SEAndroid 行動裝置管理系統之設計與實作
Design and implementation of SEAndroid device management system

李孟樵
國立高雄師範大學 軟體工程與管理學系
Meng-Chiao Lee, Department of Software Enginerring and Management, National Kaohsiung Normal University
Email: chiomle@gmail.com

楊中皇
國立高雄師範大學 軟體工程與管理學系
Chung-Huang Yang, Department of Software Enginerring and Management, National Kaohsiung Normal University
Email: chyang@nknu.edu.tw

摘 要

現今行動裝置儲存大量私人與企業資料，因此維護手機安全非常重要。本研究實作一個針對 Android 的遠端安全管理系統，運用包含 SEAndroid 之 AOSP (Android Open Source Project)開發手機端系統，並搭配具有「sepolicy 更新」、「權限控管」、「應用程式管理」三項功能模組之伺服器。本系統遠端修改 Android 設定，於核心層透過分析與更新 sepolicy 降低資料外洩風險，在應用層透過調整應用程式危險權限，與分析裝置上是否有雜湊函數不符之應用程式，並搭配主動移除或重安裝應用程式，減少惡意軟體攻擊，提升系統安全。主要貢獻為，實作遠端修改 sepolicy 與整合前述功能，並支援企業 HYOD (Here is your own device)策略下的行動裝置管理系統，維護行動裝置安全。

關鍵字：SEAndroid、Security Policy、Permission Management、Android Open Source Project、Mobile Device Management

一、緒論

智慧型手機全球用戶預估在 2016 年會高達 20 億[1]。在 2015 年第四季各行動裝置作業系統中 Android 市佔率高達百分之 80.7[2]位居第一。因行動裝置普及與廣泛使用在生活中，存放大量私人與企業資料，導致針對 Android 的惡意攻擊大量增加[3]。由以上可知，Android 行動裝置安全是一個重要議題。

基於各式行動裝置安全威脅，本研究蒐集過去研究，進行整合與加強。核心層 SEAndroid 相關研究，主要以 sepolicy 內容分析為主[4][5]，鮮少因應狀況即時作出永久性的 sepolicy 更新，故本研究提供 sepolicy 分析、SEAndroid 錯誤訊息蒐集與遠端即時永久性修改。應用層方面，有透過遠端方式修改應用程式權限保護使用者的隱私資訊[6]，與透過對裝置內所安裝的應用程式進行雜湊值運算，確保應用程式未遭竄改或版本錯誤，以降低安全風險[7]。上述應用層研究皆注重少數安全問題修正，本研究將其擴充與整合，在應用程式發生竄改或版本錯誤風險時，管理者可主動移除，或重新安裝版本正確之應用程式，在特定狀態下可主動修改應用程式權限，在需要時可隨時控制裝置上整體應用程式權限，降低惡意軟體攻擊可能。結合上述三項功能，提供更完整的 Android 遠端安全加強系統。

綜合以上說明，本研究整合三項相關功能，實作一套有效增強 Android 系統核心層、應用層安全的遠端管理系統，支援企業 HYOD 策略下的行動裝置管理系統[8]。主針對 Google 原生設備 Nexus 7 II 進行開發，使用 Android 6.0 之原始碼為作業系統基礎，透過修改與添加遠端管理之客戶端應用程式，再搭配修改後的推播伺服器 AndroidPN (Android Push Notification) [9]作為遠端管理伺服器，達到遠端控管行動裝置之效果。目的在於透過遠端方式修正各項安全問題，降低資訊安全風險。

二、文獻探討

2.1 SEAndroid

SEAndroid 是將 SELinux 安全模組做適度修改後，加入 Android 中，此計畫由 Security Enhancements for Android project 負責，主要是在原先 coarse granularity 的 DAC (Discretionary Access Control)機制之外，加上 fine-grained 的 MAC (Mandatory Access Control)機制，架構與運作流程如圖 1 [10]。MAC 機制能防止特權提升攻擊與應

