Домашнее задание

1. Что могло повлиять на утечку данных в Кото.Еде?

- Приложения типа "доставки" имеют как минимум 2 базы данных, первая - для пользования и анализа внутри компании, другая для курьеров.

Сразу же хочется заметить, какие данные были украдены - адреса клиентов, их заказы, телефоны и комментарии. Такую информацию обычно видят курьеры.

- Достаточно часто приложения доставки являются одним из микросервисов больших компаний

Я обратил внимание на название - "Кото.Еда", основное название компании и предполагаемый микросервис разделены точкой, возможно, что доставка еды это не единственное чем занимаются котики.

Вывод:

Данную информацию изначально можно назвать слабозащищенной так как к ней имеют доступ достаточно много людей, а это значит, что компания может недобросовестно защищать такие данные, но в любом случае этого требует законодательство (так что минимальные средства защиты должны быть), я предполагаю, что данная утечка произошла через какой-нибудь другой микросервис находящийся внутри компании, который совсем не защищен (возможно по причине того, что там еще нет никакой важной информации, либо он находится на стадии тестирования).

2. С чего начать расследование инцидента в Кото.Еде?

Первым делом нужно проанализировать сетевой трафик в тот момент, когда выгружались данные, с помощью данной информации можно определить с каким ip адресом происходило взаимодействие, дальше есть 2 варианта:

1) Если взаимодействие происходило с доверенным внутри компании микросервисом, то его нужно отключить и найти уязвимость.

2) Если взаимодействие происходило с неизвестным ip адресом (что навряд ли), то необходимо в целом оценить безопасность компании, так как данный микросервис может быть связан с другими.

Если же в сетевом трафике нет подозрительной активности, я бы подумал в сторону недобросовестных сотрудников, у которых есть достаточное количество привилегий (возможно проблема в том, что доступ имеют сотрудники, которые не должны).

3. Если бы котики жили в идеальном мире с неограниченными ресурсами для защиты данных, как можно было бы избежать инцидента?

Если прям неограниченные ресурсы, то шифрование (так как от этого процесса отказались из за того, что он трудоемкий, и нужно выполнять большое количество операций)

С другой стороны пароли сейчас хранятся в хэшированном виде, возможно можно подумать в сторону хэширования такой информации (хранить ее на таком же уровне безопасности как и пароли)

4. Как можно определить, что в базе данных содержатся персональные данные сотрудников?

По названию таблицы, столбцов, комментариев.

Там должны быть данные которые позволяют идентифицировать личность (паспорт).

5. К каким таблицам PostgreSQL вам нужен доступ, и какие данные вы будете извлекать для классификации?

Для классификации данных в PostgreSQL потребуется доступ к системным и пользовательским таблицам, чтобы проанализировать структуру базы данных, а также определить, какие данные могут содержать персональные или конфиденциальные сведения.

- Системные таблицы:

Имена таблиц и типы объектов (таблицы, индексы, последовательности).

Метаданные столбцов, включая имя, порядок и принадлежность.

Ограничения на уровне столбцов и таблиц, такие как первичный ключ, уникальность.

- Пользовательские таблицы:

Полный список таблиц и колонок для дальнейшего анализа данных.

6. Если вам не дают доступ или база недоступна для подключения из-за устаревшей версии, как решить такую задачу?

Если доступ к бд в live режиме недоступен, то можно попросить ее дамп у администратора БД, так нам будет доступна резервная копия для анализа.

Также можно создать тестовую бд для эмуляции (если имеется хотябы описание/схема)

7. Какие таблицы PostgreSQL вы бы проанализировали для выявления избыточных прав доступа и оценки рисков, и какие данные из них помогут в этом?

Таблица pg_roles, содержит информацию о ролях в PostgreSQL, включая их привилегии и атрибуты.

pg_auth_members, показывает связи между ролями.

Таблица information_schema.role_table_grants описывает все назначенные для таблиц и представлений права, в которых праводателем или правообладателем является текущая активная роль.

pg_default_acl хранит начальные привилегии, назначаемые вновь создаваемым объектам.

8. Какие действия предпринять, если обнаружено, что роль PUBLIC имеет доступ к чувствительным данным?

Так как роль PUBLIC применяется ко всем пользователям базы данных по умолчанию, мы получаем большую угрозу безопасности. После определения объектов, доступ к которым предоставлен роли PUBLIC, нужно отозвать лишние права. Далее назначить доступ к этим данным тем, кому он необходим.

