Аннотация – в документе представлен анализ уязвимостей в решения класса Mobile Device Management (MDM). Анализ охватывает различные аспекты этих уязвимостей, включая их технические детали, потенциальные векторы атак и последствия для специалистов по безопасности и организаций в различных отраслях.

Анализ предоставляет высококачественную сводную информацию об этих уязвимостях, предлагая ценную информацию специалистам по безопасности, ИТ-администраторам и другим специалистам. Понимая эти уязвимости и их последствия, организации могут лучше защищать свои решения MDM, повышать уровень безопасности и снижать риски, связанные с этими недостатками. Этот документ служит важным ресурсом для тех, кто хочет защитить свои системы управления мобильными устройствами от сложных киберугроз.

I. BlackBerry MDM

В системе управления мобильными устройствами BlackBerry MDM обнаружены значительные проблемы в механизмах аутентификации и, несмотря на сложность анализа для восстановления большей части кода приложения, была получена информация о формировании запросов discovery, выполняемых в отношении конечных устройств.

A. Reverse и сложности MITM

Процесс реверса кода клиента BlackBerry MDM, в частности, из Android APK, включает в себя несколько сложных шагов и инструментов, предназначенных для анализа приложения на уровне кода, но имеет решающее значение для понимания функциональности приложения, выявления потенциальных уязвимостей и обеспечения надёжности мер безопасности.

1) Понимание структуры APK

Android Package Kit (APK) - это, по сути, формат файла пакета, используемый операционной системой Android для распространения и установки мобильных приложений. Это zip-архивный файл, содержащий все необходимые файлы для запуска приложения для Android. К этим файлам относятся:

- **AndroidManifest.xml**: указаны разрешения, которые должно иметь приложение, а также аппаратные и программные функции, необходимые приложению.
- **classes.dex**: скомпилированный исходный код Java, преобразованный в исполняемый формат Dalvik.
- **Ресурсы**: ресурсы, используемые приложением, такие как изображения, строки и файлы макета.

2) Роль dex2jar

Инструмент dex2jar играет ключевую роль в процессе reverse-исследование. Он предназначен для выполнения преобразования файлов DEX (Dalvik Executable) в файлы Java Archive (JAR). Инструмент dex2jar работает путем извлечения файла classes.dex из APK и преобразования его в файл JAR, который затем может быть декомпилирован в исходный код Java с помощью различных Java-декомпиляторов.

3) Декомпиляция JAR-файла

После преобразования файла DEX в файл JAR с помощью dex2jar следующий шаг включает декомпиляцию файла JAR для получения исходного кода Java. Именно здесь в игру вступают Java-декомпиляторы. Для декомпиляции JAR-файла можно использовать такие инструменты, как JD-GUI, JADX или FernFlower, которые предоставляют представление исходного кода приложения. Этот декомпилированный исходный код имеет решающее значение для понимания внутренней работы приложения, хотя важно отметить, что код может не совсем соответствовать исходному коду из-за процессов компиляции и декомпиляции.

4) Анализ декомпилированного кода

Используя исходный код Java, полученный в процессе декомпиляции, аналитики могут начать изучать код для различных целей, таких как:

- **Анализ безопасности**: Выявление уязвимостей в системе безопасности приложения, таких как закодированные секреты, небезопасная сетевая связь или ненадлежащие методы хранения данных.
- **Понимание функциональности**: Получение представления о том, как работает приложение, включая его взаимодействие с серверными компонентами, обработку данных и динамику пользовательского интерфейса.
- **Проверка соответствия**: Обеспечение соответствия приложения соответствующим правилам и стандартам, особенно в отношении защиты данных и конфиденциальности.

5) Pinning сертификата

Pinning сертификата — это метод безопасности, используемый для предотвращения атак "Человек

посередине" (MitM) путём обеспечения того, чтобы приложение доверяло только определённым сертификатам. Этот метод особенно важен для мобильных приложений, которые обрабатывают конфиденциальные данные, таких как клиент BlackBerry MDM.

6) Как работает Pinning сертификата

- **Внедрение сертификатов**: приложение внедряет копию сертификата сервера или его открытый ключ в само приложение на этапе разработки
- **Процесс проверки**: когда приложение устанавливает соединение с сервером, оно сравнивает сертификат сервера с фиксированным сертификатом. Если сертификаты совпадают, подключение разрешено; в противном случае подключение отклоняется.
- **Предотвращение атак MitM**: Этот процесс предотвращает перехват злоумышленниками данных, передаваемых между приложением и сервером, и их подделку, поскольку им потребуется предоставить точно такой же сертификат, который ожидает приложение.