*本研究接受科技部編號：MOST 102-2221-E-017-003-MY3 經費補助

用程式漏洞[11]。

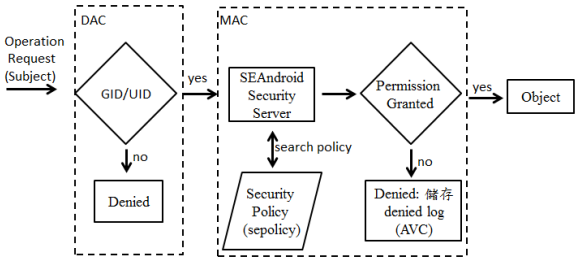

圖 1 SEAndroid 運作流程

在運作過程中參考的 security policy 檔案是由使用者空間映射到核心內，此判斷採用白名單方式，需要透過專用語言撰寫。在行動裝置中的 security policy 檔是由原始碼內數個.te (type enforcement)檔組合而成，檔案內定義各項要求狀態，此語言格式如下：

rule_name source_type target_type:class perm_set;

其中 rule_name 有 allow、dontaudit、auditallow 和 neverallow 規則關鍵字。source_type 辨識 subject，target_type 辨識 object，兩者皆可以是一個或多個先前定義的 type 或 attribute。class 是 object 的分類，如檔案或目錄等分類。perm_set 用來指定來 source_type 對 target_type 擁有的 permissions。

因判斷需要，會將系統上不論是進程或文件都會開機或創建時，被分配一個 security context（也稱作 security label 或 label），由三個主要部分組成格式如下：

user:role:type[:range]

可以在 android 內使用 ps -Z 或 ls -Z 查看進程或是檔案的 security context，其中 type 部分就是前段所提到識別 subject 與 object 時的識別文字，也可將多個 type 分成一個群組（稱為 attribute），直接對一個群組撰寫 security policy。

如果系統拒絕訪問時，會將此次行為紀錄在核心日誌檔中，於 Android 上可透過 dmesg 指令查看，關鍵字為 AVC，此訊息內容可以看出發生時間、source context、target context 等資訊，可作為分析之用。

有關 SEAndroid 運作相關檔案在原始碼內 /external/sepolicy 目錄下，此目錄在 Android 6.0 原始碼內有 98 個檔案，其中.te 檔有 72 個，此目錄下檔案，編譯後會加入 boot.img 中，在裝置內 SEAndroid 運作相關檔案位置與功能說明如表 1[12]。

表 1 SEAndroid 運作相關檔案位置與功能說明

裝置內檔案	內容與用途
/sepolicy	原始碼內.te 檔結合後的二進制檔，用於運行時判斷。
/file_contexts	使用在 filesystem 的 label 分配。
/property_contexts	使用在系統 property 的 label 分配。
/seapp_contexts	應用程式 process 的 label 分配，主要透過應用程式 UID 與應用程式證書之 seinfo 作判斷。
/service_contexts	分派各項服務 label。
/system/etc/security/mac_permission.xml	依據應用程式證書分派 seinfo 值，此 seinfo 值用於 seapp_context 分配應用程式 label 時的參數。

2.2 Android 架構

Android 作業系統是基於 Linux 核心所開發的而成，此作業系統由 Google 主導的開放手機聯盟(Open Handset Alliance; OHA)所成立的 Android 開放原始碼專案 (Android Open Source Project; AOSP)維護與發展[13]。任何廠商或開發者可依據此作為主軸，於其之上發展各自的行動裝置作業系統。

此作業系統架構如圖 2[14]：最底層是 Linux 核心層，於 Android4.3 版本起加入 SELinux，依序往上有，使用 C 與 C++語言撰寫的原生函式庫、虛擬機器、系統服務管理員、Android 和 Java API 和最上層的應用程式。