9. Какие действия следует предпринять, если в трафике обнаружена выгрузка ранее классифицированных данных?

Нужно определить ip адрес на который происходит выгрузка и заблокировать данное соединение на уровне firewall или сетевого оборудования, далее нужно проверить учетные записи пользователей с которых происходит выгрузка и отключить их.

После этого нужно провести анализ инцидента, определить объем утечки, причину утечки.

Далее нужно устранить причину утечки.

10. Какие данные из трафика можно использовать для анализа инцидента?

1) Метаданные трафика, эти данные позволяют идентифицировать отправителей, получателей и параметры передачи.
2) Используемые протоколы (HTTP, HTTPS, FTP, SSH и т. д.).
3) Содержимое пакетов, к примеру HTTP-заголовки (Host, User-Agent, Referer), заголовки аутентификации (Authorization, Cookie).
4) Содержимое передаваемых данных (Полезная нагрузка)
5) Информацию о сессии в целом, время начала/конца сессии, продолжительность, последовательности пакетов (flows), связывающие IP-адреса, порты и протоколы.

11. У вас есть сервис Кото.Еда, состоящий из множества микросервисов, каждый из которых имеет собственную базу данных, включая сервис заказов, который подвергся утечке. Можем ли мы определить список всех микросервисов (и их баз данных), с которыми он взаимодействует?

Нужно проанализировать исходный код данного сервиса, первым делом можем посмотреть на используемые библиотеки, к примеру requests говорит нам о том, что мы имеем связь с внешним сервисом, далее следует найти url других сервисов с которыми мы взаимодействуем через вызовы API. Обычно для большего удобства адреса таких сервисов записываются в переменные, на это тоже нужно обратить внимание. Следует искать внешние HTTP запросы. Для того, чтобы определить к каким БД мы имеем доступ, в коде нужно найти моменты подключения к БД.

В данном сервисе следует проанализировать БД на предмет подключений к внешним базам. Если доступно логирование запросов к базе данных, нужно проверить их на предмет взаимодействия с внешними базами.

После выполнения анализа мы получим: список микросервисов, список баз данных, типы взаимодействий.

12. Как безопасно хранить пароли в Data Security, которые используются для подключения к базам данных для классификации?

В PostgreSQL наиболее безопасный способ хранения паролей — использовать специализированные решения для управления такими данными (Мне удалось найти HashiCorp Vault, AWS Secrets Manager). Пароли нужно хранить в бд в хэшированном/зашифрованном виде.

Если пароли хранятся локально или в конфигурационных файлах, их также нужно зашифровать. (Но тут нужно защищать конфигурационные файлы)

Если пароли хранятся в коде (к примеру для подключения к бд), то нужно хранить их в переменных окружения и передавать их в приложение, также в зашифрованном виде.

Крайний способ - хранить пароль в удаленном хранилище и пользоваться им в случае необходимости. Нужно помнить, что пароли должны быть изолированы и правильно подобраны (в соответствии с требованиями ИБ)

13. Какой механизм аутентификации вы бы использовали для пользователей системы Data Security? Например, как вы бы реализовали проверку пользователей, входящих в саму систему?

Я бы предложил такой механизм:

1) Пользователь регистрируется, тоесть передает имя, логин, пароль.
2) Пароль хэшируется и сохраняется в БД.
3) Далее пользователь логинится (отправляет логин и пароль).
4) Пользователь получает одноразовый код на второе устройство, вводит его. (OTP)
5) Генерируется токен JWT с информацией о пользователе.
6) Клиент получает токен и использует его для дальнейших запросов.
7) При каждом запросе токен проверяется на валидность.

Для системы Data Security лучше всего использовать JWT для аутентификации с возможностью добавления двухфакторной аутентификации (OTP). Это обеспечит безопасность, масштабируемость и удобство интеграции с другими сервисами.

Также нужно внедрить системы мониторинга для предотвращения инцидентов и большей вероятности.

14. Как можно, подключившись к базе данных, определить, в какой среде она находится: production, тестовая или разработка?

В тестовой среде есть вероятность, что некоторые сущности будут иметь приписку "test". Могут присутствовать дополнительные таблицы для отладки или тестов.

Тестовая среда вероятно содержит меньше данных чем production.

В тестовой бд располагаются тестовые данные заглушки, тогда как в продуктовой можно увидеть реальные ФИО, настоящие адреса, почты.

