Journal of Cheng Shiu University Vol.26 pp75-84 (2013)

探討 Android 系統安全機制

許博學
正修科技大學資訊工程系

摘 要

Android 系統安全機制的良窳，攸關龐大數量 Android 裝置在私人及企業使用上的安全性，影響至廣且深。妥善的應用程式隔離措施與權限管制資源存取是它的優點，但允許執行原生碼及未全面強制權限管制卻造成安全漏洞。本文提出對 Android 系統安全機制的改進建議，與現階段使用者可行的安全強化措施。

關鍵詞：Android，安全，權限，弱點。

Study in Android System Security Mechanisms

Po-hsueh Hsu

Department of Computer Science and Information Engineering Cheng Shiu University

Kaohsiung, Taiwan 83347, R.O.C.

ABSTRACT

Whether Android system security is good enough will make a great impact on information security, since a huge amount of Android devices are currently used for security-critical private and business applications. Android system has advantages of a very good design of application isolation mechanism and permissions control for system resources, but it is flawed by native code execution and unenforced permissions checking for caller's authority. Both suggestions for improvements on Android system security and feasible security consolidations for mobile users are major issues in this paper.

Keywords: Android, Security, Permission, Vulnerability.

壹、 研究重點

高度資訊化時代最棘手的問題，莫過於如何保障資訊安全。除了考量手機上重要個資外洩的危機之外，逐漸流行的帶你的裝置來上班(BYOD)也可能導致公司商業機密被竊。我們就最大使用量的 Android 裝置來探討其系統安全機制的特色及弱點，並且提出改進建議。我們也對 Android 裝置的使用者提供簡單易行的安全強化措施，希望能在資訊安全保障上有所貢獻。

1.1 研究動機

這是一個移動互聯網的時代，人們隨身攜帶智慧型行動裝置，隨時隨地上網存取資料、網路購物及銀行交易，甚至進入公司網路辦公。依據 Gartner 資料統計，2012 年第三季 Android 平台的市占率高達 72%，2012 年整體智慧型行動裝置的銷售量預計達到 8.21 億台，預估 2013 年智慧裝置產品的出貨量將可達到 12 億台。那麼如果 Android 系統存在著安全弱點再加上不良的手機使用行爲，這樣對資訊安全會造成多大的衝擊呢？

1.2 相關研究

Enck 等人發表在 IEEE Security & Privacy 的論文 Understanding Android Security [1]是一篇重要的文獻，對 Android 安全模型及元件間通訊機制有精彩的論述。當然 Google 官方文件[2,3]也是重要的參考資料，尤其在說明其獨特的權限管制作業上（假使有某個應用程式嘗試使用受保護的 API 功能卻未取得權限，系統在執行該應用程式時就會出現安全性例外並且終止應用程式）。對於加強 Android 裝置安全之研究，美國國家安全局提出了 SEAndroid 計畫[4]，本文作者則在教育部資助下，開發出一款嶄新設計理念的行動防毒軟體[5]。

每個系統一定都會有弱點，GingerMaster [6]是一款針對 Android 2.3 版本弱點進行攻擊來取得 root 特權的惡意程式。Fraunhofer 機構甚至設計出漏洞攻擊框架工具[7]，能夠依據各個 Android 版本已知弱點，對多款手機自動化執行弱點測試。至於未取得權限的應用程式就無法傷害 Android 裝置，這是真的嗎？有多篇的研究文獻[8-12]否定了這個論點，主要原因是手機預載的系統程式並未全面強制權限管制。Woodpecker 自動化工具[11]就八款市售流行手機所做的測試，共找出了 13 個權限漏洞。

Google 公司針對 Android Market 上惡意 Apps 氾濫狀況，在 2011 年推出自動化檢測惡意 Apps 的工具 Bouncer [3]。Bouncer 這項服務會在 Android Market 上背景運作，掃描檢測出 Android Market 上的惡意程式。同時這項服務也被應用在開發者帳號的管制機制，未來當有不良記錄的開發者上傳新的 App 時，Bouncer 會優先對此 App 進行分析，避免惡意程式、間諜程式或木馬入侵 Android Market。這項服務，使得在 Android Market 從 2011 年第 1 季至 2 季中旬，所潛在惡意程式下載量已經減少了 40%，對降低 Android Market 上惡意 Apps 氾濫狀況有很大的幫助。

