Android Application Vulnerabilities

Sun* Cyber Security Research

OWASP Top 10 mobile risks

- M1: Improper Platform Usage
- M2: Insecure Data Storage
- M3: Insecure Communication
- M4: Insecure Authentication
- M5: Insufficient Cryptography
- M6: Insecure Authorization
- M7: Client Code Quality
- M8: Code Tampering
- M9: Reverse Engineering
- M10: Extraneous Functionality

M1. Improper Platform Usage

1. Insecure version of OS installation allowed

Không cho phép cài đặt ứng dụng lên các phiên bản OS quá cũ do có nguy cơ mất an toàn từ chính OS.

Có thể quy định trong AndroidManifest.xml, phần minSDKversion. Tạm xác định min Sdk = 24 và min OS = Android 7.

2. Abusing Android components through IPC intents

Với Activity Hijacking (Intents): điều này xảy ra khi một Activity phụ thuộc vào Intent ngầm. Kẻ tấn công đăng ký một Intent Filter chính xác và kiểm soát nó, chạy một Activity độc hại thay cho cái ta mong muốn, vì vậy người dùng sẽ dùng một ứng dụng sai mà không hay biết.

Với Malicious Broadcast Injection (exported): Nếu một BroadcastReceiver được exported luôn tin tưởng các broadcast Intent gửi đến mà không kiểm tra các action, data nhận được, nó có thể nhận được những action không hợp lệ hoặc lệnh điều trên dữ liệu độc hại từ broadcast Intent và có thể chuyển đến cho các Services và Activities làm cho các Intent độc hại có thể lan tràn trong ứng dụng.

3. Default credentials on Application Server

Xác định thông tin đăng nhập mặc định trên Backend server

Ví dụ: Máy chủ ứng dụng Tomcat sử dụng tomcat/tomcat, admin/tomcat

4. Security misconfiguration on Server API

Security Misconfiguration ở API Server, có thể liên quan tới sử dụng các phiên bản cũ có lỗi, CORS, HSTS…

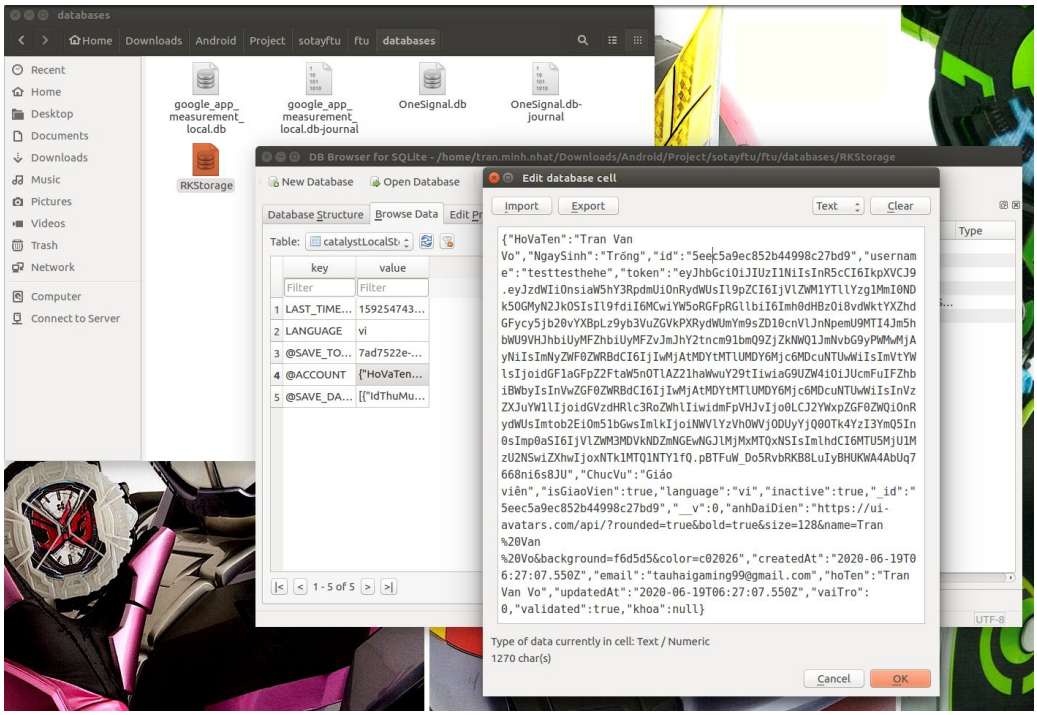
5. Minimum device security requirements absent

