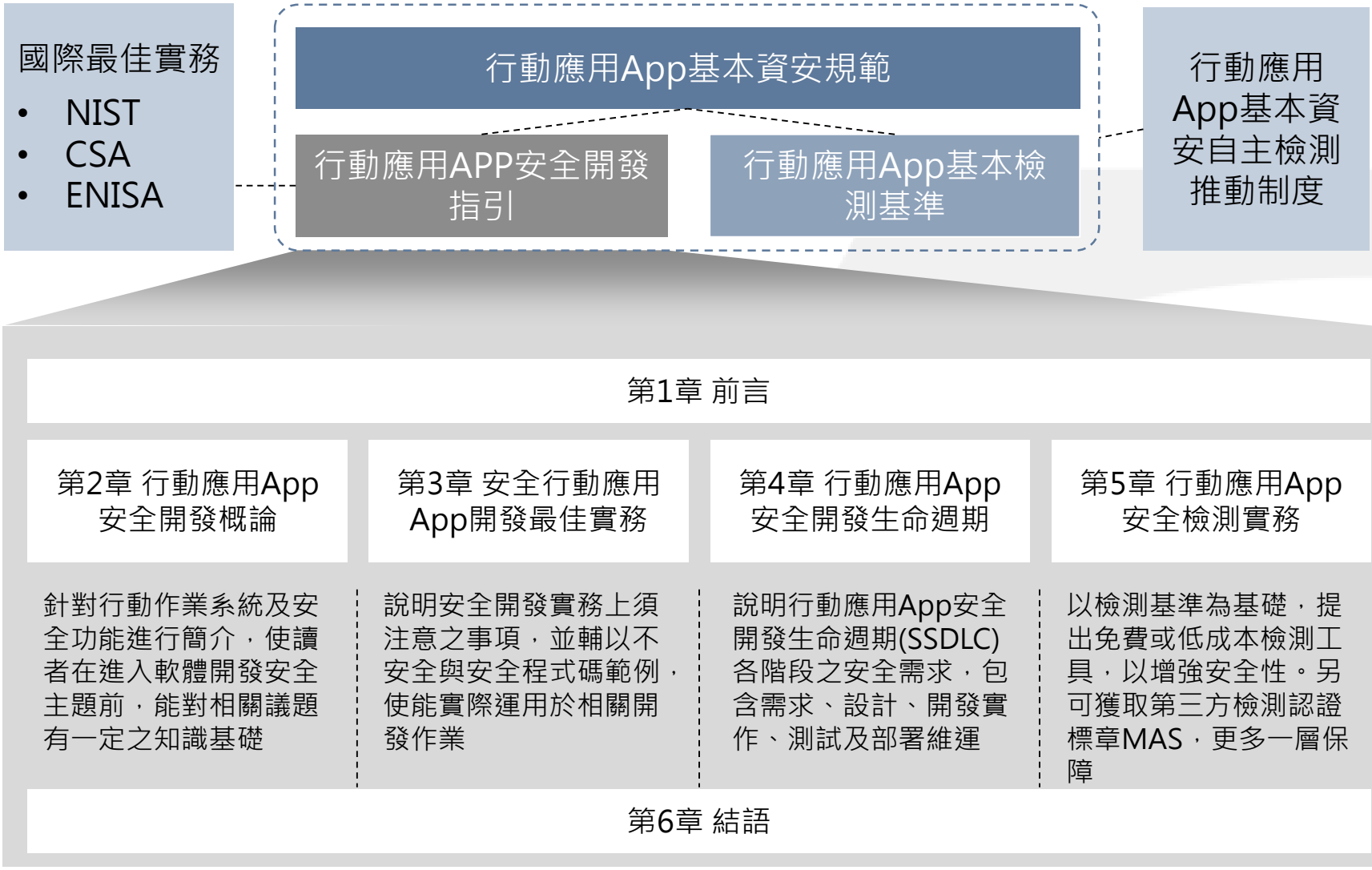
行動應用App安全開發說明會
Android基礎概念與實務入門

指導單位：經濟部工業局

執行單位：財團法人資訊工業策進會、中華民國資訊軟體協會

協辦單位：台北市電腦公會、中華民國資訊安全學會

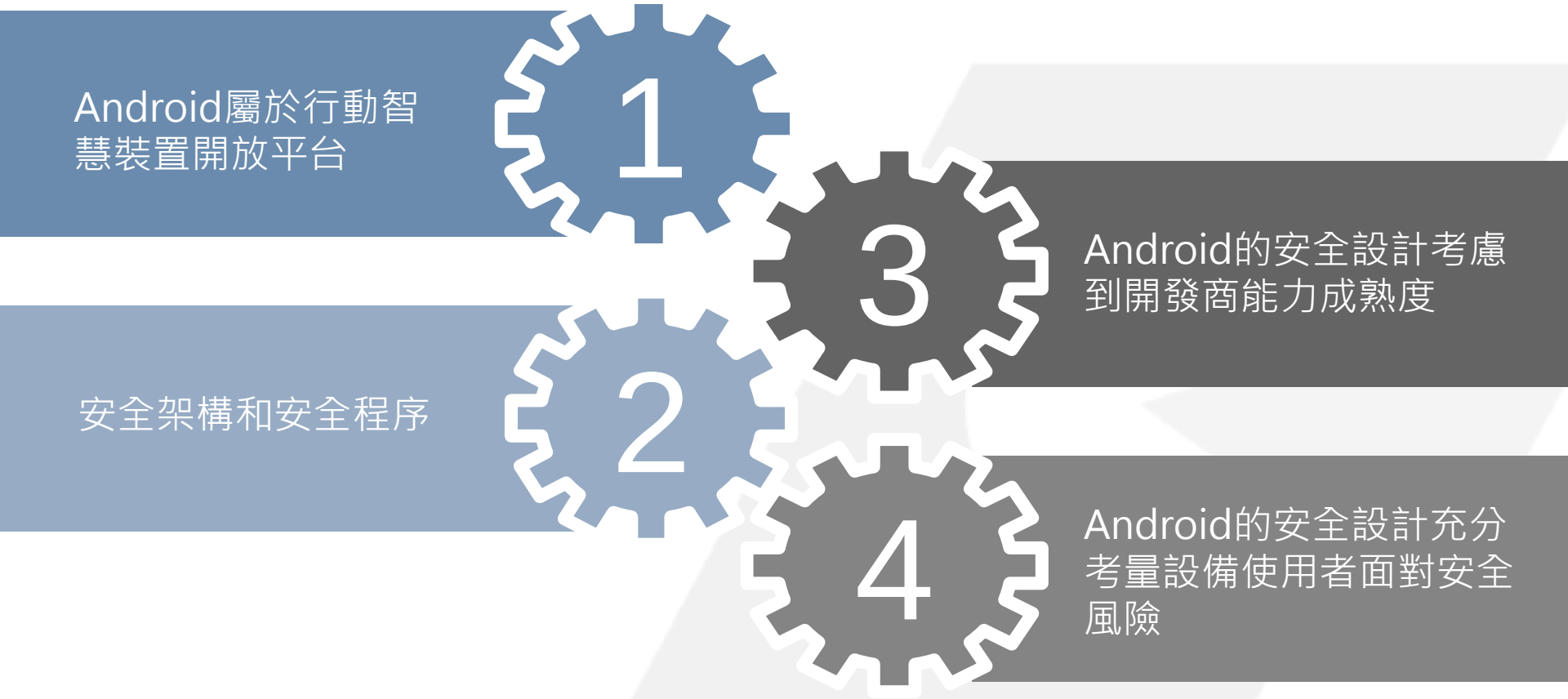
行動應用APP安全開發指引架構簡介



課程大綱

第1單元	行動應用App安全基礎概論	0.5小時
第2單元	安全軟體開發生命週期	0.75小時
第3單元	安全行動應用設計最佳實務	1.25小時
第4單元	行動應用App安全檢測流程與工具	0.5小時

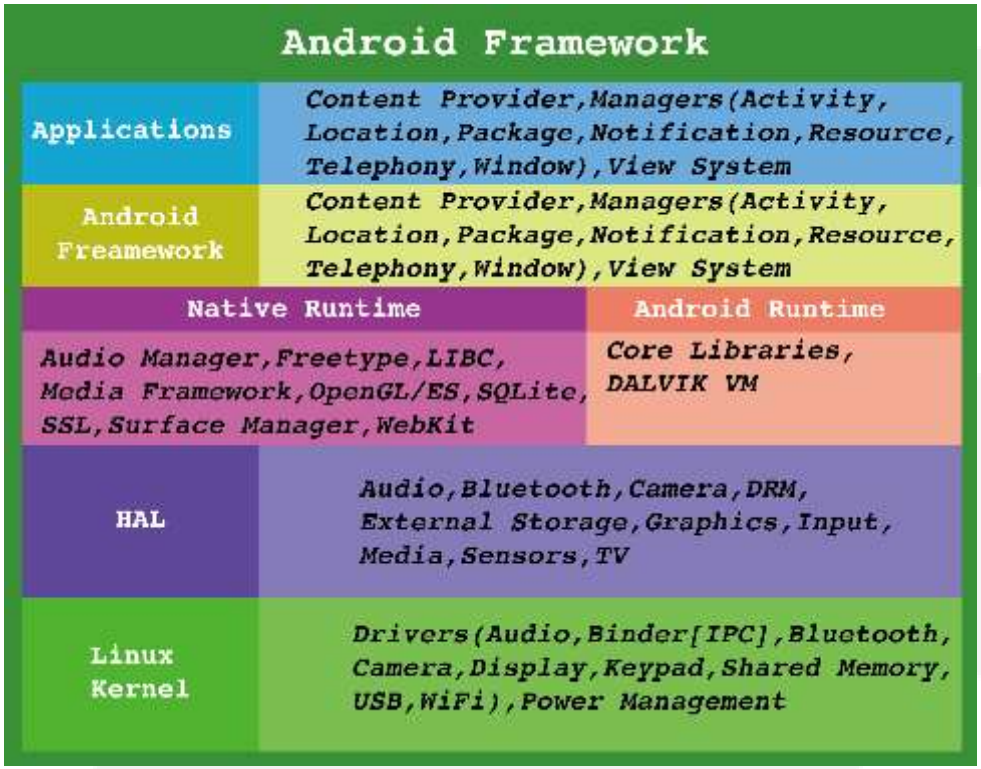
Android安全概觀



Android平台系統架構

Android是採用Linux核心的軟體平台和作業系統，它採用了軟體堆疊(Software Stack)，或稱為「軟體疊層架構」。

Android就是在Linux系統上增加了Java虛擬機Dalvik，並在Dalvik虛擬機上搭建一個JAVA的Application framework，而所有的應用程式都是基於JAVA的application framework之上。



Android軟體堆疊架構

資料來源：https://source.android.com/security/index.html

Android架構中的各層

應用程式層 (Applications)	▪ 以Java語言寫成。 ▪ 可從程式市集下載免費或商用的App。 ▪ 每個App都有自己的執行空間，不會干擾其他的App。
