Газизова Светлана
Positive Technologies

Остаётся ли в тайне ваш диалог с LLM?

Как использовать большие языковые модели безопасно!

Что нас ждет завтра?

LLM — это инструмент бизнеса, но без должной защиты он превращается в **лазейку для хакеров, утечек данных и финансовых катастроф**. Не сливает ли ваша LLM по запросу клиентские данные и не генерирует ли вредоносный код? Такие риски уже **стоят компаниям миллионов** — от штрафов регуляторов до потери доверия клиентов.

Безопасность LLM — это не просто «галочка», а **стратегическая инвестиция**.

15%

Рост затрат на AI Sec

Из-за рисков, которые несет в себе использование Gen AI

Gartner, Cool Vendors 2024 for AI Security

55%

Сотрудников

Используют в работе несанкционированные GenAI, создавая дополнительные риски

Salesforce, Generative AI Adopters Use Survey

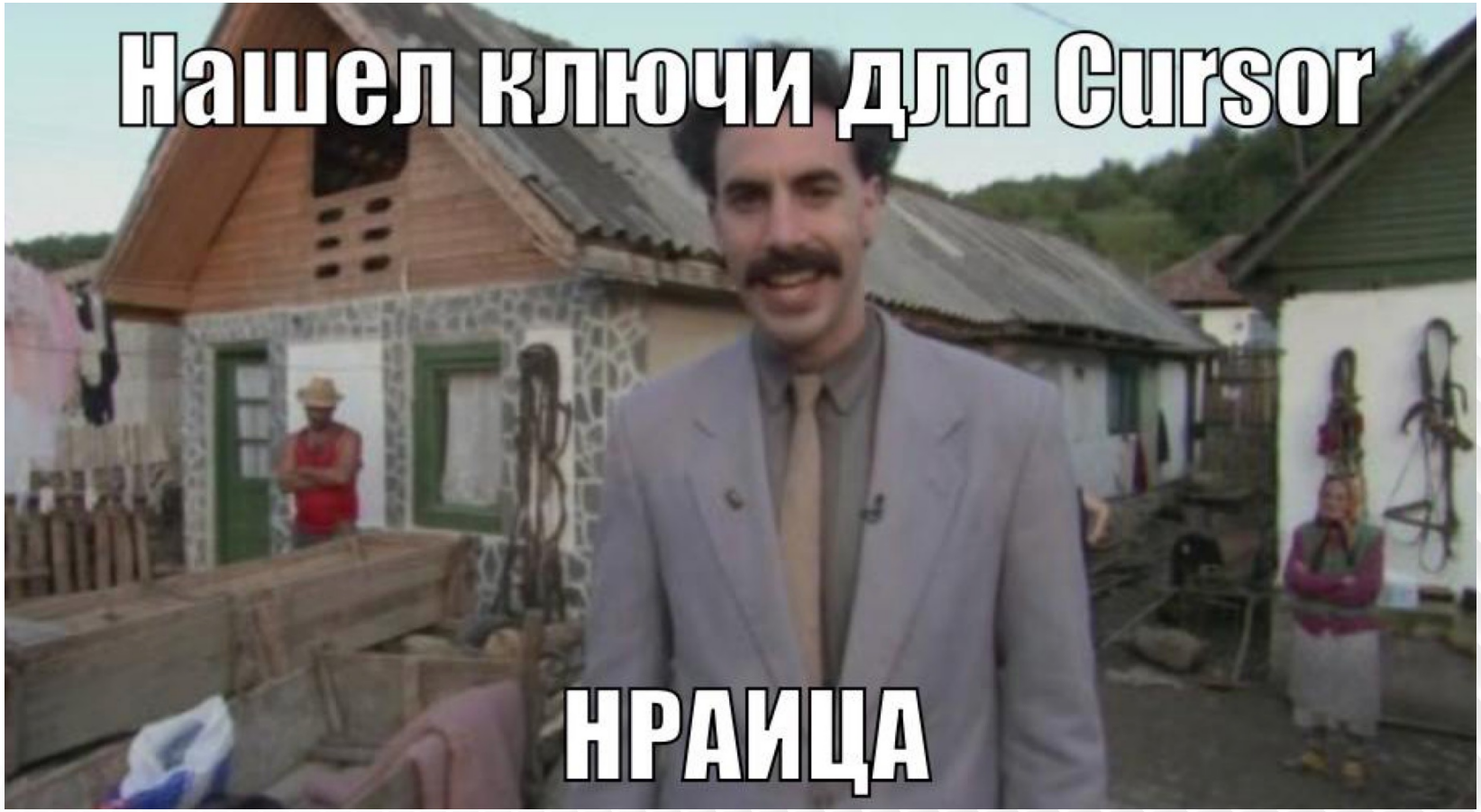
62%

Атак на AI

Совершаются, в большинстве своем, внутренними сотрудниками

Gartner, AI in Enterprise Survey

LLM для разработчиков...💁‍♀️



Кейс 1

Задача: **Генерация кода**

Сценарий: Разработчики используют LLM (GitHub Copilot, ChatGPT, Llama, Antropic и тд) для автоматического написания кода, рефакторинга или поиска уязвимостей.