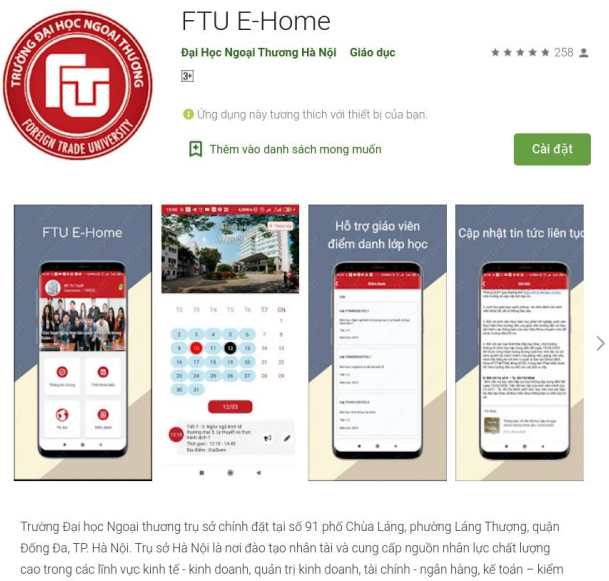
M2. Insecure Data Storage

1. Unencrypted credentials in databases.

Ứng dụng lưu trữ thông tin tài khoản không mã hoá ở các file database: *.sqlite, *.db

Khi cài ứng dụng trên thiết bị root, kẻ tấn công có thể tải các file database về thiết bị và đọc được toàn bộ thông tin (trong trường hợp DB không mã hóa)





2. Sensitive data storage in plain text.

Ứng dụng lưu trữ các thông tin nhạy cảm ở dạng plain-text, dễ dàng đọc được bởi các chương trình chỉnh sửa văn bản.

3. Insecure cookie storage.

Khi ứng dụng lưu trữ cookie không an toàn, kẻ tấn công có thể dùng các dữ liệu này để mạo danh phiên đăng nhập hợp lệ của người dùng.

4. Store credentials outside Sandbox.

File của ứng dụng được lưu trong SD Cards (external storage) chứa thông tin nhạy cảm và có thể được đọc và thay đổi bởi bất kỳ ứng dụng nào khác.

5. Unencrypted backup file.

Ứng dụng không mã hóa file backup, có thể để lộ thông tin khi Hacker đọc được.

6. Store encryption key locally.

Ứng dụng lưu trữ khóa giải mã không an toàn, hardcoded key, kẻ tấn công có thể dùng khóa này để giải mã các dữ liệu đã bị mã hoá.

7. Improper file permission or allow global permission.

Thiết lập các quyền không an toàn khiến cho các ứng dụng khác có thể đọc/ghi các tập tin nhạy cảm lưu bên trong ứng dụng.

8. The keyboard cache is disabled on text inputs that process sensitive data.

Bộ đệm bàn phím bị vô hiệu hóa trên các trường input liên quan dữ liệu nhạy cảm.

9. No sensitive data, such as passwords or pins, is exposed through the user interface.

Không có dữ liệu nhạy cảm, như mật khẩu hoặc mã pin, được hiển thị thông qua giao diện người dùng.

10. No sensitive data is included in backups generated by the mobile operating system.

Không có dữ liệu nhạy cảm được lưu trong các bản backup được tạo bởi hệ điều hành di động.