應用程式層框架 (Application Framework)	▪ 實作JNI技術。 ▪ 開發員在寫App時可以完整的存取使用統一的應用程式標準介面。 ▪ 在寫App時可以重覆使用各個元件。
函式庫 (Libraries)	▪ Android所有元件都是由C/C++函式庫所組成。 ▪ Android所有的應用程式都必需經由API來使用這些功能。
Android執行層 (Runtimes)	▪ 使用Android自有的Android Runtime來執行Java程式。 ▪ 包含Core Libraries及Dalvik Virtual Machine。
硬體抽象層 (Hardware Abstraction Layer)	▪ 將Android的Framework與Linux Kernel進行區隔。 ▪ 當Android Framework需要接觸到硬體功能時，皆需透過HAL向底層呼叫，讓Android Framework可以在完全不考慮驅動程式的情況下發展。
作業系統內核層 (Linux Kernel)	採用Linux內核。App存取設備內所有資源，如照相功能、全球定位系統、藍牙功能、電話功能及網路連結等皆透過作業系統存取。

Android平台安全架構

以最安全和可用最高的行動作業平台為目標

安全控制目標		
保護使用者資料	保護系統資源(包括網路)	提供App隔離

為了實現上述安全控制，Android提供以下關鍵安全功能

- 藉由Linux kernel強化作業的安全性
- 強制所有App使用Application Sandbox(沙箱)
- 保全程序間的溝通管道與內容
- App簽名(Signing)
- 由App定義(Application-defined)的權限和使用者授予(User-granted)的權限

