ПРАКТИКУМ №1 СТРОИМ КАРТУ НАШЕГО "ОБЛАЧНОГО ФАЙЛОВОГО ХРАНИЛИЩА"

- **Шаг 1: Определяем нашу систему**
- **Шаг 2: Определяем акторов - Кто взаимодействует с системой?**
- **Шаг 3: Определяем внешние системы - С чем мы интегрируемся?**
- **Шаг 4: Определяем взаимодействия - Что передается через границы доверия?**
- **Подводим итог**

ОПРОСНИК ДЛЯ КОНТЕКСТА

- Кто пользователи? Доверяем ли мы ему?
- Какие внешние системы? Как мы аутентифируемся?

ОПРОСНИК ДЛЯ КОНТЕЙНЕРОВ

- **Протоколы? Как они общаются?**
- **Где данные?** Где сама БД? Что на счёт кэша?

ОПРОСНИК ДЛЯ КОМПОНЕНТОВ

- **Где валидация? Где проверка, что это не исполняемый файл?**
- **Что в логах?**

ПРАКТИКУМ №2 'ИЩЕМ УЯЗВИМОСТИ' (АНАЛИЗ УГРОЗ)

Для КОНТЕКСТА (внешние взаимодействия):

- 'Как пользователь доказывает, что это именно он? Достаточно ли только пароля?'
- 'Как мы защищаем данные, когда они передаются между системой и мобильным приложением?'
- 'Что произойдет, если злоумышленник перехватит ключ API к нашему Email-сервису?'

Для КОНТЕЙНЕРОВ (внутренние компоненты):

- 'Где хранятся файлы пользователей? Кто имеет к ним доступ?'
- 'Как общаются между собой веб-сервер и база данных? Зашифрован ли этот канал?'
- 'Где хранятся пароли пользователей? Как они защищены?'

ПРОБЛЕМЫ НАЙДЕНЫ. КАК ИХ РЕШИТЬ?

- Принцип наименьших привилегий
- Глубина защиты (Defense in Depth)
- Zero Trust ("Не доверяй, проверяй")

ПАТТЕРН "API GATEWAY"

- Было: Схема, где клиент напрямую обращается к разным сервисам.
- Стало: Схема, где клиент обращается к единому API Gateway, который перенаправляет запросы к сервисам.
- Выгода: Единая точка для: Аутентификации, Авторизации, Брандмауэра WAF, Логирования.

ПАТТЕРН "SERVICE MESH & MTLS"

- Было: Схема связи между сервисами с подписью "HTTP (открытый текст)".
- Стало: Схема, где между каждыми двумя сервисами находится Sidecar-прокси, а связь подписана "mTLS (зашифровано)".
- Выгода: Сквозное шифрование, контроль доступа "сервис-сервис".

ПАТТЕРН "SECRETS MANAGEMENT" (VAULT)

- Проблема: Иконка ключа в конфиг-файле с красным крестом.
- Решение: Иконка сервиса (HashiCorp Vault), от которого ключи "выдаются по запросу" сервисам.
- Выгода: Никаких хардкод-секретов, централизованный аудит и ротация.

ПРИНЦИП "СЕТЕВОЙ СЕГМЕНТАЦИИ"

- Было: Одна общая сеть, где веб-сервер, БД и кэш "видят" друг друга.
- Стало: Схема с тремя "полочками" (сетями): Публичная, Служебная (DMZ), Приватная. БД находится в Приватной сети.
- Выгода: Изоляция критичных компонентов.

ПРАКТИКУМ №3 МОДЕРНИЗИРУЕМ НАШЕ "ФАЙЛОВОЕ ХРАНИЛИЩЕ"

- Задача: "Используя изученные паттерны, предложите архитектурные изменения для устранения уязвимостей".

КЛЮЧЕВЫЕ ВЫВОДЫ

1. C4 — это карта. Без нее анализ безопасности вслепую неэффективен.
2. Безопасность проектируется, а не добавляется. Заложите ее в основу.
3. Используйте паттерны. Не изобретайте велосипед для стандартных угроз.