Марченко Глеб
TG: @keepitawesome

Данные vs пользователи

отдать нельзя закрыть

О чем мы говорим, когда говорим о безопасности данных?

Пусть наша North Star - это 0 потерянных рублей на инцидентах с данными:

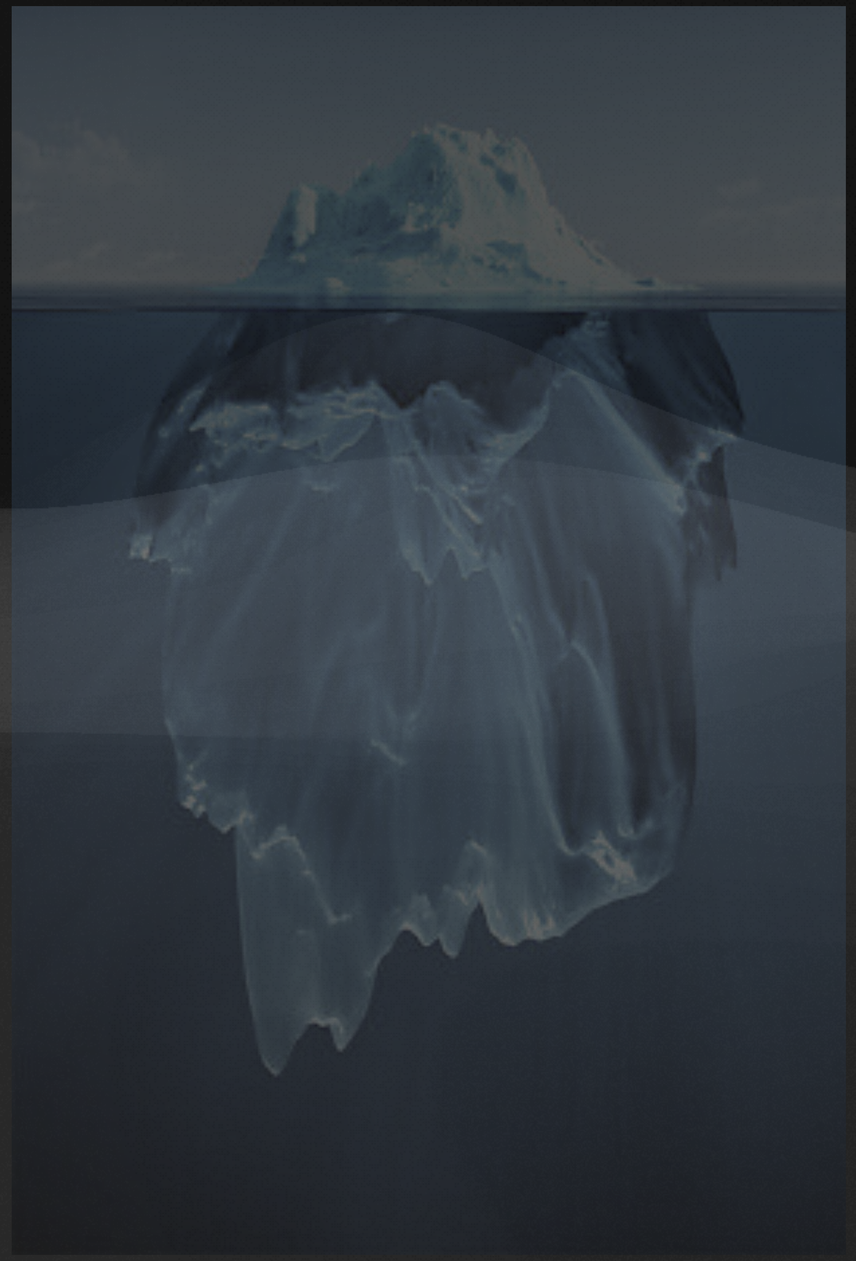
- Утечки
- Штрафы
- Утрата
- Отравление
- Фрод
- Недополученная прибыль etc.

В чем проблема такого подхода?

Данные должны работать и приносить прибыль - а для этого их нужно давать людям

От кого мы ждем угрозы?