在基礎安全功能尚包括：

- 身分認證(Authentication)
- 全磁碟加密(Full Disk Encryption)
- 安全增強式Linux(Security-Enhanced Linux in Android)
- 驗證啟動(Verified Boot)

安全維護計畫(Security Program)

1 設計審查

從Android平台開發生命週期的早期，Android平台即設計了安全流程與創建豐富和可配置的安全模型。

該平台的每個主要功能均由工程和安全資源審查，有適當的安全控制整合到系統的體系結構。

2 滲透測試和程式審查

在平台的開發過程，Android核心系統所創建和來自開源的元件均受到大量的安全審查。這些審查是由Android的安全團隊、、Google的資訊安全工程團隊，及獨立安全顧問執行。

審查的目的是找出App薄弱環節和潛在漏洞以及開源平台既有漏洞，在正式版本釋出前將由外部安全專家模擬分析這些類型的漏洞。

3 開源與社群評論

Android開源專案均經過任何有關方都能夠參與的廣泛的安全審查活動。Android版還使用已經過嚴格外部安全審查的開源技術，如Linux內核。Google Play為使用者和企業提供有關特定App提供一個論壇，可使用者的直接獲得資訊。

4 安全事故應變流程

即使有以上的安全控制措施，上架後還是可能出現的安全問題，這也是為什麼Android專案建立了一個全面的安全事故應變流程。

全天候Android安全團隊不斷監控Android特有的和潛在漏洞的一般安全討論社群。

一旦合法地發現問題，Android團隊以既有應變流程，使安全漏洞的快速緩解，以確保潛在的風險對所有Android使用者衝擊最小化。

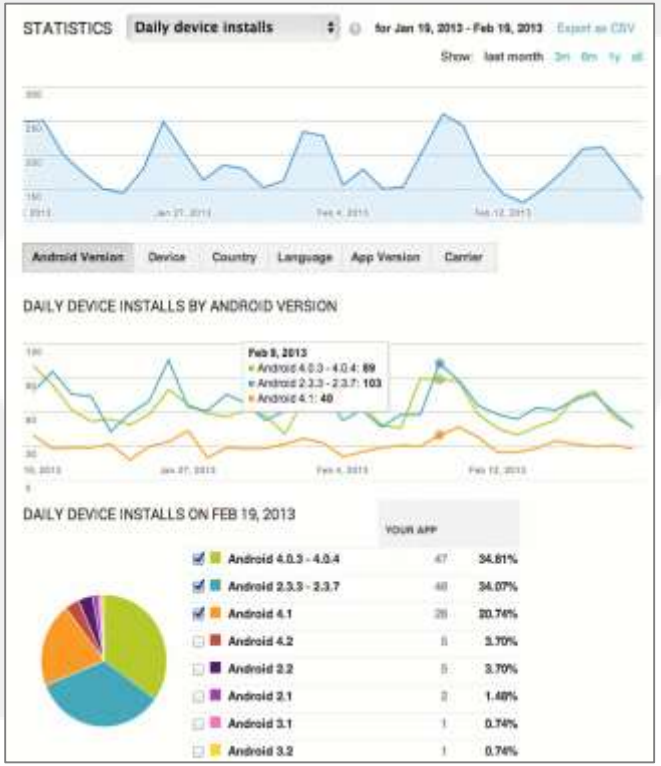
要開發Android App，你要準備幾件事

1. 安裝Java JDK
2. 安裝Eclipse軟體、Eclipse ADT及Android SDK(或Android Studio整合式開發環境)
3. 下載適當的Android作業系統版本
4. 學習Java語言
5. 於Google Play開發人員控制台註冊帳號

App要上架，你應該做這幾件事

STEP 1	STEP 2	STEP 3
取得帳號	**上傳App**	**上架完成**
於「Google Play 開發者控制台」*註冊帳號，並交註冊費	註冊成功後，上傳App	使用者即可下載使用

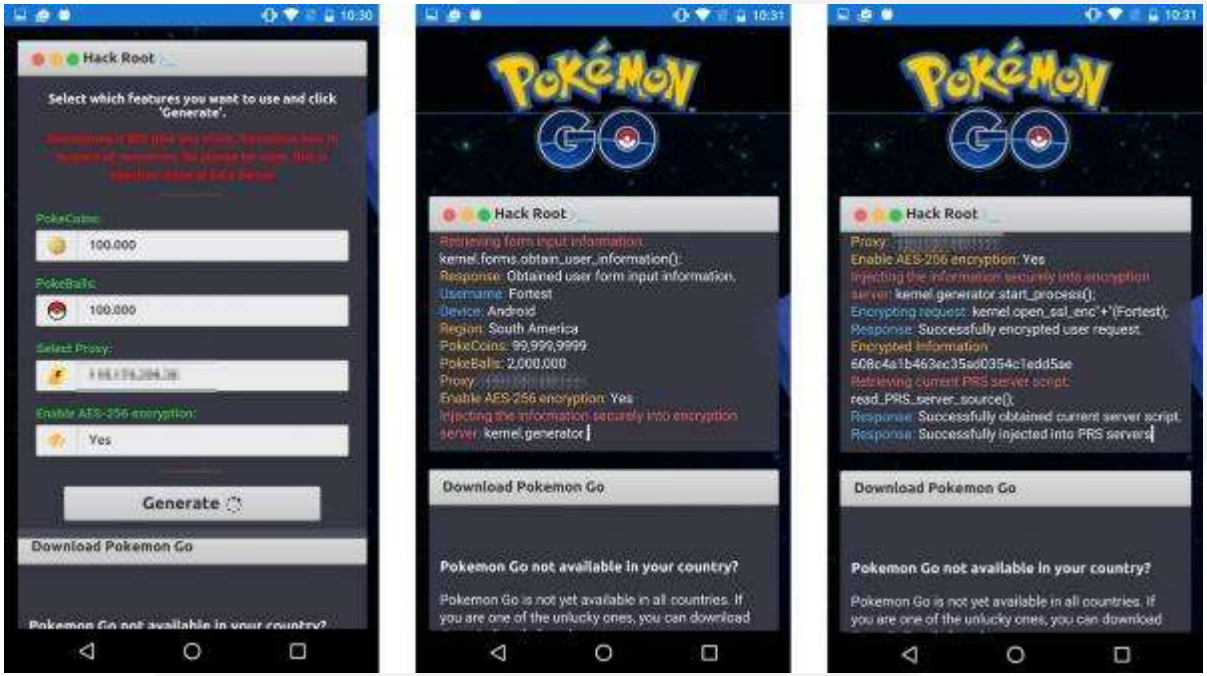
Google Play開發者控制台 (Google Play Developer Console)
提供行動應用App上架、行銷及後續追蹤的控制台，需要付費註冊開發人員帳號才能使用。



