

SENTRA

ЧЕК-ЛИСТ ПО ТЕСТИРОВАНИЮ AI- СИСТЕМ

ШАГ 1. АУДИТ АРХИТЕКТУРЫ И МОДЕЛИРОВАНИЯ УГРОЗ

- Изучите архитектуру приложения, потоки данных и точки интеграции с LLM для выявления поверхности атаки.
- Проведите моделирование угроз с учетом специфики ИИ (определите технические слабости и сценарии потенциального злоупотребления системой).
- Убедитесь в наличии архитектурной изоляции доверенных системных инструкций от недоверенного внешнего ввода.

Инструменты / Фреймворки: Методология PASTA (Process for Attack Simulation and Threat Analysis), MITRE ATLAS, NIST AI RMF.

ШАГ 2. АУДИТ ЦЕПОЧЕК ПОСТАВОК ИИ

- Проанализируйте спецификации компонентов машинного обучения (ML-BOM) для выявления устаревших или уязвимых библиотек.
- Оцените безопасность сторонних базовых моделей и адаптеров (например, LoRA) на предмет внедрения бэкдоров или вредоносного кода перед их интеграцией.
- Проверьте подлинность моделей через криптографические подписи и хеши файлов.

Инструменты: OWASP CycloneDX + Syft (генерация SBOM), Modelscan (поиск уязвимостей сериализации моделей), Trivy / Gype / OWASP Dependency-Check (поиск CVE), Sigstore / cosign и проверка хешей файлов.

ШАГ 3. АУДИТ БАЗ ЗНАНИЙ И ВЕКТОРОВ

- Проанализируйте конвейеры сбора данных и векторные базы (в системах RAG) на устойчивость к отравлению данных (Data Poisoning).
- Проверьте наличие строгого контроля версий наборов данных для отслеживания любых манипуляций.
- Оцените механизмы управления доступом (permission-aware) к базам, чтобы исключить утечки конфиденциальной информации между разными группами пользователей.

Инструменты: Promptfoo, Garak, DVC (Data Version Control) для версионирования датасетов и отслеживания манипуляций.

ШАГ 4. ОЦЕНКА УЯЗВИМОСТИ К ПРЯМЫМ ПРОМПТ-ИНЪЕКЦИЯМ

- Проведите активное Red Team тестирование интерфейса модели с использованием статических и динамических наборов промптов, провоцирующих обход правил.
- Оцените устойчивость модели к атакам смены ролей, использованию кодировок (например, Base64), малоресурсных языков и многошаговым (multi-turn) атакам.

Инструменты: PyRIT (от Microsoft), Promptfoo, Garak, HarmBench, Prompt Fuzzer / ps-fuzz, HouYi.

ШАГ 5. ПРОВЕРКА НА НЕПРЯМЫЕ И МУЛЬТИМОДАЛЬНЫЕ ИНЪЕКЦИИ

- Протестируйте парсинг внешних данных: загрузите в систему документы или укажите URL-адреса, содержащие скрытые текстовые команды (белый текст, невидимые unicode-символы, HTML-комментарии и т.д.).
- Проведите оценку кросс-модальных атак, в том числе встраивая payload в EXIF-метаданные изображений или накладывая текст на тестируемые картинки.

Инструменты: ASCII Smuggler (для внедрения невидимого контента в промпты), Promptmap (для тестирования экземпляров ChatGPT).

ШАГ 6. АУДИТ СИСТЕМНЫХ ПРОМПТОВ НА УТЕЧКИ

- Проанализируйте содержимое всех системных инструкций приложения на наличие конфиденциальных данных, API-ключей, матриц доступа и логики.
- Убедитесь, что критичные проверки безопасности не делегируются самой LLM (через текст промпта), а контролируются внешним кодом.
- Выполните атаки извлечения (Extraction Attacks), пытаясь обойти защитные фильтры и заставить модель раскрыть свои инструкции.

Инструменты: PyRIT, Promptfoo, Garak.

ШАГ 7. ОЦЕНКА БЕЗОПАСНОСТИ ОБРАБОТКИ ВЫВОДА

- Проверьте, как конечные системы и браузеры обрабатывают текст или код, сгенерированный LLM.
- Убедитесь во внедрении строгой валидации и контекстно-зависимого экранирования вывода (например, SQL escaping, HTML-кодирование).
- Протестируйте инфраструктуру на наличие традиционных уязвимостей XSS, SSRF, SQL-инъекций или удаленного выполнения кода (RCE), вызванных ответами ИИ.

Инструменты: Традиционные сканеры SAST/DAST/IAST для проверки конвейеров веб-приложения, Deerp-rwning, методология OWASP ASVS.

ШАГ 8. АУДИТ ИЗБЫТОЧНОЙ АГЕНТНОСТИ И ИНСТРУМЕНТОВ

- Оцените доступы, выданные AI-агенту (плагины, протокол MCP), на строгое соблюдение принципа наименьших привилегий (ограничение функционала и разрешений).
- Убедитесь во внедрении механизма «Human-in-the-loop» для ручного подтверждения опасных и высокорисковых действий.
- Протестируйте устойчивость агента к «отравлению инструментов» (Tool Poisoning) — подмене текстовых описаний функций на сервере.

Инструменты: GOAT (Generative Offensive Agent Tester), Llamator.

ШАГ 9. АУДИТ КОНТРОЛЯ ПОТРЕБЛЕНИЯ РЕСУРСОВ

- Проверьте наличие жестких лимитов запросов (Rate Limiting), квот токенов на пользователя и тайм-аутов.
- Убедитесь, что API скрывает детальную информацию о вероятностях (logit_bias, logprobs) для защиты от кражи архитектуры.
- Оцените устойчивость системы к атакам типа Denial of Wallet (DoS через истощение финансового баланса) при помощи отправки нестандартных или чрезмерно объемных запросов.

Инструменты: LLM Canary, традиционные инструменты нагрузочного тестирования API.

ШАГ 10. АУДИТ СРЕДЫ ВЫПОЛНЕНИЯ И МОНИТОРИНГА

- Изучите качество и полноту логирования взаимодействий с LLM (трассировка сложных цепочек вызовов агентов и RAG-запросов).
- Проверьте конфигурацию алертов на аномалии: всплески активности токенов, генерация токсичного контента, аномально длинные сессии, многократное использование малоресурсных языков.

Инструменты: SplxAI, Lakera Guard, Langfuse, phospho (платформы для мониторинга и фильтрации трафика LLM-приложений).