

// раздатка к воркшопу

AI в работе **Security Engineer**: реальные способы **ЭКОНОМИТЬ ЧАСЫ** каждую неделю

// главные тезисы

- | | |
|---|---|
| <p>1 AI не заменяет инженера и это не «будущее через пять лет» — он уже сегодня даёт новичку в разы быстрее стать продуктивным.</p> <p>2 AI не делает вас экспертом — он позволяет учиться в несколько раз быстрее.</p> <p>3 Сегодня это персональный наставник, помощник и ускоритель: быстрее осваиваешь технологии, увереннее решаешь задачи, меньше рутины.</p> | <p>4 Полезен прямо сейчас: не через пять лет, не после AGI. И не нужно быть Senior — даже новичок применяет AI каждый день.</p> <p>5 AI неплохо пишет код и черновики — но проверяет всё равно инженер.</p> <p>6 Сильнее через несколько лет будут не те, кто игнорирует AI, а те, кто умеет работать вместе с ним.</p> |
|---|---|

// рабочий приём · задача → AI → проверка → доработка

- | | |
|---|---|
| <p>1 Задача — сформулируй проблему в 1–2 фразы</p> <p>2 Контекст — дай документ / код / пример и ограничения</p> <p>3 Черновик — получи первое решение: код, план, презентацию, объяснение</p> | <p>4 Проверка — факты, безопасность, корпоративный контекст, краевые случаи</p> <p>5 Доработка — доведи руками до готового и безопасного вида — и повтори</p> |
|---|---|

// безопасность работы с AI

Проверка — на тебе

// чек-лист безопасного использования AI · 10

- [] Сформулировал задачу в 1–2 фразы
- [] Дал модели контекст и ограничения
- [] Не вставил ничего чувствительного (секреты, PII, внутренние данные)
- [] Попросил перечислить допущения и краевые случаи
- [] Прочитал и понял каждую строку вывода
- [] Проверил факты по первоисточнику (CVE, флаги, команды)
- [] Учёл корпоративный контекст
- [] Проверил на небезопасные паттерны (TLS, shell, secrets, eval)
- [] Протестировал результат
- [] Понял решение достаточно, чтобы объяснить коллеге

// правила безопасного использования · 7

- 1 Не вставляй секреты, ключи, токены, PII, внутренние конфиги и клиентский код (кроме одобренных инструментов).
- 2 Любой AI-код — недоверенный ввод: прочитай, пойми, протестируй.
- 3 Проверь факты (CVE, команды, конфиги) по первоисточникам.
- 4 Помни про корпоративный контекст — AI его не знает.
- 5 Предполагай prompt injection, когда ввод контролирует кто-то извне.
- 6 Человек одобряет всё, что действует (агенты).
- 7 Учись на выводе — не вставляй бездумно, разбирайся.

// где AI подводит — и что делать · 5

- ! **Врёт и галлюцинирует.** Несуществующий CVE, флаг, синтаксис → сверяй факты с первоисточником.
- ! **Придумывает API.** Ссылается на функции и опции, которых нет → проверяй по документации.
- ! **Не знает корп-контекст.** Не в курсе ваших процессов и данных → добавляй контекст сам.
- ! **Зависит от промпта.** Слабый ввод — слабый вывод → формулируй задачу и ограничения.
- ! **Ограничения на задачах ИБ.** Осторожничает или ошибается в security-темах → нужна экспертиза инженера.

// с чего начать

Стартовые промпты и инструменты

// стартовые промпты · 5

Разбор документации / технологии

«Объясни этот раздел так, будто я новичок в [технология]. Дай mental model на 5 минут и назови 3 вещи, в которых я скорее всего ошибусь».

Черновик кода / скрипта

«Напиши небольшой, хорошо прокомментированный скрипт, который делает [X]. Приоритет — читаемость и безопасные значения по умолчанию. Перечисли допущения и необработанные краевые случаи».

Презентация из документа

«Собери из этого документа черновик презентации на [N] слайдов в деловом стиле. Ничего не выдумывай — только факты из текста. В конце перечисли, что осталось за кадром».

Проверка идеи / инструмента

«Я хочу сделать [инструмент]. Предложи архитектуру, найди слабые места, дай альтернативы и короткий roadmap».

Ревью безопасности

«Проверь этот код на проблемы безопасности: инъекции, небезопасные значения по умолчанию, секреты, валидация ввода. Будь конкретным, указывай строки».

// инструменты под задачу

ChatGPT / Claude

Объяснить документацию, брейншторм, черновой текст и код

когда: быстрый вопрос, разбор идеи

Cursor

Писать и править код с AI прямо в редакторе

когда: есть проект, код нужен рядом

Claude Code

Агент в терминале: читает репозиторий, идёт по шагам

когда: многошаговая задача с файлами

Claude Design

Быстрые визуалы и черновики слайдов

когда: презентации, схемы

// ничего не выдумывай — только факты из текста

// применяй с первого дня

Шаблон, понедельник и что дальше

// шаблон рабочего процесса

«задача → AI → проверка → доработка» — держи под рукой для любой новой задачи:

Задача: _____

Что даю модели (контекст): _____

Что НЕ даю (чувствительное): _____

- ☐ Черновик получен ☐ Проверен построчно / по фактам
- ☐ Корп-контекст учтён ☐ Доработан руками
- ☐ Понимаю решение и могу объяснить

// что сделать в понедельник · 4

- 1 Выбери 1–2 рутинные задачи недели: документация, отчёт, скрипт или презентация.
- 2 Проведи каждую через приём: задача → контекст → черновик от AI → проверка → доработка.
- 3 Засеки сэкономленное время — это и есть «часы каждую неделю».
- 4 Сохрани этот файл и применяй чек-лист с первого дня.

// применяй с первого дня

// что дальше · обучение CyberED

Что дальше — на обучении CyberED

Сегодня — база под контролем человека. Дальше эти вещи ускоряют ещё сильнее и разбираются на обучении CyberED:

- > **AI-агенты и YOLO-режимы:** агент действует сам, человек проверяет диффы.
- > **Мультиагентные системы и оркестрация:** рои агентов, где одни агенты управляют другими.
- > **Скилы и плагины,** расширяющие возможности инструментов.
- > **3 интерфейса работы с агентами:** IDE (Cursor, Codex, Antigravity, Windsurf), Plugin (GitHub Copilot), CLI (Claude Code, Cursor CLI, Codex CLI, Gemini CLI, OpenCode, Amp).
- > **Эволюция воркфлоу:** 2025 (микроменеджмент → план-выполнение-ревью-тест → SDD «доверяй, но проверяй») → 2026 (YOLO → Ralph Loops → мультиагентные рои).
- > **8 стадий эволюции программиста 2024–2026** (по Steve Yegge): от ChatGPT и автодополнения до «агенты оркестрируют агентов».
- > **Принцип «будь боссом»:** джуниору — учись и будь скептиком; сеньору — строй и получай удовольствие; всем — жди неоднозначных результатов, но в сумме это плюс.
- > **Не отвлекайся на хайп:** терминология (vibe coder / agentic coder / coding agents) ещё устаканивается.

Сильнее через несколько лет будут не те, кто игнорирует AI, а те, кто умеет работать вместе с ним. Скорость — от AI, безопасность и понимание — от тебя.