Отчет о тестировании требований безопасности

|  |  |  |
| --- | --- | --- |
| Наименование приложения | | TUT.BY Афиша |
| Версия | | 25.3.1 |
| Дата проверки | | 04.03.2020 |
| Проверил | | Павлив Елизавета |
| Конфиденциальная информация | | Данные об участии в розыгрышах, электронная почта и пароль к ней, телефонный номер. |
| **Требования безопасности** | | **Результат** |
| 1 | Ненадежное хранение данных | - |
| 1.1 | Локальное хранение конфиденциальных данных  STORAGE-1 и STORAGE-2 | - |
| 1.1.1 | наличие разрешений AndroidManifest.xml для read/write во внешнем хранилище | <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> |
| 1.1.2 | наличие прав доступа к файлам MODE\_WORLD\_READABLE или MODE\_WORLD\_WRITABLE | нет |
| 1.1.3 | наличие классов и функций:  SharedPreferences;  FileOutPutStream;  getExternal \*;  getWritableDatabase;  getReadableDatabase;  getCacheDir;  getExternalCacheDirs. | SharedPreferences (используется сторонними библиотеками com.facebook, com.yandex.metrica и др., не содержит конфиденциальные данные).  FileOutPutStream, FileInputStream, getExternalStorageState, getExternalStorageDirectory, getExternalCacheDir, getExternalFilesDir, getCacheDir (используются сторонними библиотеками com.facebook, com.yandex.metrica и др. для записи кэша во внешнее хранилище – не конфиденциальные данные).  getWritableDatabase, getReadableDatabase (в базе данных хранятся данные о текущих событиях в городе, розыгрышах, включая id, присвоенный каждому участнику, а также app\_id, cid, tid – нет конфиденциальной информации, такой как электронная почта или пароль). |
| 1.1.4 | конфигурационные файлы build configs, такие как local.properties, gradle.properties | - |
| 1.1.5 | файлы ресурсов | Содержат ключи доступа к сервисам google, yandex, facebook и др. – не конфиденциальная информация.  <string name="default\_web\_client\_id">178306988327-4jimsgos8sr8vrdjfjqhr6tuoksmdtkk.apps.googleusercontent.com</string>  <string name="facebook\_app\_id">1541020166144878</string>  <string name="google\_api\_key">AIzaSyDd\_\_ISeoDPDqcZYI9euHgSlh4Ee3MIi0g</string>  <string name="google\_app\_id">1:178306988327:android:88bcdb26fd8b52f9</string>  <string name="google\_crash\_reporting\_api\_key">AIzaSyDd\_\_ISeoDPDqcZYI9euHgSlh4Ee3MIi0g</string>  <string name="com.crashlytics.android.build\_id">b2d78061-2b1a-44c8-b601-6e7ee0493ec7</string>  <string name="yandex\_appmetrica\_api\_key">be644b15-35c6-41e0-a0be-0b565044a7a5</string> |
| 1.2 | Файлы отчетов (log files) STORAGE-3 | В \data\data\by.tut.afisha.android не обнаружены файлы отчетов. С помощью Logcat было выявлено, что логи содержат только служебную информацию. |
| 1.2.1 | функции и классы, такие как:  android.util.Log  Log.d | Log.e | Log.i | Log.v | Log.w | Log.wtf  Logger | Log.d, Logger |
| 1.2.2 | ключевые слова и методы:  System.out.print | System.err.print  logfile  logging  logs | logging, logs |
| 1.3 | Передача конфиденциальных  данных третьей стороне  STORAGE-4 | Сторонние библиотеки: com.crashlytics, com.facebook, com.squareup.okhttp, com.squareup.picasso, retrofit2, com.yandex.metrica, com.vk.api.sdk, com.google.  Авторизация пользователя в приложении может быть осуществлена с помощью facebook, vk.com, google – следовательно, им передаются конфиденциальные данные, такие как электронная почта, логин, номер телефона и пароли. |
| 1.4 | Поля ввода конфиденциальных данных STORAGE-5 | Поле для ввода номера телефона или email – кэш клавиатуры включен, пароль – кэш клавиатуры отключен; |
| 1.5 | Безопасность межпроцессного взаимодействия STORAGE-6 | android: exported= “false” во всех content providers, для всех тегов <provider> не определен <intent-filter>.  Данные не защищены тегом android:permission.  android.content.ContentProvider, android.database.Cursor, android.database.sqlite, .query, .update, .delete в файлах не найдены. |
| 1.6 | Безопасность данных в пользовательском интерфейсе STORAGE-7 | Пароль маскируется заменой ввода  точками, приложение не передает  данные в пользовательский интерфейс. |