貳、 Android 系統安全機制的優點

Android 系統在原本就強健的 Linux 核心上進一步強化了安全機制，層層安全管制措施構築起銅牆鐵壁。強化的安全機制包括應用程式化 UID 檔案隔離，虛擬機器沙盒保護，系統資源存取權限管制，以及元件間通訊授權標籤管制。Android 系統強化的安全機制如何運作，將在這一節說明之。

2.1 對每一個應用程式指定一個唯一的 UID

Android 系統建置在 Linux 核心上，並且將 Linux 核心以使用者爲區別的 UID（使用者識別碼）檔案隔離機制，擴充成對每一個應用程式指定一個唯一的 UID，來達成對手機上眾多應用程式間彼此隔離的目標。應用程式基本上不能存取不同 UID 的其他應用程式的程式碼及資料，除非那個應用程式開通了一個存取通道。

2.2 用虛擬機器提供沙盒保護

Android 系統上每一個應用程式被限制執行在個別的 DVM (Dalvik Virtual Machine)虛擬機器環境內，如同沙盒(Sandbox)般被安全地隔離保護著。即使應用程式設計不夠妥當，也不會危害到其它應用程式及系統。應用程式若要存取其它資源則需呼叫系統 API，但前提是它必須擁有適當權限。

2.3 以權限管制系統資源存取

系統資源存取管制是一個相當重要的安全課題，Android 系統要求應用程式要在 AndroidManifest.xml 設定檔內宣告需求的權限，並且在安裝時獲得使用者的同意，才能依照權限呼叫系統 API 存取對應資源。例如取得 android.permission.INTERNET 權限，才能存取網際網路；取得 android.permission.RECEIVE_BOOT_COMPLETED 權限，才能在手機開機時該應用程式即自動被啓動執行。

2.4 元件間通訊授權標籤管制

Android 系統把 Linux 核心的程序間通訊(IPC)進一步細分到元件間通訊(ICC)，並且用授權標籤來管制未獲授權者無法得到元件服務。例如某應用程式的內容提供者元件，它開放資料庫存取服務並且規定授權標籤爲”com.act.mab.db”，則其它應用程式必須設定正確的 URI (content://com.act.mab.db)之後，才能經由元件間通訊管道來存取該資料庫。

參、 Android 系統安全機制的缺失

在上一節我們說明了 Android 系統強化的安全機制，看似安全無虞但也潛藏了某些安全缺失。實際上惡意程式如何利用這些缺失來攻擊得逞，我們會在這一節說明之。運用原生碼對已知

的系統漏洞攻擊以取得 root 權限，與如何繞開系統資源存取之權限管制，是本節說明的重點。

3.1 Android 系統漏洞修補進度遲緩

每個系統必定存在著安全漏洞的問題，而漏洞修補進度的快慢則影響裝置的安全性甚鉅。Android 系統是個開放式系統，並且允許手機製造商客製化系統。因此產生兩項重大安全衝擊，一項是攻擊者在研究系統程式碼後容易找出漏洞；一項是漏洞修補派送過程牽涉到手機製造商，人力物力安排繁雜使得漏洞修補進度緩慢，造成攻擊者容易利用已知漏洞攻陷 Android 裝置。

3.2 使用者難以決定應用程式要求的權限是否合理

以權限來管制系統資源存取是個很好的主意，但是當使用者在安裝程式時面對著權限清單，大多難以一一判斷每個權限要求是否合理。而在若不是允許全部的權限就不能安裝程式的狀況下，大多數人應對的方式爲不細看權限清單就直接按接受鈕來安裝程式。設計上很棒的主意但在實務上難以執行，如何落實權限管制是個需要再審慎商議的課題。

3.3 原生碼脫離虛擬機器的掌控

Android 應用程式被限制在 DVM 虛擬機器環境內執行，但是有一項例外，程式內含的原生碼(Native Code)會脫離虛擬機器的掌控，也不受權限管制，可以直接存取系統資源。事實上駭客就是利用原生碼攻擊系統漏洞，來取得最高使用者 root 權限以掌控 Android 裝置。原生碼本來是爲了提高 Android 系統效能，應用在需要密集運算場合同時也能呼叫 libc 函式庫，卻意外變成了 Android 系統安全機制最大的衝擊。

