Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Инструменты для хранения и обработки больших данных

**Лабораторная работа №3.1.**

**Тема:**

**«**Проектирование архитектуры хранилища больших

данных**»**

Выполнила: Сергеева А. И., группа: АДЭУ-211

Преподаватель: Босенко Т. М.

Москва

2024

**Задача:** создать архитектуру хранилища больших данных для компании, занимающейся анализом потребительского поведения.

**Цель работы:** обеспечить надежное хранение, эффективную обработку и анализ больших объемов данных, получаемых из различных источников, таких как веб-сайты, мобильные приложения, социальные сети, системы CRM.

**Ход работы**

**1. Требования к данным для средней компании в сфере кибербезопасности.**

**1.1 Объем данных.**

- Ожидаемый объем: 200 ТБ в год.

- Рост: 60% ежегодно.

**1.2 Скорость получения данных.**

- до 10000 событий в секунду.

**1.3 Типы данных.**

- Структурированные: конфигурационные данные, организованные в БД (30%).

- Полуструктурированные: лог-файлы веб-сервера, данные JSON/XML (50%).

- Неструктурированные: текстовые, изображения с информацией о подозрительной активности, аудиофайлы разговоров (20%).

**1.4 Требования к обработке.**

- Анализ угроз в реальном времени.

- Выявление аномалий.

**1.5 Доступность данных.**

- Время отклика: 99.999%, время отклика <1 секунды.

**1.6 Безопасность данных.**

- Многоуровневое шифрование.

- Строгое соответствие 152-ФЗ и международным стандартам безопасности.

**2. Архитектура хранилища больших данных.**

**2.1 Компоненты архитектуры.**

**Источники данных:**

- Веб-сайты и мобильные приложения.

- Социальные сети (например, сетевой трафик).

- Информация о ПО с открытым исходным кодом.

- Отчеты о безопасности.

- Открытые данные об атаках, хакерских угрозах.

- Записи о действиях пользователей.

- Recorded Future's Vulnerability Database (БД уязвимостей).

**Слой сбора данных:**

- SecurityTrails (данные о DNS, WHOIS, истории доменов и IP-адресов) для анализа сетевых угроз.

- Logstash для преобразования данных в желаемый формат, можно внедрять свои плагины в рамках безопасности (горизонтально масштабируемый конвейер обработки данных с поддержкой Elasticsearch и Kibana.

- Cloudera DataFlow для потоковых данных.

**Слой хранения данных:**

- HDFS (Hadoop Distributed File System) для хранения больших неструктурированных и полуструктурированных данных в рамках локального хранилища, можно хранить данные, генерируемые в ходе операций по кибербезопасности (лог-файлы, сетевые данные) (их в сумме 80%).

- PostgreSQL для структурированных данных, например, данных о пользователях, их всего 20%.

**Слой обработки данных:**

- Apache Storm для потоковых данных (низкая задержка менее 1 секунды, интеграция с Hadoop, встроенные обработчики позволяют выполнять различные задачи над событиями безопасности — фильтрацию, нормализацию, разбор (парсинг), обогащение сведениями об угрозах).

- Apache Metron (масштабируемая платформа для расширенной аналитики безопасности в режиме реального времени, развитие платформы при изменении условий кибербезопасности сообществом Hadoop).

**Слой аналитики и машинного обучения:**

- TensorFlow, PyTorch, Keras, Scikit-learn для моделей машинного обучения.

- CTRLHACK (платформа симуляции кибератак для анализа уязвимостей и улучшения моделей обучения, российская разработка).

- Jupyter Notebooks для интерактивной аналитики.

- Apache Superset для визуализации и дашбордов.

**Слой управления данными:**

- Apache Atlas для управления метаданными (обеспечение соблюдения политик, определение ограничений доступа на основе ролей).

**Слой оркестрации и мониторинга:**

- Ansible для оркестрации рабочих процессов, для проверки соответствия системы политикам организации и нормативным требованиям.