| 1.7 | Резервное копирование STORAGE-8 | android:allowBackup="true" |
| 1.8 | Приложение скрывает конфиденциальные данные с экрана, когда находится в фоновом режиме STORAGE-9 | FLAG\_SECURE не установлен. Конфиденциальная информация отображается в фоновом режиме. |
| 1.9 | Приложение не хранит конфиденциальные данные в памяти дольше, чем  необходимо STORAGE-10 | Конфиденциальные данные обрабатываются самим приложением или сторонними библиотеками – для входа в приложение через facebook, vk.com или google аккаунт.  Для хранения данных из полей для ввода логина/пароля EditText используются неизменяемые и непримитивные типы данных: String, StringBuilder.  Конфиденцальная информация не очищается в тот момент, когда срабатывает событие onPause LoginFragment – фрагмента для ввода логина и пароля (com.facebook, в других библиотеках и приложении - аналогично). После удаления ссылок сбор мусора не запрашивается и ссылки не перезаписываются.  Пользователь не должен повторно проходить аутентификацию каждый раз, когда приложение возобновляет работу. |
| 1.10 | Приложение требует от пользователя минимальную настройку доступа к  устройству STORAGE-11 | Settings.Secure не используется. Приложение не проверяет устройство на следующее:  1) установлена ли на устройстве блокировка с помощью PIN или пароля;  2) активирован ли механизм отладки по USB;  3) используется ли шифрование устройства;  4) рутировано ли устройство.  Проверяет установлена ли на устройстве версия Android OS >= 23 (Marshmallow); |
| 1.11 | Проверка сертификатов X.509 (NETWORK‑3) | В манифесте приложения не включен флаг android:debuggable.  ALLOW\_ALL\_HOSTNAME\_VERIFIER не используется.  - сертификат подписан доверенным центром сертификации;  - срок действия сертификата – годен;  - самоподписанный сертификат – есть. |
| 1.11.1 | Примеры использования TrustManager | 1) SSLContext sSLContext = SSLContext.getInstance("TLS");  sSLContext.init(null, new TrustManager[] { new efv(new efw(paramefu.a(), paramefu.b()), paramefu) }null);  2) TrustManagerFactory trustManagerFactory = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());  trustManagerFactory.init((KeyStore)null);  TrustManager[] arrayOfTrustManager = trustManagerFactory.getTrustManagers(); |
| 1.11.2 | Примеры использования HostnameVerifier | if (hostnameVerifier.verify(a(paramString), paramSocket.getSession())) return paramSocket; |
| 1.12 | Проверка SSL pinning  (NETWORK‑4) | Запись <pin> отсутствует в файле network\_security\_config.xml. |
| 1.13 | Тестирование настроек конфигурации безопасности  сети trust anchors  (NETWORK‑4) | В файле network\_security\_config.xml в теге <debug-overrides> присутствуют пользовательские <trust-anchors>, которые определяют <certificate src = "user">. |
| 1.14 | Наличие дополнительного канала связи  (NETWORK‑5) | В приложении можно авторизоваться с помощью электронной почты tut.by, как пользователь facebook, vk.com, google аккаунта.  Регистрация и восстановление аккаунта с помощью e-mail и номера телефона доступны во всех выше перечисленных способах. |
| 1.15 | Security provider  (NETWORK‑6) | Приложение использует и классы HttpsURLConnection, и классы TrustManager и SSLCertificateSocketFactory. Также используются методы для работы с криптографией.  Класс ProviderInstaller вызывается с помощью getClassLoader().loadClass("com.google.android.gms.common.security.ProviderInstallerImpl") и getMethod("insertProvider"), а не installIfNeeded, либо installIfNeededAsync.  Исключения перехвачены и правильно обработаны: вручную выбрасывается исключение GooglePlayServicesNotAvailableException, а в других классах, в методах проверки доступности GooglePlayServices, оно перехватывается. |
| 1.16 | Реализация аутентификации на стороне сервера  (AUTH-1) | Приложение не использует дополнительные факторы аутентификации.  При попытке пользователя участвовать в розыгрыше, сначала проводится проверка, что он вошел в систему, если не вошел, то участвовать в розыгрыше не получится, выведется соответствующе сообщение, т.е. проверка подлинности на сервере применяется последовательно. |