7) Обход Pinning сертификата с помощью Frida

Frida — это динамический инструментарий, который позволяет разработчикам и исследователям безопасности внедрять пользовательские сценарии в работающие приложения. Эта возможность делает его мощным инструментом для обхода механизмов безопасности, таких как закрепление сертификата.

8) Настройка среды:

- **Рут-устройство или эмулятор**: чтобы использовать Frida, исследователю необходимо рут-устройство Android или эмулятор с root-доступом. Это необходимо для внедрения кода в запущенное приложение.
- **Сервер Frida**: Сервер Frida установлен и запущен на устройстве. Этот сервер облегчает обмен данными между клиентом Frida (запущенным на компьютере исследователя) и целевым приложением.

9) Написание сценария для "Фриды":

- **Подключение методов SSL**: Исследователь пишет скрипт Frida для подключения методов SSL / TLS, используемых приложением. Этот скрипт перехватывает методы, ответственные за проверку сертификата.
- **Переопределение логики проверки**: Скрипт изменяет поведение этих методов, чтобы обойти проверки сертификата. По сути, это заставляет приложение принимать любой сертификат, представленный сервером, независимо от того, соответствует ли он фиксированному сертификату.
- **Эффект**: после внедрения скрипта он подключается к методам SSL / TLS и переопределяет логику проверки сертификата, эффективно обходя механизм закрепления.

---

javascript
Java.perform(function () {
var TrustManagerImpl =
Java.use('com.android.org.conscrypt.TrustManagerImpl');

TrustManagerImpl.verifyChain.implementation = function (untrustedChain, trustAnchorChain, host, clientAuth, ocspData, tlsSctData) {
// Bypass the certificate validation
return untrustedChain;
};
});

Injecting the Script:
Running the Script: The researcher uses the Frida client to inject the script into the running BlackBerry MDM client. This is done using a command like:
bash
frida -U -f com.blackberry.mdmclient -l bypass_ssl.js --no-pause

---

10) Выполнение MitM-атаки:

- **Настройка прокси-сервера**: после обхода закрепления сертификата исследователь может настроить прокси-инструмент, такой как Burp Suite или mitmproxy, для перехвата и анализа сетевого трафика между клиентом BlackBerry MDM и его сервером.
- **Анализ трафика**: после предыдущего шага возможно проверять передаваемые данные, выявлять потенциальные уязвимости и понимать схемы взаимодействия приложения.

11) Последствия обхода pinning'а сертификата

Обход сертификата подвергает приложение MitM-атакам, позволяя злоумышленникам перехватывать передаваемые данные и потенциально манипулировать ими. Это может привести к различным проблемам безопасности, таким как утечка данных, несанкционированный доступ и утечка информации.

B. Недостатки аутентификации

Discovery-Запрос является основной частью процесса аутентификации клиента BlackBerry MDM, предназначенного для определения местоположения соответствующей конечной точки MDM. Хотя он включает в себя такие механизмы, как X-authToken для проверки подлинности, зависимость от однофакторной аутентификации (SFA) создаёт потенциальные риски для безопасности. Внедрение передовых методов, таких как использование HTTPS и внедрение многофакторной аутентификации (MFA), может значительно повысить безопасность системы MDM

1) Инициализация клиента:

При запуске клиента BlackBerry MDM пользователю предлагается ввести свой адрес электронной почты. Этот адрес электронной почты используется для идентификации пользователя и инициирования discovery-процесса.

2) Выполнение discovery-запроса:

Клиент создаёт discovery-запрос, который представляет собой вызов HTTP POST. Этот запрос отправляется на заранее определённый URL discovery-службы. Запрос обычно включает адрес электронной почты пользователя и другую информацию, необходимую для определения местоположения конечной точки MDM.

3) Полезная нагрузка запроса:

Полезная нагрузка discovery-запроса включает в себя несколько ключевых фрагментов информации:

- **Адрес электронной почты**: Адрес электронной почты, указанный пользователем.
- **Информация об устройстве**: подробные сведения об устройстве, такие как версия операционной системы, тип устройства и версия приложения.
- **Политики аутентификации**: Информация о политиках аутентификации, которые поддерживает или требует клиент.