- Хакеры
  - Примерно 77%
- Инсайдеры
  - Примерно 23%
- Регуляторы
  - Неизмеримо ((



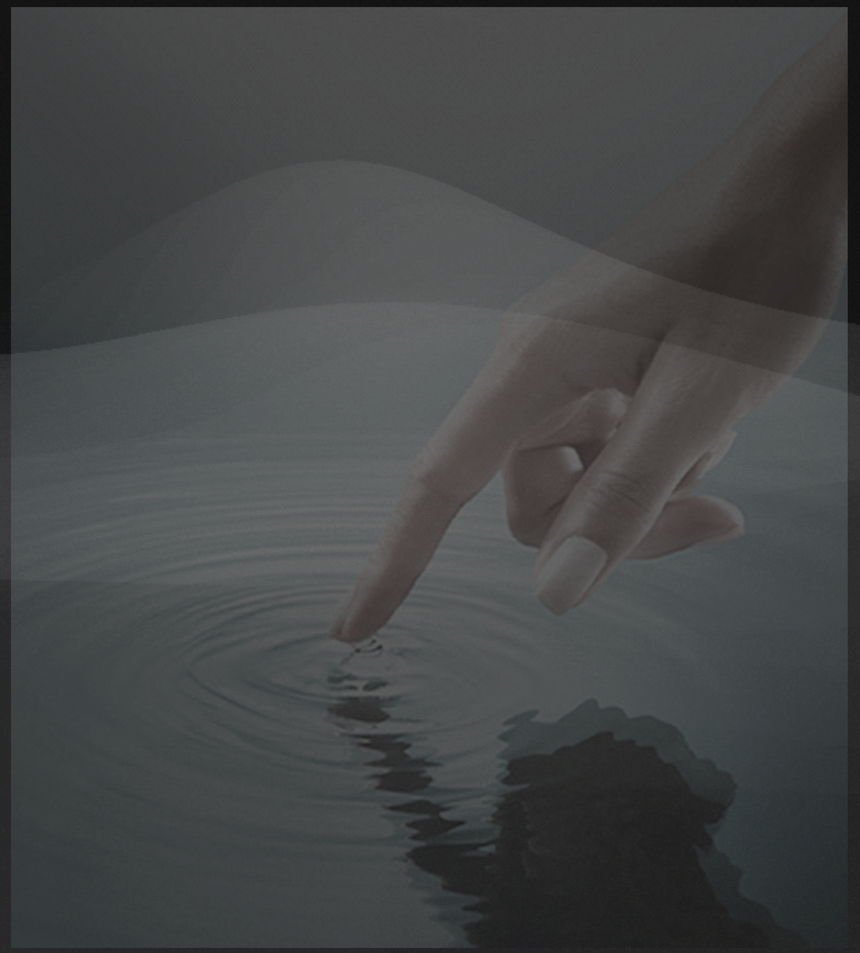
Zero Touch Data

- это подход к проектированию систем, при котором конфиденциальность данных обеспечивается по умолчанию

Касание - это единичный доступ[1], который дает возможность выгрузить набор[2] чувствительных[3] данных в небезопасную[4] среду.