行動應用App的資安新聞-冒牌Pokémon Go

App開發者，常限於開發時間緊迫、人力成本緊縮及未考慮到使用情境的安全問題，或是開發者從未接受安全程式設計實務訓練，寫出有漏洞程式。

- 夾帶木馬程式
- 暗藏著惡意的彈出視窗或廣告



資料來源：趨勢科技，http://blog.trendmicro.com.tw/?cat=15 (105/8/11)

OWASP十大行動安全風險

國際知名開放軟體安全計畫(Open Web Application Security Project, OWASP)
發布2016年行動應用App前10大安全風險侯選清單

1. M1不當使用行動作業平台(M1 - Improper Platform Usage)
2. M2不安全資料儲存(M2-Insecure Data Storage)
3. M3-不安全通訊(M3-Insecure Communication)
4. M4-不安全身分認證(M4-Insecure Authentication)
5. M5-不足夠的加密(M5-Insufficient Cryptography)
6. M6-不安全授權(M6-Insecure Authorization)
7. M7-用戶端程式碼品質(M7-Client Code Quality)
8. M8-程式碼篡改(M8-Code Tampering)
9. M9-逆向工程(M9-Reverse Engineering)
10. M10-多餘的功能(M10-Extraneous Functionality)

行動應用App常見威脅

趨勢科技將行動應用App常見威脅

1 行動裝置勒索程式

加密使用者資料，勒索使用者換取解密金鑰。

2 越權廣告程式

未經使用者同意取得或獲取過多權限，取用使用者敏感性資料。

3 費率服務盜用程式

在背景偷用一些高費率服務，讓使用者的手機帳單費用無故飆高。

4 免越獄植入木馬

竊取憑證，將木馬程式安裝在沒有越獄的裝置上。

5 有漏洞程式庫/程式開發套件

處理敏感性資料程式庫及開發工具漏洞被利用。

6 偽造的程式ID

可讓惡意程式假冒正常程式的Android FakeID漏洞。

7 內建瀏覽器跨站來源腳本存取漏洞

惡意程式可能存取正常網站所使用的資料和Cookie，取得敏感性資料。

中國大陸(China)

大陸工業與信息化部發布了一系列行動智慧裝置安全標準

行動智慧裝置安全能力設計導則 | 行動智慧裝置安全能力技術要求 ✓ | 行動智慧裝置安全能力測試方法 ✓ | 移動終端晶片安全技術要求和測試方法

YD/T 2407-2013 行動智慧裝置安全能力技術要求

- 僅制訂其基本原則：「行動智慧裝置上發生的行為和應用要符合用戶的意願 」，無規定具體的實現方法和措施。
- 作為安全開發準則參考之一：
  - 硬體安全能力要求
  - 作業系統安全能力要求
  - 週邊接口安全能力要求
  - 應用軟體安全要求
  - 使用者資料安全保護能力要求

與行動應用App安全開發相關

YD/T 2408-2013 行動智慧裝置安全能力測試方法

針對技術要求提出相對應的技術指標設計及測試方法，用於驗證行動智慧裝置是否滿足技術要求的規定。

歐盟(Europe Union, EU)

歐洲網路暨資訊安全局(ENISA)於2011年發布的「智慧型手機開發者安全開發指引(Smartphone Secure Development Guidelines for App Developers, SSDGAD)」

針對下列3個計畫進行行動應用App的安全風險評估

OWASP Mobile Top 10 Risks	OWASP Web Top 10 Risks	OWASP Cloud Top 10 Risks

行動應用App前10大控制措施分類	控制措施
1.在行動設備上識別和保護敏感資料	14
2.在設備上安全地處理密碼憑證	10
3.確保敏感資料在傳輸過程中受保護	7
4.正確地建置用戶認證、授權和連線管理機制	6
5.確保後端的API(服務)和平台(伺服器)的安全	5
6.確保使用第三方服務和App時的資料安全	3
7.應有機制蒐集及保留使用者所簽署的個資使用同意證明	6
8.建立機制以防止付費資源(電子錢包、簡訊、電話等)被未經授權的存取	7
9.確保得以安全的發布 /更新行動應用App	3
10.小心檢查程式碼在執行直譯時可能會發生的錯誤	4
合計	65

日本(Japan)

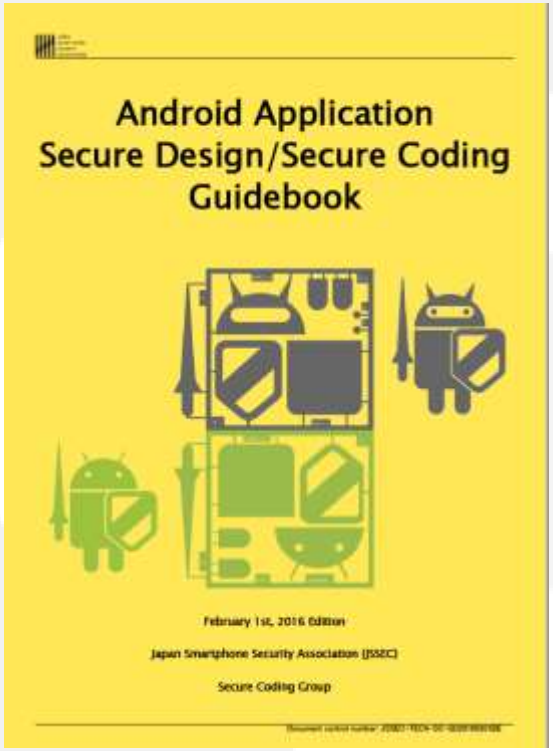
由日本智慧型手機安全論壇(Japan Smartphone Security Forum，JSSEC Forum)於2016年2月所發布的「Android App安全設計 / 安全程式碼編寫指南(Android Application Secure Design/Secure Coding Guidebook) 」。