11. The app removes sensitive data from views when moved to the background.

Ứng dụng ẩn các dữ liệu nhạy cảm khỏi giao diện nhìn thấy được khi chuyển sang chế độ nền.

M3. Insecure Communication

1. Insecure Transport Layer Protocols

Dữ liệu được truyền đi không được bảo vệ tốt dẫn đến có thể bị nghe lén và lộ thông tin.

2. Use of Insecure and Deprecated algorithms

Các ứng dụng kết nối với server sử dụng thuật toán mã hóa yếu khiến dễ dàng được giải mã bởi kẻ xấu. Điều này gây nguy hiểm cho tính bảo mật của kênh truyền giữa ứng dụng và server

3. Use of Disabling certificate validation

Cho phép Pentest chặn SSL traffic mà không cần cài đặt Chứng chỉ (checkServerTrusted nobody)

4. SSL pinning Implementation - Lack of Certificate Inspection

Kiểm tra việc triển khai SSL Pinning. Các ứng dụng không kiểm tra chứng nhận được cung cấp bởi server và vô tình chấp nhận bất cứ chứng nhận nào từ server. Điều này dẫn đến ứng dụng có thể bị tấn công MITM thông qua SSL Proxy.

5. Third-party Data Transit on Unencrypted Channel

Ứng dụng gửi dữ liệu qua kênh truyền không được mã hóa của bên thứ ba dẫn đến dữ liệu không được bảo vệ.

M4. Insecure Authentication

1. Client Side Based Authentication Flaws - Bypass Authentication Schema

Ứng dụng cung cấp chức năng chứng thực offline để có thể tương tác với các hàm chức năng bên trong sau khi chứng thực thành công.

2. Session invalidation on Backend

Ứng dụng xóa phiên làm việc của người dùng ở thiết bị, nhưng không huỷ chúng ở phía máy chủ. Qua đó kẻ tấn công vẫn có thể thực hiện các thao tác như người dùng hợp lệ thông qua các công cụ giả lập yêu cầu.

3. Session Timeout Protection

Mỗi ứng dụng cần bảo vệ người dùng cuối bằng thời gian timeout phía máy chủ, để hạn chế kẻ tấn công dùng các phiên làm việc hiện có, thực hiện các hành vi như người dùng hợp lệ.

Sau một thời gian không hoạt động (không nhất thiết thiết bị phải tắt màn hình) thì người dùng phải authen lại. Độ dài tùy yêu cầu an ninh.

4. Cookie Rotation

Đảm bảo rằng việc đặt lại cookie được thực hiện chính xác khi có thay đổi trạng thái xác thực. (Anonymous <=> User, User A <=> User B, Timeout)

Đổi Cookie khi có sự thay đổi về authentication /authorize

5. Using Device Identifier as Session

Ứng dụng sử dụng các số định danh trên thiết bị như IMEI, UDID dùng làm định danh phiên làm việc của người dùng, điều này cho phép kẻ tấn công có thể mạo danh bất kì người dùng thông qua việc thay đổi các định danh phiên (session id) đó.

6. Weak Password Policy

Ứng dụng cho phép sử dụng các tài khoản với mật khẩu yếu, có thể dễ dàng cho kẻ tấn dò, đoán ra mật khẩu, truy xuất tài nguyên hoặc chiếm đoạt tài khoản người dùng.

Policy đặt password phải đủ mạnh. Việc kiểm tra độ an toàn password phải được thực hiện cả ở Client và Server.

7. No rate limit on Login

Remote Endpoint không có cơ chế chống lại việc gửi thông tin đăng nhập quá nhiều lần

8. Weak biometric authentication

Nếu ứng dụng có chức năng xác thực sinh trắc học (vân tay, khuôn mặt) thì không được sử dụng các API chỉ đơn giản trả về true/false. Các API này cần dựa trên việc mở khóa keychain/keystore

9. Not perform 2FA

Ứng dụng nên có chức năng 2FA, và việc này phải được thực hiện cả ở remote endpoint.