Риски:

1. Модель предлагает небезопасный код (содержащий инъекции, захардкоженные пароли и тд)
2. Модель предлагает код из публичных репозиториев конкурентов

Меры митигации: Воспринимать любой код критически! Обязательная проверка с помощью SAST и запрет на генерацию кода значимых функций.

Кейс 2

Задача: **Создание схем API**

Сценарий: Модель генерирует swagger документацию и примеры запросов к API

Риски:

1. Раскрытие эндпоинтов в случае, когда модель включает в спецификацию доступную всем закрытые API
2. Создание некорректной документации

Меры митигации: Проверка документации при любом создании и изменении. Запрет использования LLM для создания схем для внутренних API

Кейс 3

Задача: **Создание автотестов**

Сценарий: Модель генерирует автотесты в плане функционального или нагрузочного тестирования, а также создает сценарии тестирования на проникновение

Риски:

1. Модель создает use-кейсы, не покрывающие весь ландшафт тестирования - тесты некорректны.
2. Создание вредоносного контента (вместо тестов – эксплоиты).

Меры митигации: Ручное ревью тестов перед запуском, безопасное окружение для сгенерированных тестов.

Кейс 4

Задача: **Анализ логов и мониторинг**

Сценарий: LLM обрабатывает логи продуктов ИБ или ИТ для поиска отклонений и аномалий.

Риски:

1. Обход детекции – злоумышленник будет маскировать атаку под привычный траффик.
2. Утечка логов и правил корреляции.

Меры митигации: Ограничение доступа модели к сырым логам. Шифрование запросов и ответов.

Кейс 5

Задача: **Генерация конфигов**

Сценарий: LLM помогает создавать файлы конфигурации, манифесты развертывания, YAML-манифесты

Риски:

1. Модель предлагает конфигурации, в которых содержатся избыточные права по умолчанию.
2. В рекомендациях модели могут быть образы с публично известными уязвимостями.

Меры митигации: Применение политики минимальных привилегий для автоматически генерируемых конфигов. Проверка образов через инструменты класса container security.

Кейс 6

Задача: **Оптимизация запросов**

Сценарий: Разработчики отправляют в LLM SQL-запросы для их оптимизации и дальнейшего использования

Риски:

1. Инъекции – когда модель может добавить уязвимый синтаксис в запрос.
2. Утечка схемы БД, когда разработчик отправляем те поля, которые содержат чувствительную информацию.

Меры митигации: Использование ORM вместо прямых запросов. Маскирование метаданных в запросах к модели.

Кейс 7

Задача: **Чат-бот**

Сценарий: Разработчики используют чат-боты для настройки devops контура (если нет выделенного специалиста)

Риски:

1. Уязвимые конструкции в ответах на запросы по настройке (rm-rf).
2. Избыточный доступ к CI/CD может быть предоставлен LLM.

Меры митигации: Контектный контроль ответов (запрет sudo, например). Верификация пользователей через SSO.

Спасибо, девелопер!