- 該指引的目的是為了促進Android App安全設計與安全程式碼編寫的綜合技巧，並保持及時更新及分享安全開發實務。
- JSSEC認為Android是開放式系統，需要較iOS花費更多精力去加強行動應用App安全性，因為Android可以不必透過Google Play來發布，缺乏了平台的安全把關，有較多被惡意運用存取使用者敏感性資料或是有更多安全性漏洞可能被利用來進行惡意攻擊。

最新版的指南與程式碼範例的URL如下：

http://www.jssec.org/dl/android_securecoding_en.pdf
Guidebook (English)

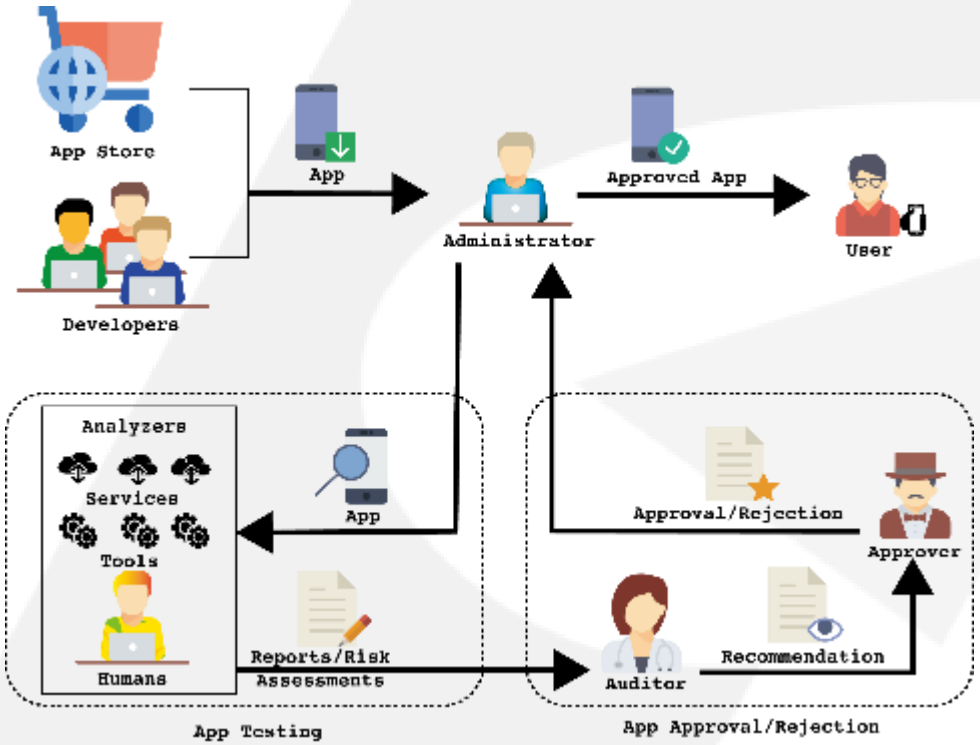
http://www.jssec.org/dl/android_securecoding_en.zip
Sample Codes (English)



美國(United States of America, USA)

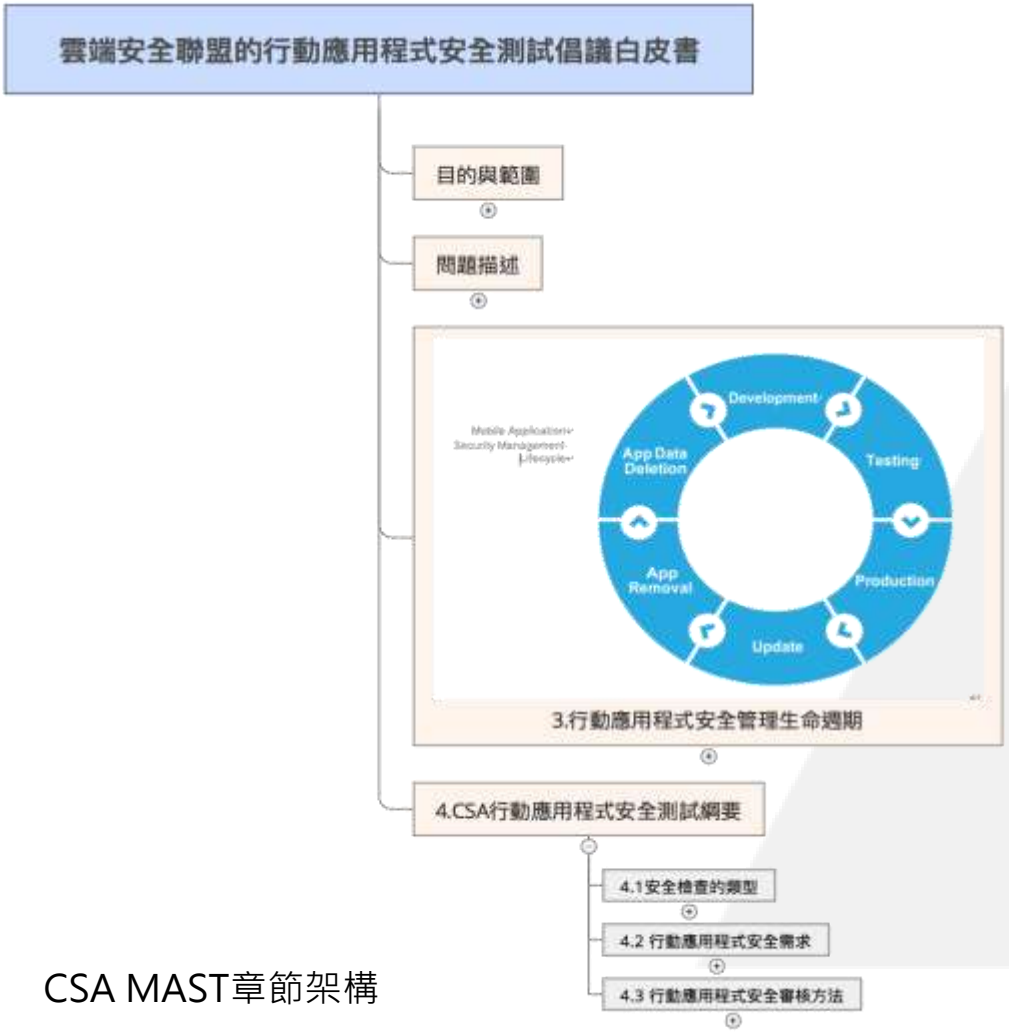
國家標準與科技機構(NIST)所發布NIST SP 800-163行動應用App安全審核(Vetting the Security of Mobile Applications)。

