные материалы дают 2. Ревьюер изучает свидетельства и ограничения, затем заполняет копию `reviewer-decision-template.json` в защищенной системе решений компании с проверенным хешем манифеста.

Манифест не подписан. verify-review устанавливает внутреннюю согласованность, но не авторство и не неизменяемость хранилища. Личность человека, согласование исключений, принятие риска и аудит хранения обеспечиваются внешними процессами. Если проверка не прошла, сохраните оригинал, выясните причину и получите новый неизмененный запуск; не правьте запечатанные материалы.

08 / Исправление и повтор

Новый коммит требует новой проверки

Цель: увидеть, как исправление меняет материалы без перезаписи истории. В обучении скопируйте исправленный app.py из примера IDOR в ту же синтетическую цель, закоммитьте и запишите новый полный SHA. В коде компании примените утвержденное исправление и функциональные/защитные тесты. Используйте новый каталог вывода и повторите сканеры, Hunter, Verifier, verify-review и разбор человеком.

```
cp /protected/commitscope-training-source/examples/ai-acceptance/idor/fixed/app.py \
/protected/training-target/app.py
git -C /protected/training-target add app.py
git -C /protected/training-target \
-c user.name=Training -c user.email=training@example.invalid \
commit -m "Synthetic ownership fix"
git -C /protected/training-target rev-parse HEAD
```

Ожидаемо: другой полный ID коммита. Повторите команду review из g05 с новым SHA и /protected/reviews/run-fixed в --out, затем проверьте и передайте каталог. Сохраните run-id и run-fixed без изменений. Сравните нормализованные находки, покрытие и политику; исчезновение находки само по себе не доказывает исправления. Ревьюер проверяет инвариант владения в разрешенных тестах и фиксирует новое решение.

Если новое дерево не чистое, зафиксируйте либо уберите изменения до review. Если запуск INCOMPLETE, исправьте причину и начните в еще одном новом каталоге. Если находка осталась или изменилась, разберите ее, а не считайте исправление успешным автоматически.

09 / Неполадки и справка

Ищите нарушенное условие, а не только код выхода

Таблица 2. Наблюдение, вероятная граница и действие

Наблюдение	Вероятная причина	Действие
preflight не прошел	ОС, Python, Git или путь	Поддерживаемая машина; точная диагностика
doctor не прошел	Нет закрепленного сканера или не та версия	Исправить bootstrap и повторить doctor
review отклонил policy	Секреты или неверная внешняя политика	Удалить данные; снова утвердить
review отклонил repo	Грязное дерево, короткий SHA или опасный путь	Очистить дерево; полный SHA и безопасные пути
Этап Claude INCOMPLETE	Вход, квота, модель, время или неверный ответ	Исправить аккаунт/этап; новый запуск
verify-review вернул 2	Материалы изменены, неполны или небезопасны	Сохранить оригинал; проверить передачу
Находка исчезла	Код изменился или детектор нестабилен	Проверить инвариант и новый точный коммит

```
commitscope preflight
commitscope bootstrap
commitscope doctor
commitscope demo --out /protected/reviews/scanner-demo
commitscope review --help
commitscope verify-review --run /protected/reviews/run-id
```

Код выхода 0 сам по себе не является решением о безопасности. GitHub Action и совместимые команды scan/ai дают сканерную либо частичную диагностику; для реального репозитория необходим назначенный ревьюер. Сверяйте руководство с установленным CLI v2.4.0 и периодически перепроверяйте поведение модели и сканеров. [G1] [G2] [G5]

Источники и дата проверки

[G1] Релиз CommitScope v2.4.0. <https://github.com/akarazhev/commitscope/releases/tag/v2.4.0> (2026-09-23)

[G2] Claude Code: установка и настройка. <https://code.claude.com/docs/en/setup> (2026-09-23)

[G3] Claude Code: аутентификация. <https://code.claude.com/docs/en/iam> (2026-09-23)

[G4] Claude Code: использование данных. <https://code.claude.com/docs/en/data-usage> (2026-09-23)

[G5] Claude Code: безопасность и недоверенные данные. <https://code.claude.com/docs/en/security> (2026-09-23)