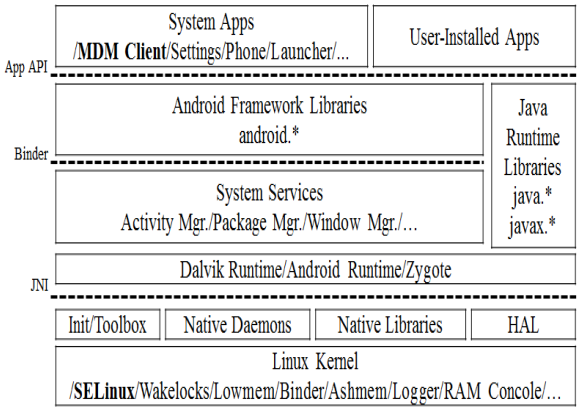

圖 2 Android 作業系統架構圖

2.3 Android Push Notification

AndroidPN (Android Push Notification)是一套針對 Android 的開放原始碼軟體推播伺服器，主要使用 JSP 撰寫而成，能在 tomcat 伺服器上執行，使用 XMPP (Extensible Messaging and Presence Protocol)傳輸協定傳送文字訊息至客戶端。本研究伺服器以此軟體為基礎，將其修改並架設成遠端管理伺服器，提供管理者遠端遙控行動裝置，藉此達到管理之效果。

2.4 行動裝置管理系統

行動裝置管理系統 (Mobile device management; MDM)是提供企業管理內部行動裝置的管理系統，主要是以遠端控制方式管理，透過此系統可以有效保護裝置內與企業有關資料，提升裝置安全性。企業導入行動裝置管理系統會使用不同策略，主要有四項如下：

- HYOD (Here is your own device)：行動裝置由企業購買，提供員工公務使用，企業可於裝置內是先安裝軟體或修改系統，為各項策略中企業掌控度最高。
- CYOD (Choose your own device)：企業預先選定一系列設備，提供員工自行選擇使用。
- BYOD (Bring your own device)：員工使用自己的設備於工作上，企業有對此類設備管理。
- OYOD (On your own device)：員工使用自己設備於工作上，企業無任何限制，此策略安全風險最高。

本研究以 HYOD 策略為主要發展基礎，但在員工使用設備規則上較為寬鬆，允許員工私人使用，在此情境下企業可對行動裝置種類有所限制，並可在裝置內預先安裝管理軟體，使企業能有更簡單與更高控制權的方法去管理行動裝置[15]。

三、系統架構與開發

3.1 系統架構

本研究實作 Android 遠端安全功能整合系統。系統由伺服器與客戶端（行動裝置應用程式）兩部分組成，系統架構如圖 3 所示。伺服器為包含「sepolicy 更新」、「權限控管」、「應用程式管理」三項功能模組之推播伺服器 (AndroidPN)；行動裝置主要由 Google 提供之 Android 6.0 原始碼修改並加入客戶端應用程式。運作方式為伺服器端推播指令觸發客戶端應用程式，應用程式上實作四項功能。各功能模組說明如下：

- 「sepolicy 更新」管理者依據行動裝置回傳的 SEAndroid 錯誤訊息 AVC 與 sepolicy 分析檔，瞭解裝置 policy 設定問題，依據情境制定不同的 policy，並將其回傳與套用，在不重新啟動裝置狀態下，即時提升核心層保護。
- 「權限控管」透過關閉裝置上應用程式權限，達到保護使用者隱私資料之目標。
- 「應用程式管理」對裝置內 APK 檔進行雜湊值運算，也對伺服器端，由管理者審核通過之 APK 進行雜湊值運算，將此兩者比對，確認 APK 檔無版本錯誤或遭到竄改，確保裝置上 APK 正確無誤，如果比對錯誤，也可即時解除或重新安裝該 APK。

