

行動應用程式自動化資安合規檢測系統之設計與 546 支上架 App 的實證評估

Design of an Automated Security-Compliance Assessment System for Android Applications and an Empirical Evaluation on 546 Published Apps

呂凱崑（國立臺灣師範大學資訊工程學系） 指導教授：紀博文 教授 專題期間：2025 年 9 月至 2026 年 9 月

本文為大學部資訊專題研究成果之技術摘要。文中真實 App 皆以產業類別代稱，漏洞細節與機密值不公開；研究全程未以任何取得之金鑰或憑證進行連線測試，並依負責任揭露原則處理。

摘要

行動應用程式（App）承載個人與金融資料，臺灣《行動應用 App 基本資安檢測基準 V4.0》（下稱 V4.0）訂有 49 項檢測項目，是金融、政府與醫療 App 送測的常見依據，但檢測仍高度仰賴人工。本專題設計並實作 APKSec Pro：以 Android 安裝檔（APK）為輸入、依 V4.0 逐項判定的自動化合規檢測系統。方法上的貢獻有三：（1）為 49 項條文逐條建立「工具能證明什麼、證據是什麼、證不出時人工做什麼」的可辯護合約，以單一模式、四狀態（通過／不通過／不適用／需人工附佐證）輸出，判定與佐證分離；（2）採召回優先與單邊確定性原則，只在證據可定位、可重放時輸出判定，可自動判定 36 項（其中 21 項為只證不通過的單邊判定）；（3）以多層驗證體系評估系統有效性，包括 1,105 項自動化測試、公開漏洞交叉驗證，以及對 546 支唯一 APK 建立獨立逆向 ground truth 並與系統判定逐條文比對。

實證結果：對 400 支真實上架的臺灣 App（銀行、券商、政府、醫院、交通與生活）統一檢測後，368 支至少有一項不通過；546 支全數比對的混淆矩陣中，獨立引擎高信心判不通過的 1,121 筆系統全數判不通過，高信心漏報 0 筆、疑誤報 0 筆；584 筆經反編譯確認的 App 自有漏洞無一筆被判為通過。研究另以靜態方法確認 64 支 App 存在可直接回報的高危問題（29 支內嵌可取出的雲端服務或伺服器私鑰、40 支主要連線停用 TLS 驗證）。

關鍵詞：Android 安全、合規檢測、靜態分析、動態分析、召回優先、負責任揭露

1 研究問題與定位

現行 V4.0 檢測需要實驗室投入大量人力，開發商在年度送測之間缺乏低成本的自我檢查方式；既有開源工具（MobSF、QARK）雖能偵測一般性 Android 弱點，卻不對應 V4.0 條文，也不區分「工具能證明的」與「本質上需人工的」項目，容易產生自相矛盾的合規假象。本專題提出四個研究問題：

- RQ1 如何把 49 項條文轉成可辯護的判定合約，使自動判定不在弱證據下給出誤導性結論？
- RQ2 以召回優先為原則的靜態分析，在真實上架 App 上是否真的不漏報？誤報是否可由人工快速撤銷？
- RQ3 系統能否在真實世界找到可辯護、值得通報的高危漏洞？研究者應如何在不觸法的前提下確認漏洞為真？
- RQ4 如何讓非資安背景者「丟一個 APK 就得到看得懂的報告」，並誠實呈現工具界線？

系統定位為「開發商平時自我檢查的最嚴格審查工具」，而非取代實驗室的送測認證。

2 方法

2.1 判定合約：單一模式、四狀態、判定與佐證分離

每一個檢測項目的結果只會是四種之一，可辯護的操作性定義如下。不通過必須有可定位的證據（具體字串、類別名、Manifest 屬性、設定檔內容、動態事件或封包紀錄），共現推論不算證據；通過須在明確宣告的檢測範圍內以可重現方法查無違反，「查得防護程式碼存在」只算佐證，因為存在不等於有效；不適用只依外部框架 API 引用缺席或 Manifest 組態自動判定；只能證明一側者採單邊確定性，可證明側輸出判定、另一側交人工。召回優先原則的意涵是：寧可多報一條可由人工快速撤銷的不通過，也不容許任何真實漏洞被吞成通過；命中類別屬第三方程式庫時仍輸出不通過並標注套件歸屬，只有能確定不構成違反的型態才降為線索。49 項合約以可讀文件與機器可讀對應表雙軌維護，由自動化測試鎖定同步。

表 1 V4.0 檢測項目的自動化涵蓋（v2.4.1；由機器可讀對應表逐項計算並以測試鎖定）

應檢測項目數	L1 25 項、L2 31 項、L3 39 項；高安全性 App 之 F 類加測 9 項（合計 49 項）
可自動判定	36 項，其中 21 項為只能證明不通過的單邊判定（通過側交人工）
需人工	13 項：需真實交易、需查官網或聯絡開發者、需長期觀察、需人工綜合認定、防護機制有效性不可靜態證明