- Git для управления конфигурациями.

- Grafana для визуализации данных мониторинга.

**3. Схемы «звезда» и «снежинка» для базового представления хранения данных и для процессов компании в сфере кибербезопасности.**

Схема «звезда» на рисунке 1 представляет вариант хранения данных о киберугрозах пользователей.

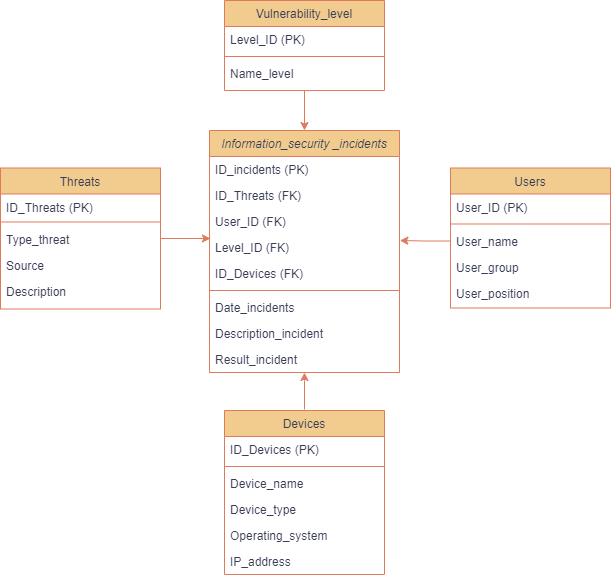


Рисунок 1 – Схема «Звезда» для данных по киберугрозам

Схема «снежинка» на рисунке 2 дополняет представление о хранении данных о нарушении безопасности.

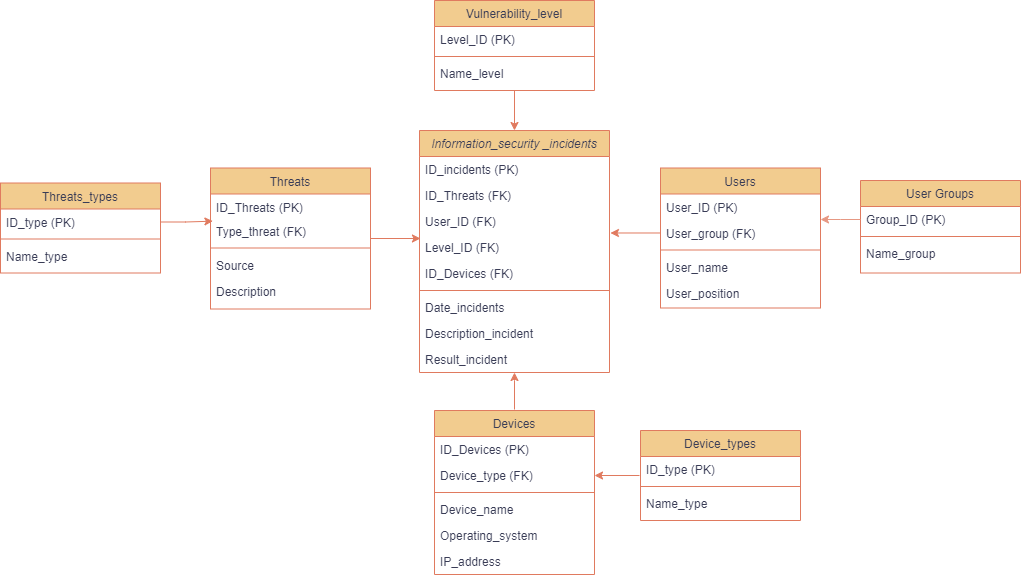


Рисунок 2 – Схема «Снежинка» для данных по киберугрозам

Схема «Снежинка» доказала большую эффективность, т. к. сокращает избыточность данных, для ее создания были внедрены дополнительные данные о типах угроз, пользователей, устройств. Также она обеспечивает целостность данных, которые важны в сфере кибербезопасности.