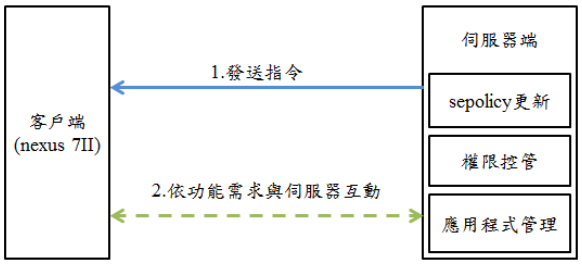

圖 3 系統架構圖

3.2 系統開發

本系統伺服器採用 Ubuntu 為作業系統，搭配推播訊息之開源軟體 AndroidPN 修改而成，詳細伺服器開發與測試環境如表 2；客戶端作業系統為 Android 6.0 原始碼，搭配由 Java 程式語言以及 Google Android APIs 開發之客戶端應用程式，詳細客戶端開發與測試環境如表 3。

功能實作上，「sepolicy 更新」更新 sepolicy 透過調整 sepolicy 檔與使用指令達到修改 sepolicy 功能，在分析時分為兩部分，第一為錯誤訊息分析調用 dmesg 擷取系統核心日誌檔之 SEAndroid 錯誤 log 訊息 AVC，第二為分析 sepolicy 內容，使用開源程式碼獲得分析訊息。「權限控管」由 Android 指令 pm (Package Manager)調整應用程式權限。「應用程式管理」實作分為兩部分，伺服器使用 openssl 計算管理者核准之 APK 計算 SHA-256 雜湊值，客戶端則是利用 MessageDigest 類別計算裝置內已安裝 APK 檔進行 SHA-256 雜湊值計算，並將結果傳至伺服器端，由伺服器端對兩個雜湊值進行比對。

表 2 伺服器端開發與測試環境

用途	套件名稱
作業系統	Linux (Ubuntu 14.04LTS)
整合開發環境	Eclipse
程式語言	Java、HTML、JavaServer Pages、JavaScript
網頁伺服器	Apache tomcat
資料庫	MySQL

表 3 客戶端開發與測試環境

用途	套件名稱
作業系統	Linux (Ubuntu 14.04LTS)
整合開發環境	Eclipse Android Studio
套件	Google Android APIs
程式語言	Java C 語言
測試環境	Nexus 7 II (Google 提供之 Android 6.0 原始碼)

四、系統功能

4.1 客戶端

本研究基於企業 MDM 的 HYOD 策略，將行動裝置派發給員工之前將 Android 6.0 原始碼做適度修改與加入 MDM 客戶端應用程式，將此修改之原始碼編譯成 ROM 再刷入行動裝置中。客戶端應用程式提供與伺服器連線資訊，如圖 4 所示。當伺服器傳輸指令修改設定時，客戶端在收到指令時會即時完成設定，並顯示通知告知裝置使用者本次修改訊息，如圖 5。

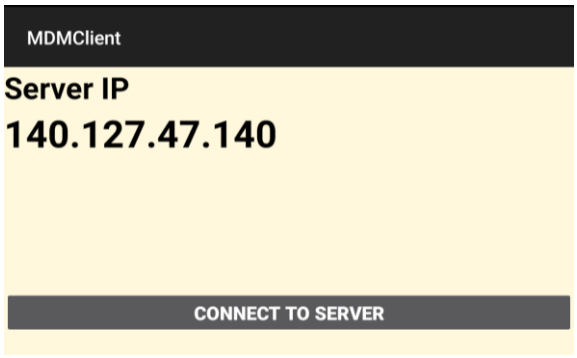

圖 4 客戶端使用介面

圖 5 客戶端顯示通知

4.2 伺服器端

本研究之行動裝置管理系統伺服器端，在「裝置管理」頁面裡顯示受本伺服器管理的行動裝置，在清單中列出目前裝置是否與伺服器連線、管理者對裝置的命名、連線 ID、該設備註冊日期，伺服器管理介面如圖 6。