3.4 允許將偷渡下載檔案設定爲可執行

應用程式執行過程中經由網際網路下載到裝置的檔案，在使用者模式下可以被設定爲可執行檔以啓動執行，造成偷渡下載(Drive-by Download)形式攻擊途徑。隱含偷渡下載之惡意程式具備機動又變裝的特色，不易被防毒軟體偵測出來，是一種威力相當強大的攻擊方式。Android 系統如果能在使用者模式下禁止將檔案設定爲可執行，就可以阻擋偷渡下載式的惡意攻擊。

3.5 無權限下可以間接存取系統元件

應用程式可以經由意圖(Intent)物件以間接存取系統元件的方式，來執行它無權限下本來應該不能執行的行爲，這是因爲 Android 系統未全面強制權限管制的緣故。例如應用程式雖然沒有網路通訊權限(android.permission.INTERNET)，但是它可以經過意圖物件呼叫瀏覽器元件的方式，間接地將裝置識別碼傳遞給伺服器 example.com，如下列程式碼所示：

```
Intent intent = new Intent(Intent.ACTION_VIEW,Uri.parse("http://example.com/?device_id=xxx"));
startActivity(intent);
```

肆、 Android 系統安全機制的改進建議

針對第三節提到的各項 Android 系統安全機制的缺失，我們在這一節提出改進建議，希望 Google 公司參酌意見盡快強固系統。畢竟 Android 系統在行動裝置上市占率最高，同時也是駭客攻擊的主要目標，盡速改進系統安全缺失以及主動修補裝置漏洞是 Google 公司無可閃避的責任。

4.1 向微軟公司看齊，每月主動發佈 Android 系統漏洞修補

雖然市售上的裝置系統是由手機製造廠商客製化的系統，但是只有 Google 公司有人力資源來修補已被挖掘出的系統漏洞。系統漏洞修補的作業方式不應該是待 Google 改版 Android 系統後，再交給各個手機製造商客製出系統映像檔後再派送到裝置上，如此作業過程冗長緩不濟急。Google 應該向微軟學習 Windows 系統漏洞修補的作業方式，至少每個月主動將修補程式碼派送到裝置上更新。也可以考慮在裝置上安置 Android Update 程式，向伺服器要求派送裝置作業系統版本應該要修補的程式碼。

4.2 自動化篩選加上專人審核 Apps 權限要求是否恰如所需

在 3.2 節討論到以權限來管制系統資源存取是個很好的主意，但在實務上交由使用者在安裝程式時做出正確的授權判斷卻是窒礙難行。Google 應該考量如下列可行又有效的 Apps 權限管理措施：首先使用自動化軟體工具篩選出有惡意嫌疑的 Apps（例如要求開機啓動權限者），接著派專人審核這些 Apps 之權限要求是否恰如 Apps 功能所需，不符者予以下架。自動化軟體工具只須對 APK 解壓縮後的 AndroidManifest.xml 檔案，實施這個 App 是否要求開機啓動權限一項檢測而已，所以可以從數量龐大的 Apps 中快速篩選出嫌疑犯。針對這些嫌疑 Apps，後續的專業審核能夠有經驗有效率地決定 Apps 之權限要求是否合理。如此不但達到以權限來管制系統資源存取的目標，使用者在安裝程式時也不用再面對不知該如何處理的權限清單。

4.3 限制非系統程式不能使用原生碼

程式內含的原生碼會脫離虛擬機器的掌控，也不受權限管制，是駭客用來攻擊系統漏洞取得 root 權限的主要武器，必定要嚴加管控原生碼的使用。建議 Android 系統應該檢查程式的 UID，一律擋掉內含原生碼的非系統程式，只允許系統程式才能使用原生碼。限制非系統程式不能使用原生碼的方案，將能大幅度降低 Android 裝置被攻陷的機會。