4) Ответ сервера:

Discovery-служба обрабатывает запрос и отвечает полезной нагрузкой в формате XML или JSON:

- **EnrollmentServiceUrl или URL службы регистрации**: URL конечной точки MDM, с которой клиент должен связаться для регистрации и дальнейшего общения.
- **AuthPolicy**: определяет тип аутентификации, требуемый сервером MDM. Это может быть OnPremise, Federated или другие поддерживаемые значения.
- **Дополнительная информация о конфигурации**: Любые другие необходимые сведения о конфигурации, необходимые клиенту для продолжения процесса регистрации.
- **Местоположение конечной точки:** Клиент использует информацию, предоставленную в ответе сервера, для определения местоположения конечной точки MDM. Именно в эту конечную точку клиент будет отправлять последующие запросы на регистрацию, настройку и управление.

5) Последствия для безопасности

- **X-authToken**: Discovery-запрос включает заголовок X-authToken, который используется для проверки запроса. Этот токен гарантирует, что запрос является легитимным и авторизованным. Если токен отсутствует или недействителен, сервер отвечает ошибкой 401.
- **Однофакторная аутентификация (SFA)**: BlackBerry MDM, как и другие решения MDM, такие как AirWatch и MobileIron, уязвима для однофакторной аутентификации (SFA). Это означает, что первоначальная аутентификация зависит исключительно от одного фактора, такого как пароль или токен, без дополнительных уровней безопасности. Зависимость от SFA представляет потенциальную угрозу безопасности, поскольку злоумышленникам легче скомпрометировать один фактор аутентификации по сравнению с многофакторной аутентификацией (MFA), которая требует нескольких форм проверки.
- **Потенциальные уязвимости**: если discovery-запрос и последующие процессы аутентификации не защищены должным образом, злоумышленники потенциально могут перехватить сообщение или манипулировать им. Это может привести к несанкционированному доступу к системе MDM, утечке данных или другим инцидентам безопасности. Для снижения этих рисков важно обеспечить передачу discovery-запроса по защищённому каналу (например, HTTPS) и наличие надёжных механизмов аутентификации.

C. X-authToken

X-authToken — это токен безопасности, включённый в HTTP-заголовки discovery-запроса, отправляемого клиентом BlackBerry MDM. Его основная цель – аутентифицировать и авторизовать запрос, гарантируя, что он исходит из легитимного источника и не является злонамеренной или несанкционированной попыткой доступа к серверу MDM.

1) Проверка запроса

Когда сервер MDM получает discovery-запрос, он проверяет наличие заголовка X-authToken. Если заголовок отсутствует или содержит недопустимый токен, сервер ответит ошибкой 401, которая указывает на то, что запрос не прошёл процесс аутентификации и не может быть продолжен дальше.

2) Генерация токенов и управление ими

X-authToken, скорее всего, генерируется и управляется сервером MDM или связанной службой аутентификации. Сам токен может быть криптографически защищённым случайным значением, веб-токеном JSON (JWT) или любой другой формой защищённого токена, который может быть проверен сервером. Процесс получения и включения X-authToken в discovery-запрос обычно выполняется самим клиентским приложением MDM.

- **Начальная аутентификация**: Клиенту может потребоваться выполнить начальный процесс аутентификации, такой как предоставление учётных данных пользователя или информации об устройстве, для получения X-authToken.
- **Хранилище токенов**: Полученный токен затем надёжно сохраняется на клиентском устройстве либо в памяти, либо в безопасном месте хранения.

- **Включение токена**: при создании discovery-запроса клиент включает X-authToken в соответствующий HTTP-заголовок.

3) Последствия для безопасности

Наличие X-authToken в discovery-запросе является мерой безопасности, разработанной для предотвращения несанкционированного доступа к серверу MDM. Требуя действительный токен, сервер может гарантировать, что только авторизованные клиенты смогут инициировать discovery-процесс и потенциально получить доступ к конфиденциальным функциям MDM.

Однако важно отметить, что X-authToken сам по себе может не обеспечивать достаточной безопасности, если им должным образом не управлять и не защищать.