- 幫助企業了解App審核流程
- 協助企業規劃進行審核的整個過程
- 協助企業了解開發App的安全要求
- 協助企業了解各種App的漏洞，和測試這些漏洞的方法
- 協助企業了解該App是否適合安裝在組織的行動裝置上



行動應用App審核流程及角色　資料來源：NIST SP800-163

雲端安全聯盟的行動應用App安全測試倡議白皮書



CSA MAST章節架構

- 由雲端安全聯盟(Cloud Security Alliance, CSA)提出的行動應用APP安全測試(Mobile Application Security Testing, MAST)白皮書，旨在協助企業或個人在使用行動應用App時，可降低其潛在的風險及資安威脅。
- MAST定義了一個行動應用App的安全開發框架，以處理使用者隱私與資安的議題。

行動應用App基本資安規範

- 於104年由經濟部工業局委託資策會研議，供業界於開發行動應用App時的自主遵循參考。
- 為非強制性規定，主要目的在於提升我國行動應用App基本安全防護能力，該規範建議從設計初始階段即導入基本資安概念，並提醒App開發者應強化資訊安全意識，逐步完善自身App全防護能力。
- 該規範列舉17項關於行動應用App開發的資安技術要求，並提出3種不同的行動應用App安全分類。

下載網址:http://www.communications.org.tw/news/policy/item/8743-0814.html

編號	資訊安全技術要求事項	安全分類		
		一	二	三
1	4.1.1.1.行動App發布	V	V	V
2	4.1.1.2.行動App更新	V	V	V
3	4.1.1.3.行動App安全性問題回報	V	V	V
4	4.1.2.1.敏感性資料蒐集	V	V	V
5	4.1.2.2.敏感性資料利用	V	V	V
6	4.1.2.3.敏感性資料儲存	V	V	V
7	4.1.2.4.敏感性資料傳輸		V	V
8	4.1.2.5.敏感性資料分享	V	V	V
9	4.1.2.6.敏感性資料刪除	V	V	V
10	4.1.3.1.付費資源使用			V
11	4.1.3.2.付費資源控管			V
12	4.1.4.1.使用者身分認證與授權		V	V
13	4.1.4.2.連線管理機制		V	V
14	4.1.5.1.防範惡意程式碼與避免資訊安全漏洞	V	V	V
15	4.1.5.2.行動App完整性			V
16	4.1.5.3.函式庫引用安全	V	V	V
17	4.1.5.4.使用者輸入驗證	V	V	V

安全分類
第一類：純功能性。
第二類：具認證功能與連網行為。
第三類：具交易功能(包括認證功能及連網行為)。

行動應用App基本資安檢測基準

以「行動應用App基本資安規範」中對行動應用App的安全分類為主，再參考OWASP之Mobile Security Project-Top Ten Mobile Risks與NIST SP 800-163 Vetting the Security of Mobile Applications，評估及審驗相關行動應用App之風險項目，藉此制定安全檢測項目並規劃提出各項目之細項檢查事項、執行條件與預期結果等。

亦可作為
- 第三方檢測機構進行檢測之參考基準
- 開發者針對App安全進行檢測之依據

經濟部工業局行動應用App安全基本規範

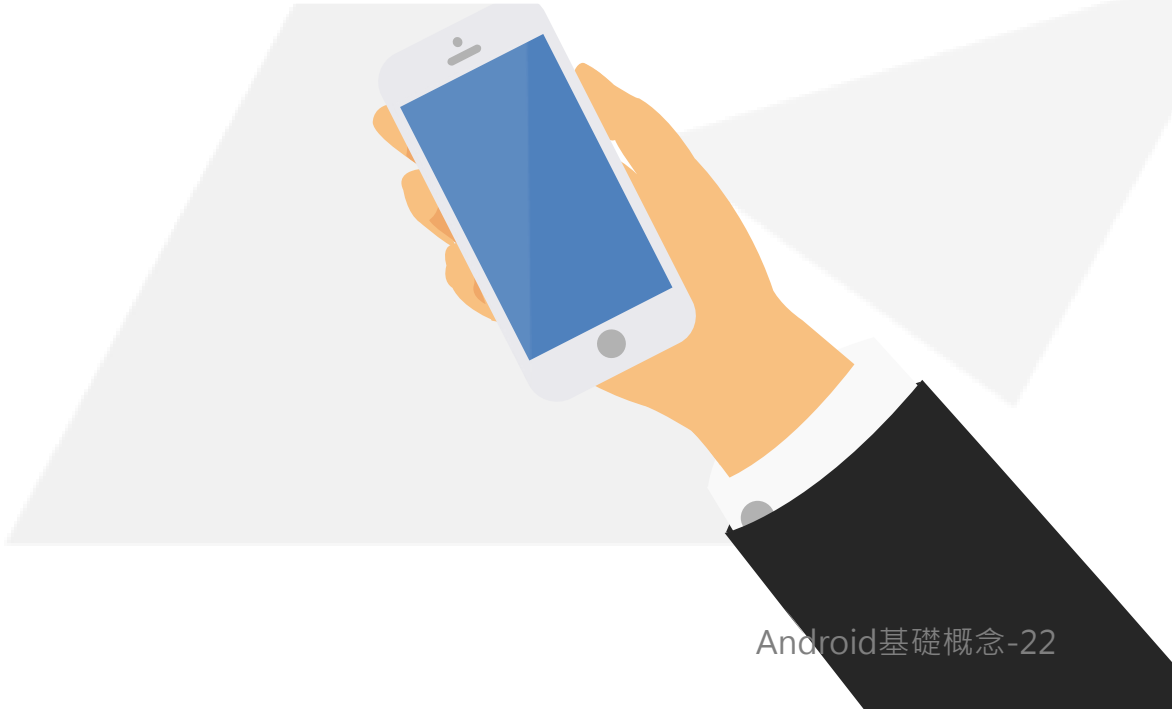
下載網址:https://www.communications.org.tw/phocadownloadpap/app/v2.0_1.pdf