4.4 禁止應用程式將下載檔案設定爲可執行

美國國家安全局的 Android 安全強化計畫 SEAndroid [4]證實了讓應用程式下載的檔案可以被設定爲可執行檔，是一項 Android 系統的重大安全漏洞。這種偷渡下載的惡意攻擊既隱匿又機動，是一種威力強大很難防範的攻擊模式。Android 系統應該密切監視並且禁止一般的應用程式將下載檔案設定爲可執行，就可以阻擋偷渡下載式惡意攻擊。

4.5 強制檢查呼叫者權限，避免系統元件被無權限者委託執行

在 3.5 節提到應用程式雖然沒有網路通訊權限，但是它可以經過意圖物件呼叫系統瀏覽器元件，以委託方式間接地達成通訊目的。改進建議爲在系統元件被呼叫時應先強制檢查呼叫者是否有對應的權限，唯確認權限後才接受委任。資安專家 Andre Moulu 發現 Samsung S3 預載的某個服務具備將 APK 檔案複製到 SD 卡內某個目錄的功能，還有另一個服務會從這個目錄安裝 APK 檔案，但都同樣不會貫徹權限機制或檢查呼叫者。透過這兩項服務，惡意程式就能將 APK 檔案放入對應的目錄進而安裝到系統內，不僅神不知鬼不覺，當然更不需取得任何系統權限[12]。

4.6 預防手機對手機與 PC 對手機的病毒傳播感染

目前行動裝置上的病毒傳播感染還很少見，不像桌上/膝上裝置的蠕蟲病毒強大的傳播感染力那麼的令人聞之色變。不過只要 Android 裝置的 USB 除錯模式啓用（Android 3.1 內定啓用 USB 主機模式）再插入 USB 纜線，那麼實作 Android 除錯橋接 adb 協定的病毒就能主動從其他 PC/手機端將病毒傳播到這台 USB 連線的手機上。對付這種可能的病毒傳播方式，預防方案是強制使用者輸入預設密碼後才能接受遠端程式安裝 (ex. adb install xxx.apk）。

伍、 Android 手機使用者的安全強化措施

Android 手機的資訊安全建立在兩項基礎上，一項是強固的 Android 作業系統，另一項則是良好的使用行爲。慎選 Apps 下載使用以及絕不刷機，可以相當程度地幫助維護手機的資訊安全。我們也推薦兩款設計理念不同但是都很實用突出的防毒軟體 TrustGo 與 MAB 戰警，來幫助守衛你的手機。如果你是熱門免費遊戲的重度愛用者，那麼你一定會對善於幫你挑出可疑遊戲的 MAB 戰警留下深刻的印象。

5.1 不去第三方市集下載 Apps，在 Google Play 下載 Apps 之前檢視開發商信譽及使用評價

智慧型手機令人著迷之處在於下載市集上各式各樣的 Apps，經由 Apps 的使用豐富了人們的生活。然而第三方 Apps 市集完全不做審核管理，是惡意程式散播的溫床。建議使用者只在官方市集 Google Play 與電信商市集如 Hami Apps 下載，並且先檢視開發商信譽及使用評價，滿意之後再安裝使用。同類型的 Apps 相當多可選擇的空間很大，慎選信譽佳的開發商發行的 Apps 是維護手機資訊安全的基本做法。

5.2 絕不刷機，跟上漏洞修補及系統改版

有些人爲了完全掌控使用權而將手機刷機，取得 root 特權之後可以依他的意來使用手機。刷機後他可以移除電信商或製造商加入的程式，也可以到第三方市集免費下載本來要付費的程式，但是方便門後卻也打開了後門讓惡意程式得以長驅直入。建議使用者絕對不要刷機，才能夠避免手機被惡意者暗中掌控卻不自覺的危況。此外接獲系統改版或者程式更新通知時，及時修補漏洞的做法也能降低裝置被攻陷的機會。

5.3 安裝 TrustGo 免費防毒軟體

在加強手機安全的種種方案中，最有效益的方案莫過於安裝免費防毒軟體。在此我們推薦 TrustGo 防毒軟體，它具備優越的病毒偵測、隱私防護以及裝置保全功能。在病毒偵測方面它是 AV-Comparatives 認定的惡意軟件檢測最好的產品之一，在隱私防護方面它讓你檢視那些應用程式可以存取聯絡人簡訊等重要個資並提供雲端資料備份回復功能，在裝置保全方面能夠尋找你的手機與遠端鎖定手機。集合這麼多實用功能在一個免費軟體內，TrustGo 防毒軟體是你手機安全的最佳守門員。