В production базе будет активность пользователей или служб, тогда как в тестовой среде активность часто ограничивается только действиями разработчиков.

Проверить домены, в тестовой среде вероятно будет использоваться localhost (127.0.0.1), тогда как в продуктовой среде мы можем увидеть реальный действующий домен.

15. Как определить, пользуются базой данных PostgreSQL или нет? Если база используется, как выявить объекты, к которым пользователи не обращаются, лежат мертвым грузом?

Нужно использовать системное представление pg_stat_activity, чтобы увидеть текущие подключения и активность. Если включено логирование запросов, проверьте журналы активности.

Используем pg_stat_user_tables, чтобы посмотреть, к каким таблицам обращались. С помощью запросов можно выявить какие таблицы не используются. Также с помощью запросов можно проверить отальные объекты которые не используются (индексы (pg_stat_user_indexes), последовательности (pg_class + pg_stat_user_tables), функции (pg_proc).

Нужно автоматизировать поиск неиспользуемых объектов.

16. Как вы бы классифицировали данные в ClickHouse?

Определение целей классификации:

1) Оптимизация запросов.
2) Обеспечение безопасности (например, защита персональных данных).
3) Выявление "холодных" и "горячих" данных.
   Классификация данных по критериям:
4) Чувствительность данных (персональные, коммерческие, публичные).
5) Частота использования данных.
6) Тип данных (числовые, строковые, временные, метрики).

17. В какой таблице и в какой колонке в базе данных Confluence хранится информация о страницах?

В таблице confluence_page хранится информация о страницах.

Некоторые колонки этой таблицы:

1) Body. Содержимое тела страницы, которое хранится в виде объекта JSON.
2) Created at. Дата и время создания страницы.
3) Created by. Уникальный идентификатор создателя страницы.
4) Status. Текущий статус страницы.
5) Title. Название страницы.
6) Type. Указывает, является ли контент страницей или публикацией в блоге. С помощью этого столбца можно фильтровать данные по типу контента. 1
7) Updated at. Дата и время последнего обновления страницы.
8) URL. URL страницы.

18. Как определить, что сотрудник работает в CRM системе, открывает карточки сотрудников компании и фотографирует их данные (камера в очках)? Можно ли по фото понять, кто его сделал?

Добавить задержку на открытие карточек и отслеживать моменты, когда с одного устройства пытаются открывать много карточек. (Настроить оповещение о массовом открытии карточек)

Можно извлечь метаданные и по времени определить кто в этот момент работал в CRM системе.

Необходимо установить камеры видеонаблюдения в местах, где сотрудники работают с CRM, возможно подключить камеры, которые распознают устройства (например, очки с камерой) или нехарактерные движения.

В целом нужно:

1) Ограничить использование персональных устройств в офисе (таких как очки с дополнительными функциями).
2) Использовать экраны с функцией защиты от просмотра (privacy screens)

19. Назовите таблицу, в которой хранятся пользователи 1С. Если данные зашифрованы, можно ли получить их содержимое и как это сделать? Можно ли сбросить пароль?

Пользователи в программе 1С хранятся в таблице V8users.

В 1С данные хэшируются с помощью объекта «ХешированиеДанных». Он позволяет получить хеш по алгоритмам MD5 или CRC32, принимая на вход строку или «ДвоичныеДанные». Данные можно расшифровать используя инструменты для брутфорса (hashcat, john the ripper), либо же онлайн сервисы такие как crackhash. Если данные именно зашифрованны, то их можно расшифровать узнав алгоритм шифрования. Если имеется доступ к БД, то можно заменить пароль привилегированного для просмотра данных лица пустым паролем, тем самым получить доступ к данным, либо сбросить пароль.

В разделе администрирования пользователей можно сбросить пароль через интерфейс, либо через файл .1cd с помощью сторонних утилит, например, 1C:DBExplorer.

20. Над какой частью функционала Data Security вы бы хотели работать прямо сейчас?

Я бы хотел:

1) Работать с системами обнаружения и предотвращения вторжений (IDS/IPS). Заниматься мониторингом сетевого трафика и анализом аномалий для обнаружения и блокирования потенциальных угроз.
2) Заниматься анализом и разграничением доступа к данным. Анализировать текущие матрицы доступов к чувствительной информации, выявлять недостатки, разрабатывать (либо использовать готовые) инструменты для разграничения и выстраивания принципа минимальных привилегий (least privilege).