行動應用App基本資安檢測基準-文件架構

資料來源：經濟部工業局行動應用App基本資安檢測及制度說明簡報

課程大綱

第1單元	行動應用App安全基礎概論	0.5小時
第2單元	安全軟體開發生命週期	0.75小時
第3單元	安全行動應用設計最佳實務	1.25小時
第4單元	行動應用App安全檢測流程與工具	0.5小時



行動應用App安全開發生命週期方法論

常見安全開發生命週期(SSDLC)方法論

- Cigital的Touchpoint
- Microsoft的Security Development Lifecycle(SDL)
- OWASP的Security Assurance Maturity Model(SAMM)

方法論比較摘述請詳見附件1「安全軟體開發生命週期方法論比較」

Cigital Touchpoint方法論

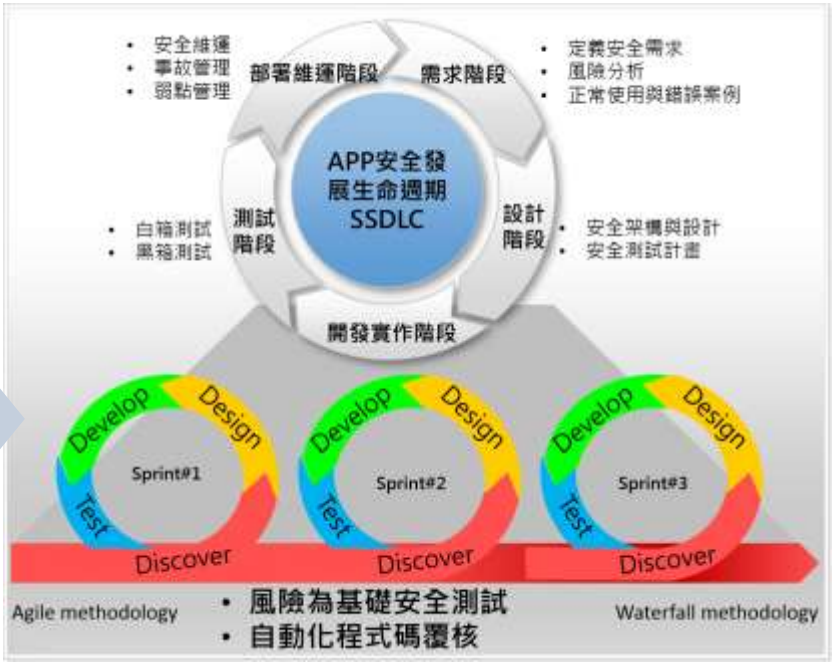
資料來源：
www.cigital.com

安全軟體生命週期比較

以TouchPoint方法論為例

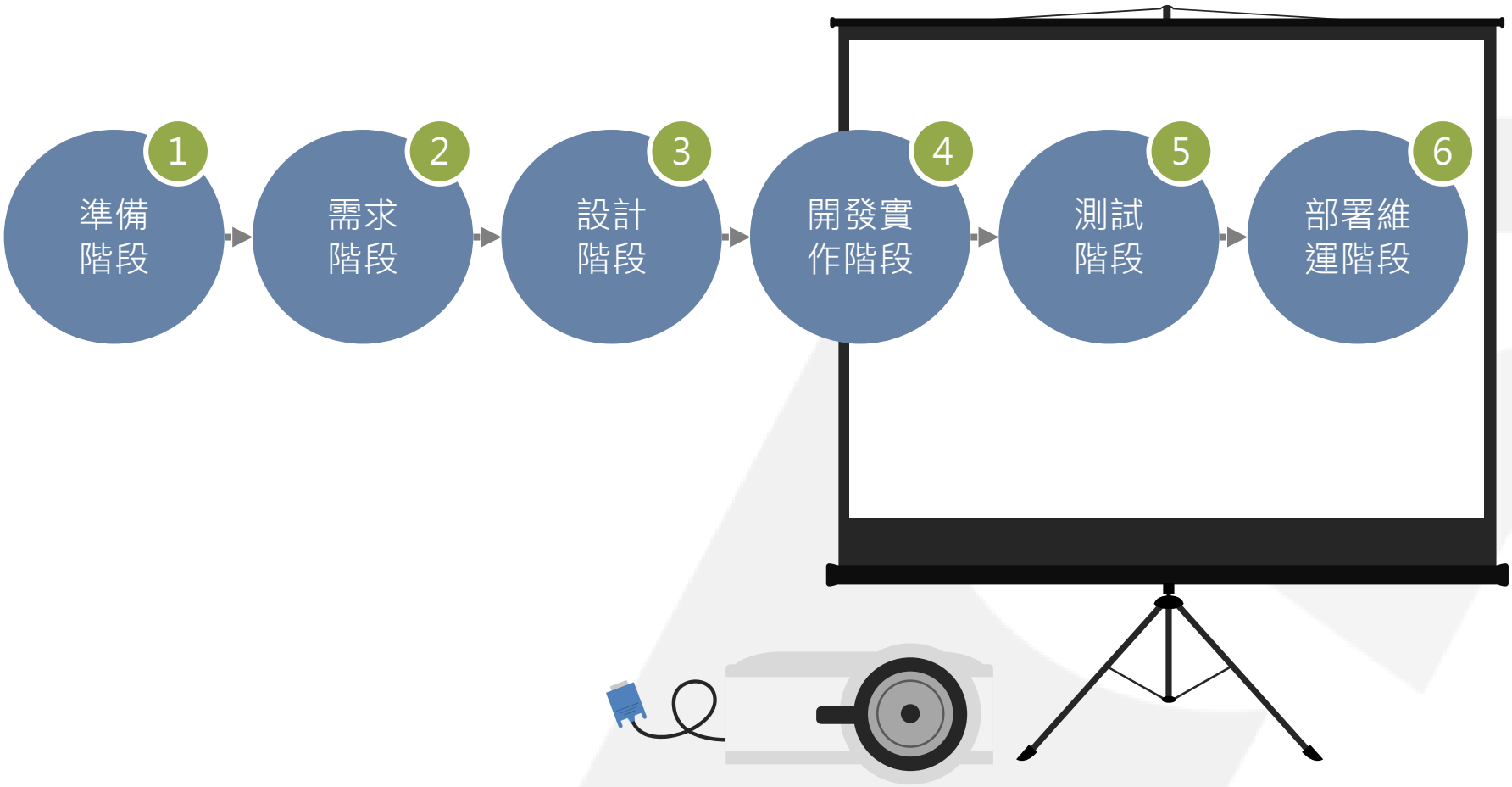
參考Software Security Building Security In, Gary McGraw, Cigital CTO