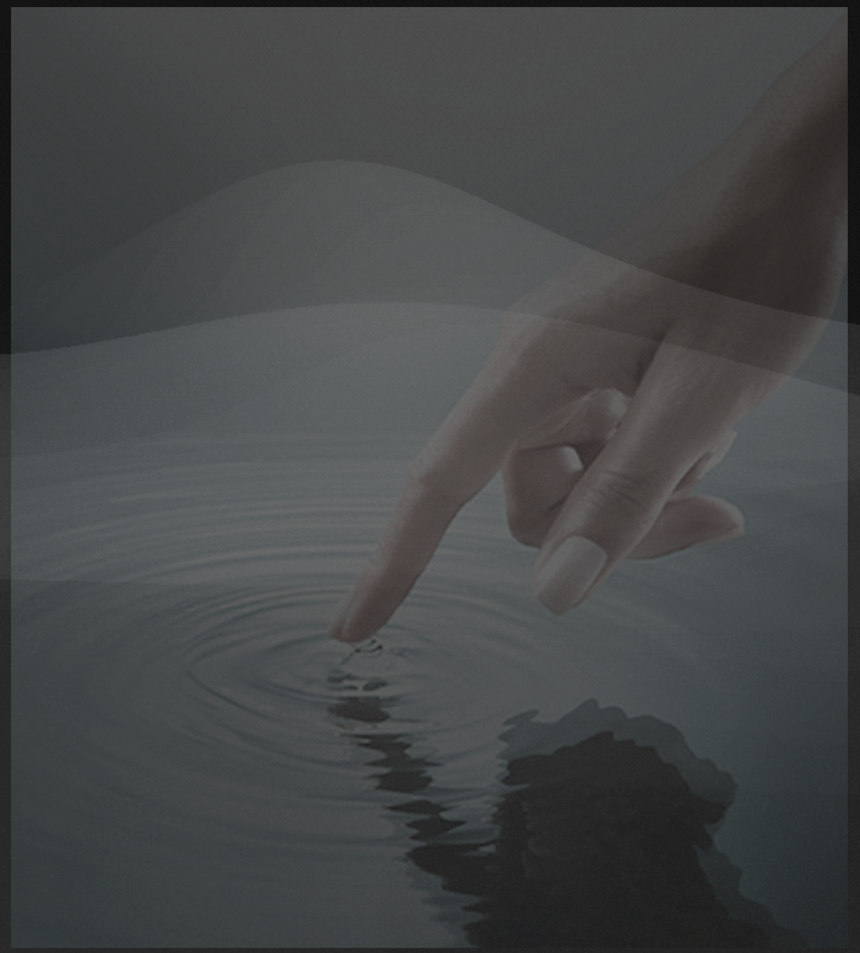
Какие бывают касания?

- Доступы через приложения
  - Таблицы, дашборды и витрины
  - Сервисы для работы с данными клиентов
  - Избыточное раскрытие данных
  - Платформы наблюдаемости и т.п.

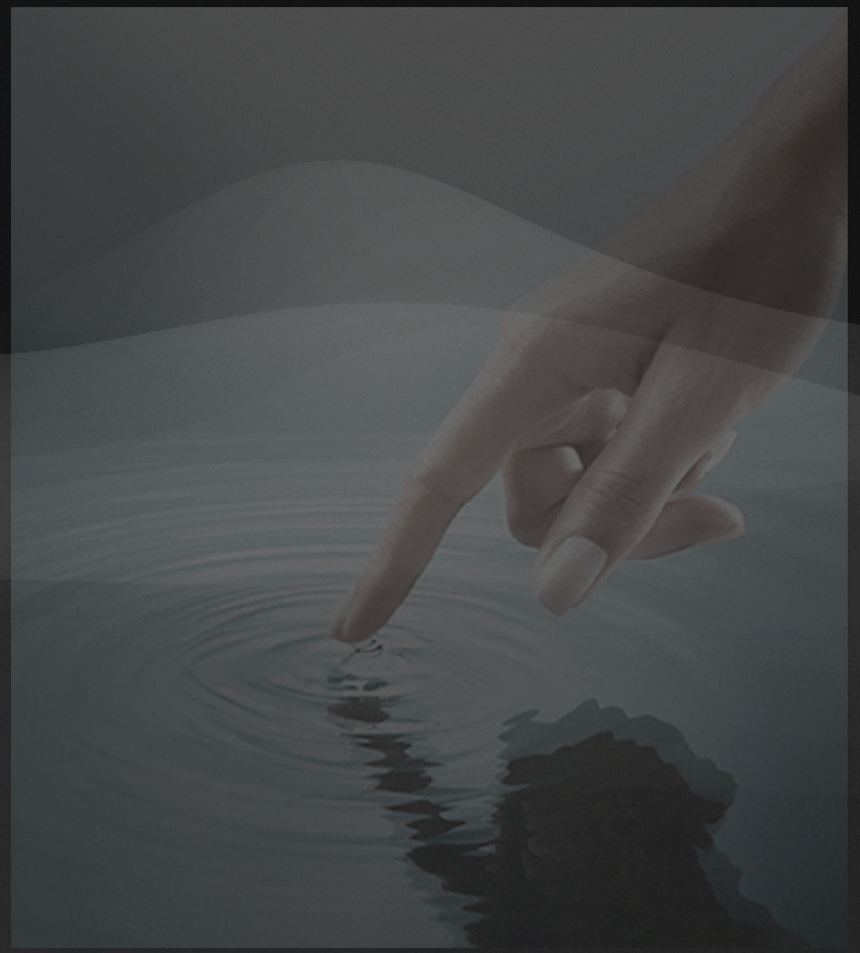


Какие бывают касания?