圖 6 伺服器管理介面

4.2.1 sepolicy 更新

在「sepolicy 更新」介面，在選擇設備後，點選更新按鈕即可即時傳送預先準備之 sepolicy 檔案到裝置上，並即時套用新版本 sepolicy，或選擇 dmesg AVC log 可擷取紀錄於裝置 SEAndroid 之核心拒絕訊息並將之存送至伺服器，並顯示於下方表格內，此兩項功能介面如圖 7，或選擇 seinfo 可擷取裝置內目前透用的 sepolicy 檔案的分析訊息，並顯示在下方表格中，如圖 8。管理者可透過兩項分析功能，分析目前裝置內 SEAndroid 執行狀態有無問題，與 sepolicy 檔案現狀是否符合管理者期望，並可以透過更新功能即時修復或增強裝置安全性。

圖 7 sepolicy 更新與 AVC log 之管理介面

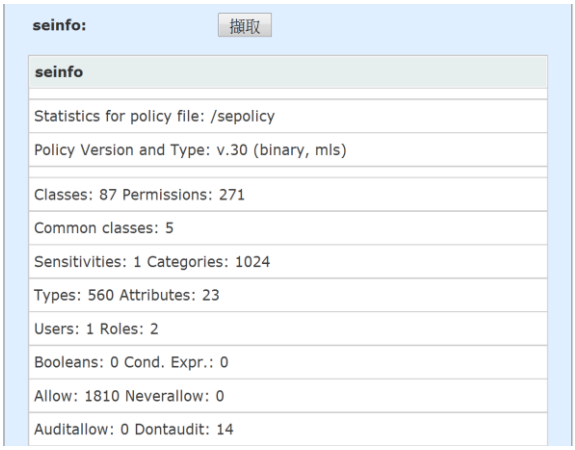

圖 8 seinfo 顯示畫面

4.2.2 權限控管

「權限控管」介面顯示可管理應用程式權限選項如圖 9，選定欲管理裝置後可透過 ON、OFF 按鈕，開啟或關閉裝置上所有第三方應用程式指定權限。

圖 9 應用程式權限管理之管理介面

4.2.3 應用程式管理

功能「應用程式管理」管理介面顯示如圖 10 所示，選定裝置後按下計算，可即時計算裝置上所安裝的第三方應用程式 APK 檔 SHA-256 計算，將其計算結果之 hash 值傳送至伺服器上，與伺服器預先準備之標準 APK hash 值進行比對，當 hash 值不同時會顯示為紅色，標示為紅色的應用程式可在管理欄位內選擇解除安裝或重新安裝功能，進行錯誤修正。

圖 10 伺服器管理介面

4.3 相關研究功能比較

在過去 MDM 研究中，有關注能在多種行動裝置作業系統中運作和客戶端與伺服端驗證機制[16]，也有主要目標為應用程式正確性驗證與產生安全檢測報告 [7]，而本研究基於過去研究找出較少被關注的核心層安全加強機制，進行設計與實作。因主要目標不同，在實作時有不同功能限制和因應目標的系統特色，上述研究中主要功能比較如表 4 相關研究功能比較所示。

五、結論

本研究貢獻在於以企業 HYOD 策略下的行動裝置管理系統為基礎，建設一套支援此環境的 Android 核心層與應用層的遠端管理系統，本系統能協助管理人員針對行動裝置安全問題做出即時回應，提升企業內部行動裝置安全，主要針對核心層 SEAndroid 的 security policy 進行分析，與線上即時更新，分析目前狀態資訊，由伺服器發送新的 security policy，在行動裝置上即時套用，透過此功能可從核心限制各 process 權限，即時補強弱點，應用層方面以管理第三方應用程式為主，可透過權限控管功能與 APK 驗證功能，限制權限使其在最小範圍內運作，並驗證目前已安裝之應用程式是否版本與內容正確，在正確且最小範圍運作，在 APK 驗證錯誤時可立即移除或重安裝應用程式，確保行動裝置內企業與私人資料不外洩。未來將對 SEAndroid 進行更深入探討與研究，提供更豐富與細緻的 security policy 管理功能。