Các giao dịch nhạy cảm đòi hỏi phải có xác thực tăng cường.

M5. Insufficient Cryptography

1. Insecure Token Creation

Ứng dụng sử dụng cơ chế sinh Token không an toàn, không đủ độ dài, tính ngẫu nhiên, có thể đoán được, hoặc dùng thuật toán có thể bị phá vỡ.

2. Use of Insecure and/or Deprecated Algorithms

Ứng dụng sử dụng các thuật toán không an toàn hoặc đã bị lỗi thời, dễ dàng dò ra bản gốc của dữ liệu được mã hoá như: RC2, MD4, MD5, SHA1

3. Weak Custom Cryptography Methodology

Nhà phát triển tự xây dựng một thuật toán mã hoá, không sử dụng các thư viện mã hoá mạnh hỗ trợ sẵn, khiến cho kẻ tấn công có thể dễ dàng dịch ngược mã nguồn, phá vỡ phương pháp mã hoá của ứng dụng.

M6. Insecure Authorization

1. Missing function level access control / Bypass Authorization Schema

Ứng dụng hiện thực chức năng xác thực người dùng không chính xác, vì thế kẻ tấn công có thể thực hiện các hàm chức năng từ người dùng thấp quyền lên người dùng có quyền cao hơn, kể cả quyền quản trị.

2. Insecure Direct Object references

Các object sử dụng các tham chiếu để lấy giá trị, nhưng không phân quyền chính xác, dẫn tới việc lộ thông tin mà người dùng không có quyền truy cập.

M7. Client Code Quality

1. Content Providers: SQL Injection and Local File Inclusion

Với SQL Injection: Ứng dụng sử dụng SQLite để lưu giữ các dữ liệu của người dùng, nhưng không lọc các dữ liệu đầu vào khiến cho kẻ tấn công có thể thực hiện các câu truy vấn kèm theo để lấy thông tin trong DB.

Với LFI: Ứng dụng cung cấp chức năng xử lý tập tin, sử dụng thư viện với các thiết lập không an toàn, và không lọc các dữ liệu trước khi xử lý các Input đầu vào khiến cho kẻ tấn công có thể duyệt/đọc các thư mục khác hoặc tập tin hệ thống.

2. Insufficient WebView hardening / Javascript Injection / Malicious input from external sources

Ứng dụng cho phép thực thi Javascript nhưng không lọc dữ liệu đầu vào, khiến cho kẻ tấn công có thể lợi dụng để thực thi các đoạn Javascript độc hại nhằm đánh cắp cookie/session của người dùng.

Tất cả input có nguồn gốc từ ngoài ứng dụng và từ người dùng đều cần được xác thực và kiểm tra kỹ để tránh các lỗi về injection và XSS.

3. XML Injection

Ứng dụng cho phép thực thi Javascript nhưng không lọc dữ liệu đầu vào, khiến cho kẻ tấn công có thể lợi dụng để thực thi các đoạn Javascript độc hại nhằm đánh cắp cookie/session của người dùng.

Tất cả input có nguồn gốc từ ngoài ứng dụng và từ người dùng đều cần được xác thực và kiểm tra kỹ để tránh các lỗi về injection và XSS.

4. Abusing URL schemes or Deeplinks

Ứng dụng sử dụng deep link được define trong source code để khi người dùng ấn vào URI thì OS tự động gọi đến App hoặc Activity trong App.

Khi click một URl, hệ thống Android sẽ làm từng bước sau, theo tuần tự, cho đến khi thành công:

1. Mở ứng dụng mặc định theo setting của người dùng có thể xử lý URL, nếu được chỉ định
2. Mở ứng dụng duy nhất có thể xử lý URL
3. Cho phép người dùng chọn ứng dụng từ dialog

5. Broadcast Thief

Một implicit Intent công khai mà không được bảo vệ với các loại quyền "Signature" hoặc "SignatureOrSystem".