- Крит в местах общего пользования
  - Таск-трекеры и базы знаний
  - Файлообменники и шары
  - Репозитории
  - Подставь свое



Какие бывают касания?



- Прямой доступ к хранилищам и транзитным узлам
  - Продуктивные БД
  - Возможности перекладки из прода в тест
  - Самоуправляемые хосты, свободный доступ к проду
  - Бэкапы и слепки
  - Кэши, транзитные хосты и базы данных

Как убрать касание?

Или хотя бы сделать его контролируемым

Забрать доступ:

- Пересмотр и отзыв лишнего
- Минимальные привилегии и контроль PPM
- Доступ только по необходимости, короткий TTL и эфемерные секреты

Как убрать касание?

или хотя бы сделать его контролируемым

Запрет выгрузки набором:

- Контроль на уровне приложения
- Мониторинг и алертинг
- Контроль трафика
- Контроль ответов сервера, API etc.

Как убрать касание?

или хотя бы сделать его контролируемым

Сделать данные нечувствительными:

- Анонимизация и деперсонализация
- Шифрование
- Маскирование, дифференциальная приватность, whatever

Как убрать касание?

или хотя бы сделать его контролируемым

Сделать среду безопасной:

- Сетевые ограничения
- Харденинг эндпойнтов
- Контроль окружения

Немного математики

как понять, что становится лучше?

$$\text{Общий риск} = P \cdot ((1 - (1 - R)^A) \cdot L_1 + (A \cdot R - 1) \cdot L_n)$$

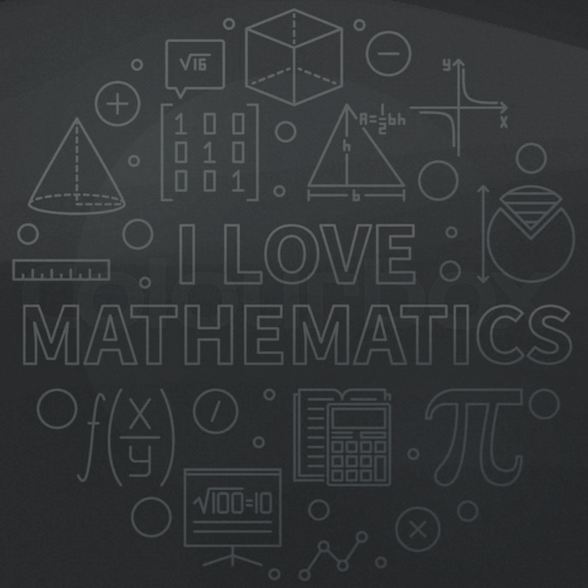
где

P - размер популяции

R - процент нарушителей

A - количество касаний (активных доступов)

L - размер финансовых потерь за 1-ю и последующие утечки



В сущности, важна лишь вероятность первой утечки:

- на 10 касаний вероятность утечки 40%
- на 100 касаний - 99%
- на 1000 касаний - практически полные 100%

Почему вероятность так велика,

а утечки происходят не раз в минуту?

Есть сопутствующие факторы и меры митигации:

- Сложность совершения касания
- Неотвратимость наказания (или иллюзия неотвратимости)
- Влияние корпоративной культуры
- Отношение выгоды к рискам

Как это коррелирует с безопасностью данных в СУБД?