表 4 相關研究功能比較

相關研究 / 功能項目	Security Policy based Device Management for Supporting Various Mobile OS	Android 平台智慧型行動裝置管理系統設計與實現	本研究
應用程式管理	無	驗證與解除安裝 APK 檔	驗證、解除安裝和重新安裝 APK 檔
裝置功能控管	相機、藍芽、螢幕擷取、YouTube、熱點、Play 商店	無	提供應用程式權限管理功能
安全檢測報告	無	SCAP 安全分數掃描	無
核心層安全加強	無	無	SEAndroid 功能 (sepolicy 更新、sepolicy 分析)
client 安裝方式	將 APK 安裝至裝置內	將 APK 安裝至裝置內	將應用程式加入 AOSP 內，編譯後刷入裝置內
認證方式	使用裝置獨特 ID、作業系統與作業系統版本	無	無
支援作業系統	Android、IOS、Windows Phone	Android	Android

參考文獻

[1] eMarketer. (2014). 2 Billion Consumers Worldwide to Get Smart(phones) by 2016. Available: http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694

[2] Gartner. (2016). Gartner Says Worldwide Smartphone Sales Grew 9.7 Percent in Fourth Quarter of 2015. Available: http://www.gartner.com/newsroom/id/3215217

[3] P. Faruki, A. Bharmal, V. Laxmi, M. S. Gaur, M. Conti, and M. Rajarajan, "Android Security: A Survey of Issues, Malware Penetration and Defenses," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 998-1022, 2015.

[4] E. Reshetova, F. Bonazzi, T. Nyman, R. Borgaonkar, and N. Asokan, "Characterizing SEAndroid Policies in the Wild," *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, pp. 482-489, 2016.

[5] xda developers.com. (2016). setools-android with sepolicy-inject. Available: http://forum.xda-developers.com/android/software/setools-android-sepolicy-inject-t2977563

[6] 董倫銘、楊中皇，基於 SE Android 結合遠端控制提升行動系統安全性之研究，*第二十一屆國際資訊管理暨實務研討會*，2015 年 11 月

[7] 張世奇、楊中皇，Android 平台智慧型行動裝置管理系統設計與實現，*2015 全國計算機會議(NCS2015)*，2015 年 12 月

[8] P. K. Gajar, A. Ghosh, and S. Rai, "Bring your own device(BYOD): security risks and mitigating strategies," *Journal of Global Research in Computer Science*, Vol. 4, 2013.

[9] SourceForge.net. (2014). Android Push Notification. Available: https://sourceforge.net/p/androidpn/

[10] N. Elenkov, *Android Security Internals: An In-Depth Guide to Android's Security Architecture*: No Starch Press, Inc., 2014.

[11] S. Smalley, and R. Craig, "Security Enhanced (SE) Android: Bringing Flexible MAC to Android," *In Proc. of the 20th Network and Distributed System Security Symposium (NDSS 2013)*, 2013.

[12] selinuxproject.org. (2013). SELinux Wiki. Available: http://selinuxproject.org/

[13] Android. The Android Source Code. Available: https://source.android.com/source/index.html

[14] K. Yaghmour, *Embedded Android*: O'Reilly Media, 2013.

[15] C. Yin, L. Liu, and L. Liu, " BYOD implementation: understanding organizational performance through a gift perspective," *PACIS 2014 Proceedings*, 2014.

[16] J. E. Lee, S. H. Park, and H. Yoon, "Security Policy based Device Management for Supporting Various Mobile OS," *2015 Second International Conference on Computing Technology and Information Management (ICCTIM)*, pp. 156-161, 2015.