**4. OLAP и OLTP.**

Для быстрого анализа данных можно создать OLAP-кубы с детализацией по типу устройства, угрозы, времени, региону, типу события, которые представлены на рисунке 3.

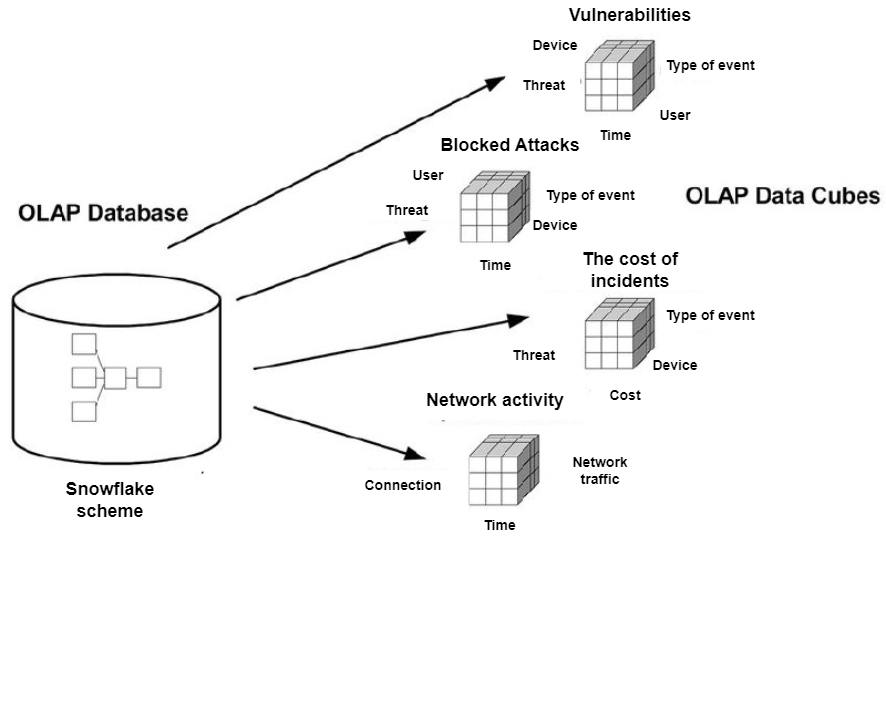


Рисунок 3- OLAP кубы для компании по кибербезопасности

В рамках OLTP источниками являются данные по аутентификации (например, попытки входа), сетевой трафик, ошибки или перезагрузки системы. Далее данные (логи безопасности) извлекаются, обрабатываются и загружаются в OLAP-кубы, которые были представлены выше. Ознакомиться с общей схемой можно на рисунке 4.