примеры из практики и просто выводы

Управляемые доступы к БД

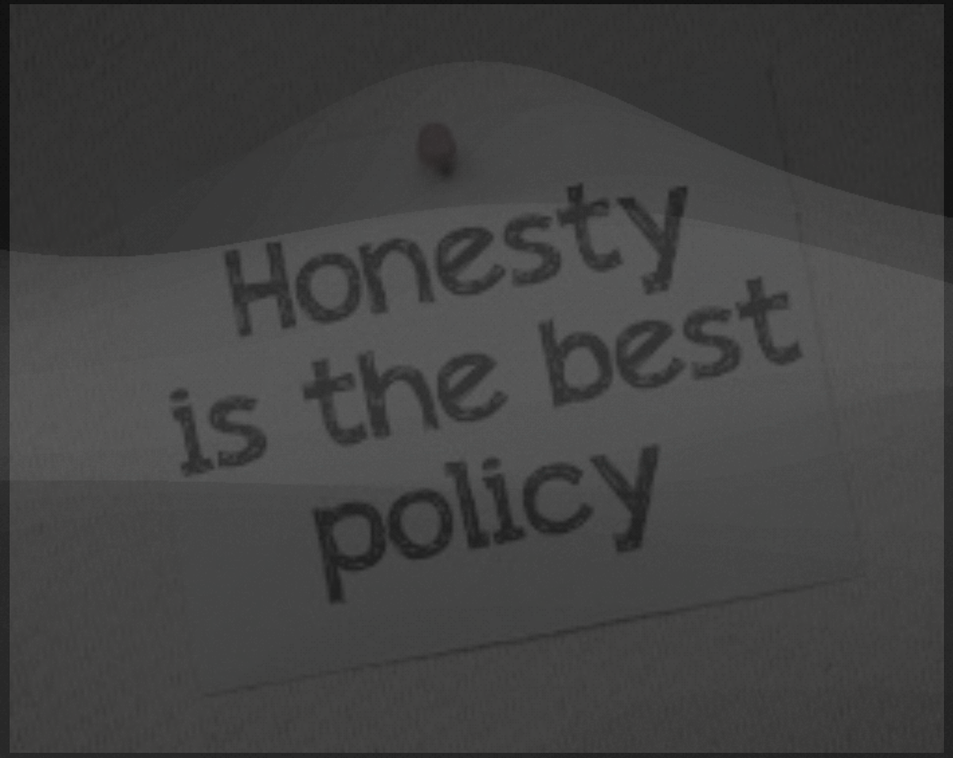
- Эфемерные УЗ в платформе баз данных
- В неплатформенных БД - доступы по тикетам с TTL
- Доступ только из защищенной среды
- Минимальное количество «живых» доступов в моменте

Как это коррелирует с безопасностью данных в СУБД?

примеры из практики и просто выводы

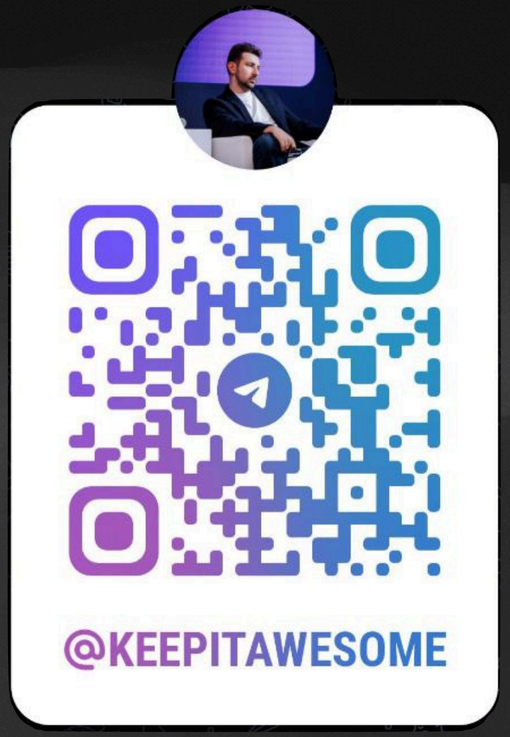
Где еще нужно по-честному считать касания?

- Тестовый контур (особенно с доступом извне)
- Базы с выходом на периметр
- Хранилища бэкапов и слепков
- Доступ к базе с сервера приложения (кто и как может туда прийти)
- Варианты?



Спасибо за внимание!

Восторги, донаты и благодарности сюда:



(недовольство тоже сюда, но это лишнее)