2.2 系統架構與檢測引擎

系統採四層設計（圖 1）。證據蒐集層自四個來源取得證據：以 Androguard 為基礎的靜態分析（Manifest、網路安全組態、DEX 位元碼與字串）、以 Frida 為基礎的動態分析（模擬器上掛鉤 API、logcat 與被動側錄）、第三方函式庫 CVE 比對與伺服器憑證線上探測，以及選配的 LLM 輔助判讀；合規引擎依逐條合約把證據轉成判定；判定層只輸出四種狀態；輸出層產生 PDF、互動式 HTML、JSON 報告與需人工工作單。

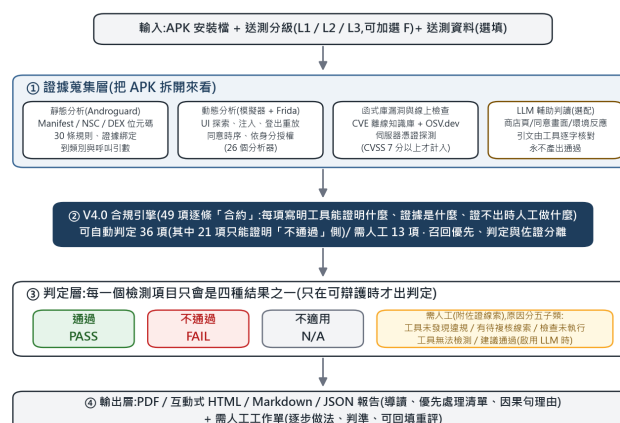


圖 1 APKSec Pro 系統整體架構（證據蒐集層、合規引擎、判定層、輸出層）

靜態分析含 8 個模組、32 條規則，證據綁定到類別與呼叫引數：WebView 規則讀位元碼確認 setAllowUniversalAccessFromFileURLs 的引數為 true 才判不通過；TLS 規則讀 checkServerTrusted 方法本體，位元碼為空實作或 HostnameVerifier 恆回傳 true 才判不通過。動態分析含 26 個分析器（自動 UI 探

索、注入機器人、登出後重放舊 token、殘留與寫檔掃描、畫面擷取檢查、同意時序觀測等)。LLM 輔助判讀 12 條規則只處理「靠讀文字就能判」的項目，且判定由工具依結構化認定推導、引文須逐字出現在工具存下的證據快照、送出前遮罩測試帳號並設費用上限；LLM 永不產出通過。

2.3 驗證體系

判定邏輯變更遵守「先改合約、經確認、再改程式、跑全套驗證」的治理流程。驗證層次依序為：pytest 自動化測試 (v2.4.1 為 1,105 項)；多 APK 回歸；公開漏洞不漏報交叉驗證 (DIVA、InsecureBankv2、AndroGoat 等的已知漏洞不得被判通過或不適用)；每條可判不通過的規則至少一支真實正樣本與一支反樣本；自建可重現的 demo APK；以及最外層的 546 支全數獨立逆向比對，每次判定碼變更即以當前版本重掃全部語料、重產混淆矩陣，翻轉逐筆歸因。

3 實證研究設計

- 語料：於模擬器經 Google Play 安裝 408 支臺灣 App，排除 8 支系統元件後為 400 支真實第三方 App；連同前期語料共 546 支唯一 APK。400 支以送測分級 L3（最嚴格範圍）統一靜態檢測。
- 獨立逆向 ground truth：(a) 獨立判定引擎只讀 APK 原文、不呼叫系統規則層，對 11 條可靜態獨立判定的條文逐支判定並分「高信心／候選」兩級；(b) 以 jadx 反編譯逐支審查 App 自有程式碼，每筆發現由「歸屬／型態」與「可達性／可利用性」兩個對抗式視角覆核，須附可定位的反編譯證據。此層的規模 (546 支、584 筆) 以本人設計的覆核流程驅動多個 AI 代理平行執行，本人逐批驗收、抽驗並裁決爭議，A1 級私鑰另逐檔以密碼學函式庫載入確認。
- 倫理界線：所有金鑰與權杖只辨識、定位、遮蔽記錄，不測試有效性、不用於連線；結果供教學與負責任揭露使用。

4 結果

4.1 400 支真實 App 的檢測總覽

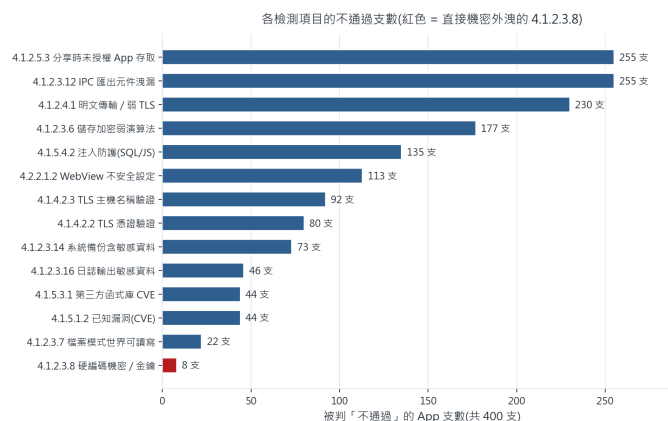


圖 2 各檢測項目的不通過支數 (400 支，送測分級 L3，v2.2.0 時點)