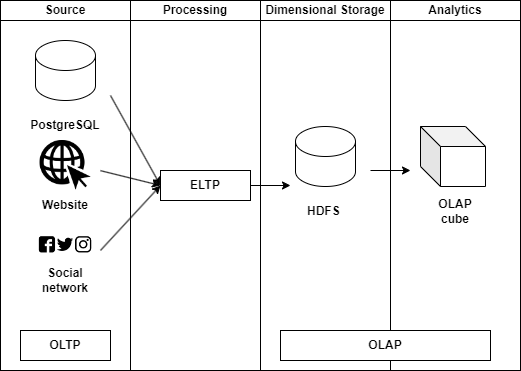
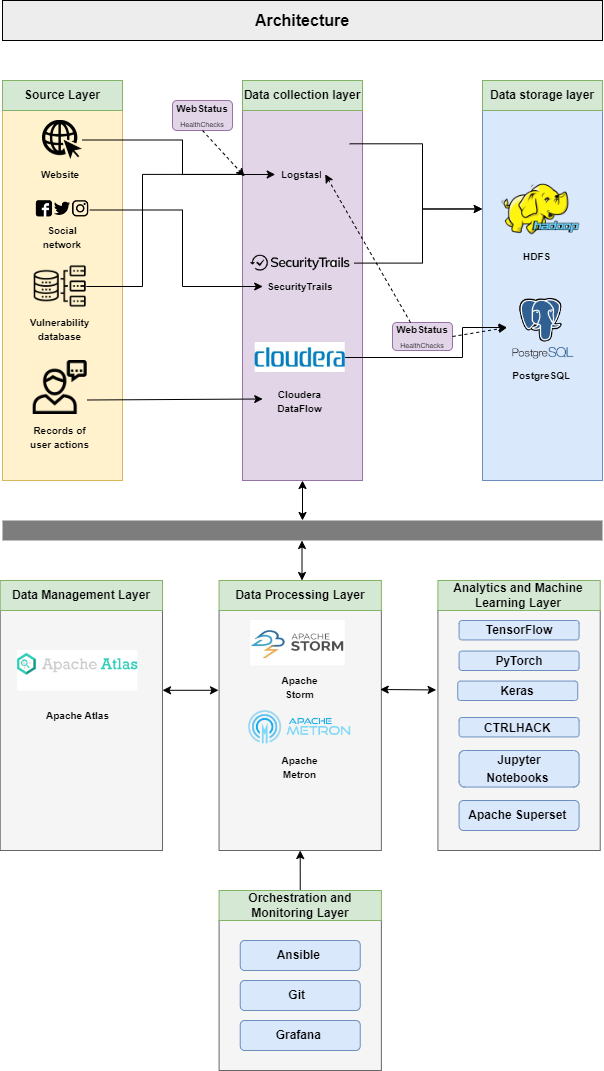


Рисунок 4 - OLAP и OLTP

**5. Схема архитектуры.**



**6. Процесс обработки данных.**

- Данные собираются из различных источников через слой сбора данных.

- Сырые данные сохраняются в HDFS для долгосрочного хранения или в PostgreSQL структурированные данные.

- Потоковые данные обрабатываются в реальном времени с помощью Apache Storm для быстрой аналитики.

- Расщиренная аналитика в режиме реального времени осуществляется с помощью Apache Metron.

- Аналитики используют Jupyter Notebooks и Superset для исследования данных и создания отчетов.

- Модели машинного обучения обучаются на исторических данных, также с помощью имитаций угроз и развертываются для прогнозирования.

**7. Масштабирование и отказоустойчивость.**

- Использование кластерной архитектуры Hadoop для горизонтального

масштабирования.

- Репликация данных в HDFS и PostgreSQL для обеспечения отказоустойчивости.

- Использование Ansible для оркестрации рабочих процессов, для проверки соответствия системы политикам организации и нормативным требованиям.

- Применение Git для управления конфигурациями.

**8. Безопасность.**

- Реализация шифрования данных с помощью HDFS Transparent Encryption.

- Применение Apache Atlas для ограничений доступа на основе ролей и соблюдения политик организации.

- Регулярное резервное копирование и план аварийного восстановления.

**Выводы**

1. Архитектура получилась масштабируемой и отказоустойчивой благодаря таким технологиям как HDFS и Ansible.
2. Разработанная архитектура учитывает, как работу с неструктурированными, полуструктурированными данными, так и со структурированными.
3. Используемые решения в основном с открытым исходным кодом, являются свободным ПО, что подходит для средней организации, которая не имеет достаточных ресурсов для дорогих инструментов.
4. Входят инструменты, которые легко подстроить под политику организации и настроить уровень доступа.
5. Включает внедрение имитаций угроз для улучшения моделей машинного обучения.
6. Используется Apache Metron, специально предназначенный для работы с большими данными по кибербезопасности.
7. Для хранения данных больше подходит снежинка, она уменьшает дублирование данных, также в сфере кибербезопасности важна целостность данных, их безопасность.