| 1.17 | Управление сеансом с отслеживанием  состояния  (AUTH-2) | Идентификатор сеанса генерируется с помощью UUID.randomUUID(), т.е. случайным образом, и не может быть легко угадан. Идентификаторы сеанса обмениваются через HTTPS.  Приложение не сохраняет идентификаторы сессии в постоянном хранилище. Сервер проверяет сеанс, когда пользователь пытается просмотреть итоги розыгрыша или поучаствовать в нем. Сеанс завершается на стороне сервера, и информация о сеансе удаляется в приложении после истечения времени. |
| 1.18 | Управление сеансом без отслеживания  состояния  (AUTH-3) | Для создания подписи используется алгоритм PBEwithHmacSHA1.  Пункты 1-4 выполняются, 5 – токены хранятся с помощью KeyStore – также выполняется. |
| 1.19 | Проверка OAuth 2  (AUTH-3) | Используется AccountManager. Все требования безопасности выполняются. |
| 1.20 | Тестирование выхода пользователя  (AUTH-4) | Приложение не выходит из системы автоматически. Если выйти из приложения, зайти снова, то участие в розыгрышах будет все еще доступно. Однако в настройках есть функция выхода из системы. Если вручную выйти из системы, то при попытке участвовать в розыгрыше пользователь будет перенаправлен на страницу входа. |
| 1.21 | Реализация парольной политики  (AUTH-5) | -минимальная длина пароля не менее 8 символов;  -максимальная длина пароля не более 20 символов;  -допустимы только латинские буквы, цифры или символы !@#$%^&\*(){}[],\_<> ;  -не происходит «обрезка» паролей, но форму регистрации нельзя отправить, пока вышеперечисленные правила не выполняются;  -есть строка состояния сложности пароля: в пароле должны присутствовать буква, цифра, специальный символ; если одна из групп отсутствует, то строка желтая, если все правила выполняются – зеленая, иначе – красная, и форму отправить нельзя. |
| 1.22 | Защита от перебора данных авторизации  (AUTH-6) | Количество попыток входа в аккаунт tut.by не ограничено. |
| 1.23 | Срок действия токена  (AUTH-8) | Время ожидания сессии не существует, тест не пройден. |
| 1.24 | Двухфакторная аутентификация  (AUTH-9, AUTH-10) | Приложение не использует механизм двухфакторной аутентификации, так как нет чувствительной деятельности. |
| 1.25 | Информирование пользователя о действиях с  его учетной записью  (AUTH-11) | Приложение отправляет уведомление на email, когда учетная запись используется на другом устройстве, сменен пароль или согласие с условиями при регистрации. Уведомление содержит метаинформацию об устройстве, дате и времени доступа, местоположении устройства в момент доступа. Об остальных действиях приложение не информирует пользователя, аудит истории действий, связанных с безопасностью, не ведет. |
| 1.26 | Локальная аутентификация в Android. Подтверждение учетных данных | Разблокированный ключ не используется во время работы приложения. Не используется setUserAuthenticationRequired. |
| 1.27 | Локальная аутентификация в Android. Биометрическая аутентификация | FingerprintManager.authenticate не вызывается, биометрическая аутентификация не используется. |
| 1.28 | Проверка использования стандартных криптографических алгоритмов (CRYPTO-2, CRYPTO-3, CRYPTO-4) | В приложении используются классы Cipher, MessageDigest, Key, PrivateKey, PublicKey, SecretKey; SecretKeySpec; пакеты java.security.\* и javax.crypto.\*.  SHA1PRNG используется в приложении.  Отсутствуют ключи, жестко закодированные в исходном коде. |
| 1.29 | Тестирование генерации случайных чисел (CRYPTO-6) | Для генерации случайных числе используются классы: java.security.SecureRandom, java.util.Random;  методы: java.lang.Math.random(), java.util.UUID. randomUUID(). |
| 1.30 | Управление ключами (CRYPTO-1, CRYPTO-5) | Использованные криптографические примитивы приведены в пункте 1.28. Для хранения ключей используется java.security.KeyStore. Для асимметричных ключей закрытый ключ используется исключительно для подписи, а открытый ключ только для шифрования. Симметричные ключи не используются повторно для нескольких целей. |