Риск	Описание	Особенности
Неконтролируемый поток данных и риски утечек	Риск случайной или преднамеренной утечки этой информации через запросы пользователей или ответы модели	Сотрудник копирует фрагмент конфиденциальной информации/ модель в ответе выдает внутреннюю финансовую информацию
Внедрение вредоносного контента и атак	Злоумышленники могут использовать LLM для генерации фишинговых писем нового уровня, создания убедительных сценариев социальной инженерии или даже для автоматизированного поиска уязвимостей в системах	Сами сотрудники или сервисы компании могут стать жертвами атак, сгенерированных моделью или через запросы к LLM
Репутационные риски	Некорректные, предвзятые, дискриминационные или даже оскорбительные ответы могут нанести серьезный удар по репутации компании и привести к юридическим разбирательствам.	Модель может сгенерировать клеветнический отзыв или нарушит авторские права
Сложность мониторинга и аудита	Невозможность отслеживать все взаимодействия с LLM в масштабах всей компании	Требуется проводить аудит взаимодействий, чтобы выявить потенциальные нарушения политик безопасности или инциденты. Требуется наличие инструмента для анализа лингвистических данных
Отсутствие гранулярного контроля	Невозможно гибко контролировать доступ к LLM	Разные департаменты и сотрудники используют LLM для разных целей. Нужно иметь возможность устанавливать гибкие политики безопасности, учитывающие контекст использования. Например, для отдела маркетинга могут быть одни ограничения на генерацию контента, а для отдела разработки — другие ограничения на анализ кода.
Производительность и масштабируемость	Любые решения в области безопасности не должны существенно замедлять работу LLM или создавать узкие места в нашей инфраструктуре	Задержки в обработке запросов могут снизить эффективность использования LLM и вызвать недовольство пользователей. Кроме того, решение должно масштабироваться вместе с ростом использования LLM

Правило Парето?

Организационные меры

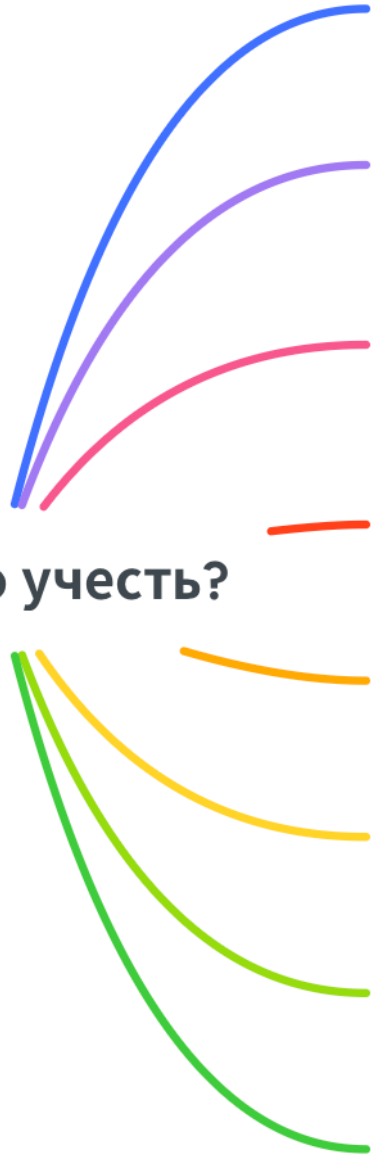
VS

Технические меры

«Бумажный» минимум

- Политика по безопасности ИИ ✓
- Внесение корректировки в политику по разработке ✓
- Внесение изменений в политику по информационной безопасности ✓
- Политика осведомленности сотрудников по использованию сторонних сервисов LLM ✓
- Регламент по использованию LLM ✓
- Политика обработки данных при работе с LLM ✓
- Инструкция для SOC по реагированию на инциденты, связанные с использованием LLM ✓

Что надо учесть?



- Управление доступом
- Защита данных
- Фильтрация входных параметров
- Безопасность RAG
- Безопасность архитектуры
- Повышение осведомленности
- Комплаенс и регуляторика
- Мониторинг и реагирование


positive
technologies
LLM Security
Чек-лист по безопасности больших языковых моделей


ПОЛИТИКА
БЕЗОПАСНОСТИ LLM
Шаблон

Доступные инструменты

🛠️ Инструменты для сканирования

Инструмент	Описание	Звезды
🔧 Garak	Сканер уязвимостей LLM	Stars 4.5k
🔧 ps-fuzz 2	Сделайте ваши приложения GenAI безопасными и защищенными 🚀 Тестируйте и укрепляйте системный промпт	Stars 483
🗺️ LLMmap	Инструмент для картирования уязвимостей LLM	Stars 39
🛡️ Agentic Security	Набор инструментов безопасности для AI-агентов	Stars 1.4k
🧠 Mindgard CLI	Интерфейс командной строки для инструментов безопасности Mindgard	Stars 26
🔒 LLM Confidentiality	Инструмент для обеспечения конфиденциальности в LLM	Stars 37
🔒 PyRIT	Инструмент идентификации рисков Python для генеративного ИИ (PyRIT) - это фреймворк автоматизации с открытым доступом, позволяющий специалистам по безопасности и инженерам машинного обучения проактивно находить риски в своих системах генеративного ИИ.	Stars 2.5k

Заходим сюда

Лучше сделать хоть что-то, чем не сделать ничего