- **Истечение срока действия токена**: Внедрение механизмов истечения срока действия токена для ограничения срока действия токена, снижения риска неправильного использования токена в течение длительного периода.
- **Ротация токенов**: регулярная ротация или обновление токена для дальнейшего снижения риска кражи токенов или повторных атак.
- **Безопасная передача**: Передача токена по защищённому каналу (например, HTTPS) для предотвращения перехвата и несанкционированного доступа.
- **Многофакторная аутентификация (MFA)**: объединение токена с дополнительными факторами аутентификации, такими как учётные данные пользователя или биометрические данные, для повышения общей безопасности процесса аутентификации.

D. Однофакторная аутентификация (SFA)

Однофакторная аутентификация использует только один фактор для проверки личности пользователя, обычно пароль или другой фактор, основанный на знаниях. Хотя этот метод широко используется из-за своей простоты, он считается менее безопасным, чем многофакторная аутентификация (MFA), которая требует двух или более факторов для аутентификации.

1) Уязвимость в MDM-приложениях

Само приложение BlackBerry MDM уязвимо для SFA. Это означает, что пользователи могут получить доступ к функциональности MDM и потенциально конфиденциальным данным или конфигурациям, предоставив только один фактор аутентификации, такой как пароль.

2) Отсутствие многофакторной аутентификации (MFA)

Отсутствие MFA в приложении MDM считается недостатком системы безопасности, поскольку оно увеличивает риск несанкционированного доступа. Если злоумышленнику удастся скомпрометировать пароль пользователя (с помощью фишинга, атак методом перебора или других средств), он потенциально может получить доступ к приложению MDM и связанным с ним ресурсам без каких-либо дополнительных барьеров аутентификации.

3) Последствия для безопасности

Использование SFA в приложениях MDM может иметь серьёзные последствия для безопасности, включая:

- **Несанкционированный доступ**: Используя только один фактор аутентификации, злоумышленник, получивший учётные данные пользователя, может легко получить несанкционированный доступ к приложению MDM и связанным с ним ресурсам.
- **Утечки данных**: Если приложение MDM управляет конфиденциальными данными или конфигурациями, успешное нарушение может привести к утечке данных, компрометации устройств или другим инцидентам безопасности.
- **Комплаенс-риски**: Многие отраслевые нормативные акты и стандарты безопасности требуют использования MFA для доступа к конфиденциальным системам или данным, особенно в регулируемых отраслях, таких как здравоохранение, финансы и государственное управление.
- **Увеличение векторов атак**: Отсутствие MFA в приложении MDM расширяет поверхность атаки, поскольку злоумышленнику нужно скомпрометировать только один фактор аутентификации, чтобы получить доступ.

E. Однофакторная аутентификация (SFA) в BlackBerry MDM

Наличие однофакторной аутентификации (SFA) в клиенте BlackBerry MDM вызывает серьёзные опасения по поводу безопасности. SFA использует только одну форму аутентификации, обычно пароль, для подтверждения личности пользователя. Этот метод по своей сути менее безопасен по сравнению с многофакторной аутентификацией (MFA), которая требует двух или более независимых учётных данных для проверки. Вот ключевые проблемы безопасности, связанные с SFA в клиенте BlackBerry MDM:

1) Подверженность атакам

- **Фишинговые атаки**: SFA очень уязвима для фишинговых атак, когда злоумышленники обманом заставляют пользователей раскрывать свои пароли. После взлома пароля злоумышленник может получить несанкционированный доступ к MDM-клиенту и, возможно, ко всей корпоративной сети.
- **Атаки методом перебора**: Злоумышленники могут использовать автоматизированные инструменты для проведения атак методом перебора, систематически пробуя различные комбинации паролей, пока не будет найдена правильная. Без дополнительных уровней безопасности SFA мало что делает для предотвращения подобных атак.

- **Credential Stuffing**: используются списки скомпрометированных паролей, полученных в результате других взломов, для получения доступа к учётным записям. Поскольку многие пользователи повторно используют пароли в разных сервисах, SFA не обеспечивает адекватной защиты от этого типа атак.
- **Социальная инженерия**: злоумышленники часто используют методы социальной инженерии, чтобы манипулировать пользователями и заставить их разглашать свои пароли. SFA не предлагает никаких дополнительных шагов проверки для противодействия этой тактике.
- **Замена SIM-карты и переадресация вызовов**: злоумышленники могут перехватывать SMS-сообщения и голосовые вызовы с помощью замены SIM-карты и переадресации вызовов, которые являются распространёнными методами обхода SFA, когда он полагается на OTP на основе SMS