5.4 安裝會主動提示可疑程式的 MAB 戰警

MAB 戰警是我們針對 Android 手機最大的安全威脅，未知型病毒攻擊，在教育部 101 年資訊軟體人才培育推廣計畫案經費補助以及與中華電信研究院產學合作下，研發出來的防毒軟體，可以在 Hami Apps 市集免費下載。MAB 戰警使用上很簡單，它的警訊會提示「如果你認爲這個 APP 不應該有開機啓動權限（可能是惡意程式），請卸除此可疑 APP」。建議除了通訊類如 LINE、市集類如 GOOGLE PLAY、防毒類如 AVIRA 之外，你應該卸除此可疑 APP，改爲安裝同類型但未要求開機啓動權限者。尤其如果此可疑 APP 是遊戲類型者，那就絕對是惡意程式。

陸、 結論

隨著越來越龐大數量的Android裝置普遍被使用在私人及企業用途上，我們不能坐視Android系統的安全漏洞會對資訊安全造成多大的衝擊。這篇論文敘述 Android 系統獨特的應用程式隔離機制與對資源存取的權限管制措施，同時探討駭客如何利用原生碼以及未全面強制權限管制的安全漏洞來攻陷 Android 裝置。針對 Android 系統安全機制的改進建議，與現階段使用者可行的安全強化措施，也是本文討論的重點。如果 Android Market 上的 Bouncer 這項服務再搭配 Android 平台使用沙盒技術、更嚴格的權限認可等等安全機制，應可以更有效的來防範惡意程式攻擊。

參考文獻

1. Enck, W., Ongtang, M. and Mcdaniel, P., "Understanding Android Security," IEEE SECURITY & PRIVACY, Jan. 2009, http://css.csail.mit.edu/6.858/2012/readings/android.pdf

2. Android Security Team, “Android Security Overview,” Dec. 2011, http://source.android.com/tech/security/index.html

3. Lockheimer, H., “Android and Security,” Feb. 2012, http://googlemobile.blogspot.com/2012/02/android-and-security.html

4. National Security Agency, “SEAndroid Project Page,” Jan. 2012, http://selinuxproject.org/page/SEAndroid

5. 許博學,「設計一個 Android 智慧型手機上的防毒軟體」,正修學報,卷 24,頁 59-67,Nov. 2011。

6. Jiang, X., “GingerMaster: First Android Malware Utilizing a Root Exploit on Android 2.3 (Gingerbread),” Aug. 2011, http://www.cs.ncsu.edu/faculty/jiang/GingerMaster

7. Fedler, R., Banse, C., Krauss, C., and Fusenig, V. "Android OS Security: Risks and Limitations," May 2012, http://www.aisec.fraunhofer.de/content/dam/aisec/de/pdf/tech%20reports/AISEC-TR-2012-001-Android-OS-Security.pdf

8. Lineberry, A., Richardson, D. and Wyatt, T., “These Aren’t The Permissions You Are Looking For,” in DEFCON 18, Aug. 2010, https://www.defcon.org/images/defcon-18/dc-18-presentations/Lineberry/DEFCON-18-Lineberry-Not-The-Permissions-You-Are-Looking-For.pdf

9. Vidas, T., Votipka, D. and Christin, N., “All your droid are belong to us: A survey of current android attacks,” in 5th USENIX Workshop on Offensive Technologies, Aug. 2011, http://static.usenix.org/event/woot/tech/final_files/Vidas.pdf

10. Cannon, T., “No-permission android app gives remote shell,” Dec. 2011, http://viaforensics.com/security/nopermission-android-app-remote-shell.html

11. Grace, M., Zhou, Y., Wang, Z., and Jiang, X., "Systematic Detection of Capability Leaks in Stock Android Smartphones," NDSS 2012, Feb. 2012, http://www.csc.ncsu.edu/faculty/jiang/pubs/NDSS12_WOODPECKER.pdf

12. Qiu, Y., “Bypassing Android Permissions: What You Need to Know,” Nov. 2012, http://blog.trendmicro.com/trendlabs-security-intelligence/bypassing-android-permissions-what-you-need-to-know/