Một BroadcastReceiver độc hại có thể đọc được thông tin tất cả các broadcast công khai của tất cả các ứng dụng.

Ngoài ra, Hacker có thể phát động DOS dựa trên các ordered broadcasts bằng cách ngăn không cho gói broadcast tới được các Receivers cần đến của ứng dụng dựa trên độ ưu tiên của chúng.

6. Build in wrong mode

Ứng dụng chưa được build ở chế độ phát hành, thiếu các cài đặt phù hợp cho bản dựng phát hành (ví dụ: không thể chạy ở chế độ debug; không lộ các thông tin quan trọng, các thông tin debug,... ra Log; Xử lý hết các exception có thể lường trước;...).

7. 3rd party services may be exploited by known vulnerabilities

Những thành phần của bên thứ 3 như libraries, framework, chưa phải phiên bản mới nhất và có thể bị khai thác bằng các lỗ hổng đã được công bố.

8. Error handling logic in security controls

Nếu có lỗi trong quá trình kiểm soát an ninh an toàn thì mặc định phải từ chối truy cập

Phải có xử lý lỗi một cách có kiểm soát

M8. Code Tampering

1. Unauthorized Code Modification

Ứng dụng không có các cơ chế phát hiện bản thân chúng bị thay đổi bởi các kỹ thuật patching nên có thể bị chèn mã độc và các cơ chế bảo mật dễ dàng bị vượt qua.

Có cơ chế kiểm tra tính toàn vẹn các file code bằng cách hash source code smali. Đảm bảo khi code đổi thì hash values cũng sẽ bị đổi.

2. Runtime Manipulation

Ứng dụng không có cơ chế bảo vệ trước các công cụ hook, không phát hiện được code hoặc dữ liệu đã bị thay đổi trong quá trình chạy.

M9. Reverse Engineering

1. Lack of Code Obfuscation

Khi release sản phẩm, nhà phát triển không sử dụng các kỹ thuật Code Obfuscation, khiến cho kẻ tấn công có thể dễ dàng dịch ngược mã nguồn, thấy rõ các tên hàm (function) và dễ dàng đoán được chính xác các chức năng bên trong, vẽ lại luồng thực thi ứng dụng.

2. Lack of Root Detection

Rủi ro khi sử dụng các thiết bị đã Root. Không có cơ chế phát hiện, ngăn chặn việc chạy trên thiết bị root.

3. Lack of Emulator Detection

Ứng dụng không có cơ chế phát hiện và phản hồi khi được chạy trên các chương trình ảo hóa, giả lập.

4. Device Binding

Ứng dụng triển khai chức năng ràng buộc với thiết bị (device binding) bằng cách sử dụng chứng chỉ nhận dạng được tạo thành từ các đặc điểm duy nhất của mỗi thiết bị.

Cookie / data đó chỉ chạy với thiết bị được sử dụng từ đầu, nếu được chuyển đổi sang thiết bị khác thì ko được. Dùng UUID thiết bị để kiểm tra.

M10. Extraneous Functionality

1. Lack of Anti-Debugging Method

Nhà phát triển không dùng các kỹ thuật chống gỡ lỗi (debug), khiến cho kẻ tấn công có thể thực hiện các kỹ thuật phân tích dữ liệu trong lúc thực thi của ứng dụng, từ đó có thể tìm ra các key mã hoá hoặc các giá trị biến quan trọng của ứng dụng, từ đó phục vụ cho các phương pháp khai thác khác

2. Passwords/ Connection String disclosure

Identify sensitive information (Credential) between mobile and API

Hidden and Unscrutinised functionalities

Các chức năng ẩn không có trên UX/UI nhưng vẫn tồn tại, dẫn đến kẻ tấn công sử dụng chúng và biết được các thông tin, back-end test, demo, staging, hay UAT environments.