368 支至少有一項不通過、32 支零不通過，合計 1,574 條不通過旗標（每支平均 3.9 條；各項目的不通過支數見圖 2）。最常見者為匯出元件、明文傳輸或弱 TLS、儲存加密弱演算法與注入防護，多屬強化不足或第三方程式庫殘留；零不通過的 32 支多為防護完善的銀行與金融 App，反證系統具鑑別力。一條不通過是一個待人工複核的旗標，不等於一個可被入侵的洞。

4.2 546 支全數獨立逆向比對

表 2 546 支全數比對的混淆矩陣 (獨立判定引擎 vs 系統; v2.4.1 判定碼, 11 條可靜態獨立判定之條文)

獨立判定引擎 \ 系統	不通過	需人工	通過 / 不適用
高信心不通過 (1,121 筆)	1,121	0	0
候選不通過 (僅字串層級訊號)	—	—	2 (覆核後皆非漏洞; 原第 3 筆 LibreTube WebView 橋接於 v2.4.1 修正 後改列線索)

第二層以 584 筆經反編譯確認的 App 自有漏洞 (294 支) 對照系統判定：自動判不通過 322 筆、連同線索交人工 262 筆、判通過 0 筆，召回優先的底線「真漏洞不得被吞成通過」在全數語料上成立；交人工者集中於 WebView 設定、無固定格式的硬編碼機密與注入三類，皆附線索供檢測員複核。全數比對亦回饋至判定邏輯：每次修改後重掃 546 支，26,754 個判定中的翻轉逐筆歸因、無非預期變動；加殼 App (47 支) 於靜態只看得到殼時不給出通過。

4.3 可回報的高危發現 (匿名)

584 筆已確認漏洞依攻擊者所需前置條件分四級，A 級 (可直接回報) 70 筆、64 支：

- A1 可直接取出的機密 (30 筆、29 支)：安裝檔內嵌雲端服務的服務帳戶私鑰、TLS 伺服器私鑰或管理員金鑰。代表型態包括雲端資料庫的管理員金鑰、以私鑰與憑證的 modulus 相等作數學確認的 TLS 伺服器私鑰，以及由公信憑證機構簽發的伺服器私鑰連同寫死的金鑰庫密碼。
- A2 主要連線停用 TLS 驗證 (40 筆、40 支)：App 自行實作信任所有憑證的 TrustManager 或恆回傳 true 的 HostnameVerifier，並掛在登入、下單、掛號、繳費等主要連線；分佈於券商 15 支 (多來自同一代工下單元件)、銀行 4 支、醫療 6 支、交通與政府 5 支、電信影音零售 8 支。

每一筆皆以靜態方法確認為真 (結構自證、憑證數學配對、反編譯定位)，未以任何金鑰或憑證連線測試；機密值不公開，通報對象與時程由指導教授與研究者依 ISO/IEC 29147 負責揭露原則決定。



圖 3 系統報告的逐項判定頁 (以自建示範 App 產出 ; 4.1.2.3.8 硬編碼機密，理由寫成「因為 / 代表 / 所以 / 怎麼修」，私鑰本體已遮蔽)

5 未來工作

以 LLM 讀取「可疑字串＋使用位置」的反編譯前後文並維持引文逐字核對，擴大自動判定之涵蓋；以可達性與污點分析縮減需人工項目；以動態分析與執行期 DEX 擷取涵蓋加殼 App；對 400 支語料每季重掃，建立臺灣上架 App 合規狀態的時間軸；依負責任揭露原則進行廠商通報並追蹤修補。

6 本人角色與可驗證性

研究問題、49 項判定合約與判定原則、系統架構、驗證體系的設計，以及每一次判定邏輯變更的裁決與最終驗收，皆由本人完成；指導教授指導研究方向與判定原則。程式實作採「合約先行」的工作流程：本人撰寫規格與合約，由 AI 程式助理（Claude Code）在本人的規格、審查與測試門檻下協作產出程式碼，本人負責審閱、驗證與裁決。程式碼、合約、測試與遮罩後的 ground truth 鏡像皆入版本控制（v2.4.1，1,105 項測試）；倉庫因含真實 App 之漏洞紀錄暫為私有，可應審查需要提供唯讀存取。真實 App 語料與未遮罩的漏洞證據不入庫。技術棧：Python 3.10+、Androguard（靜態）、Frida 與 ADB（動態）、jadx（反編譯覆核）、fpdf2 與 HTML（報告）、tkinter（GUI）、OSV.dev（CVE）、Claude API（選配 LLM 判讀）。

AI 工具使用聲明：本研究之程式實作與第 3 節所載之反編譯覆核以 AI 代理協作完成，其產出均在本人訂定的規格、測試與驗證門檻下審閱並裁決；本文撰寫過程使用生成式 AI 工具協助文字整理與排版，內容、數據與觀點皆出自本人之研究與成果報告，由本人逐句核對並負全責。