加入敏捷式開發概念



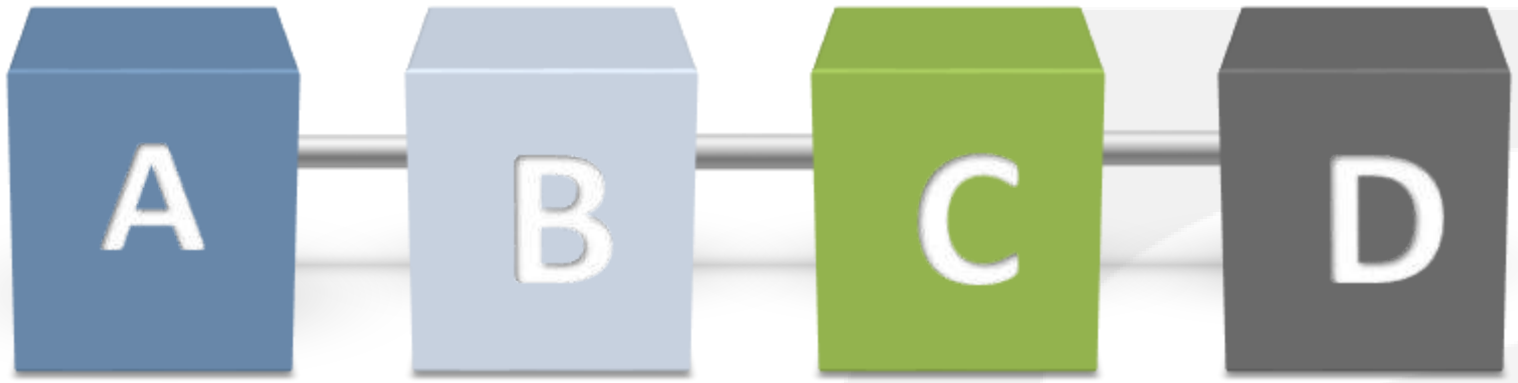
參考Software Security Building Security In，Gary McGraw,Cigital CTO & http://www.screenmedia.co.uk/blog/2014/08/what-is-agile-development-a-brief-introduction/

行動應用App安全開發生命週期各階段



1.準備階段

在行動應用App專案正式開始之前，除了準備開發環境外，還需要準備什麼？



建立安全開發流程

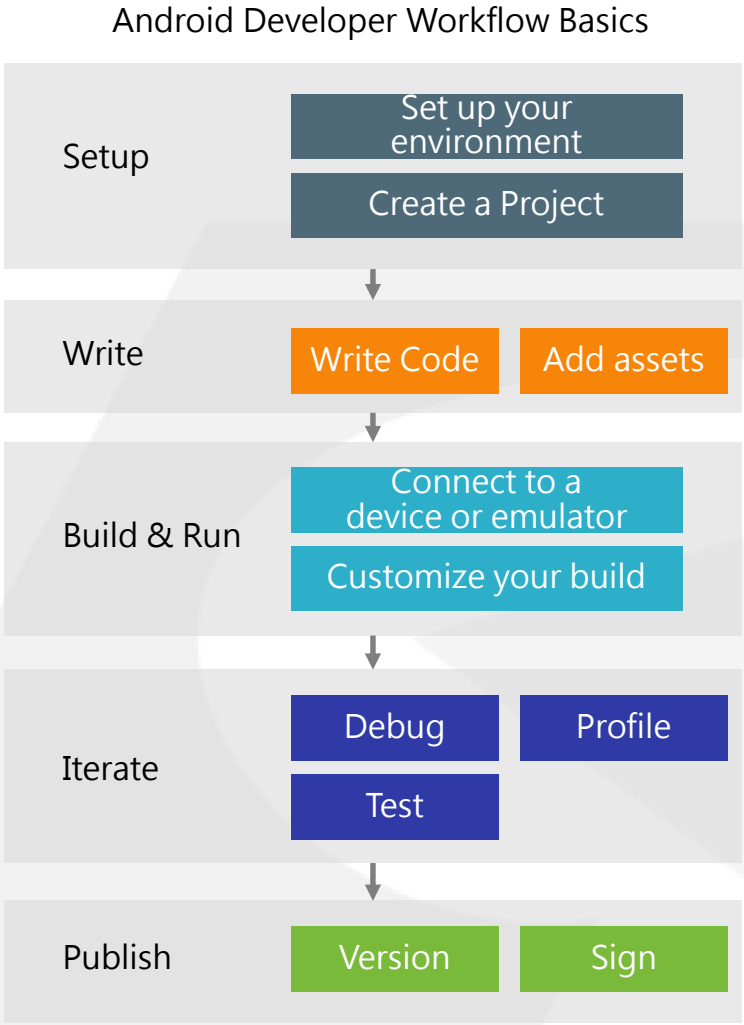
進行安全開發實務教育訓練

測試第三方API/函式庫

建立原始碼安全控管機制

A.建立安全開發流程

- 中大型的開發團隊
  - 採用傳統SDLC
    - 企業中應有完整的「應用系統開發管理規範」，包含需求單、系統分析/設計文件、程式規格書、測試報告、上線/過版申請單等文件
  - 採用敏捷式開發
    - 可運用Trello、Jira等專案及問題追蹤工具，建立專案團隊的溝通管道及確保軟體開發議題能被有效解決。
- 個人開發者
  - 可參考如Android建議的開發流程。



資料來源：Android Developer

B.執行安全開發教育訓練

行動應用App開發相關人員每年應至少參加1次以上的內部或外部訓練，範圍包含：

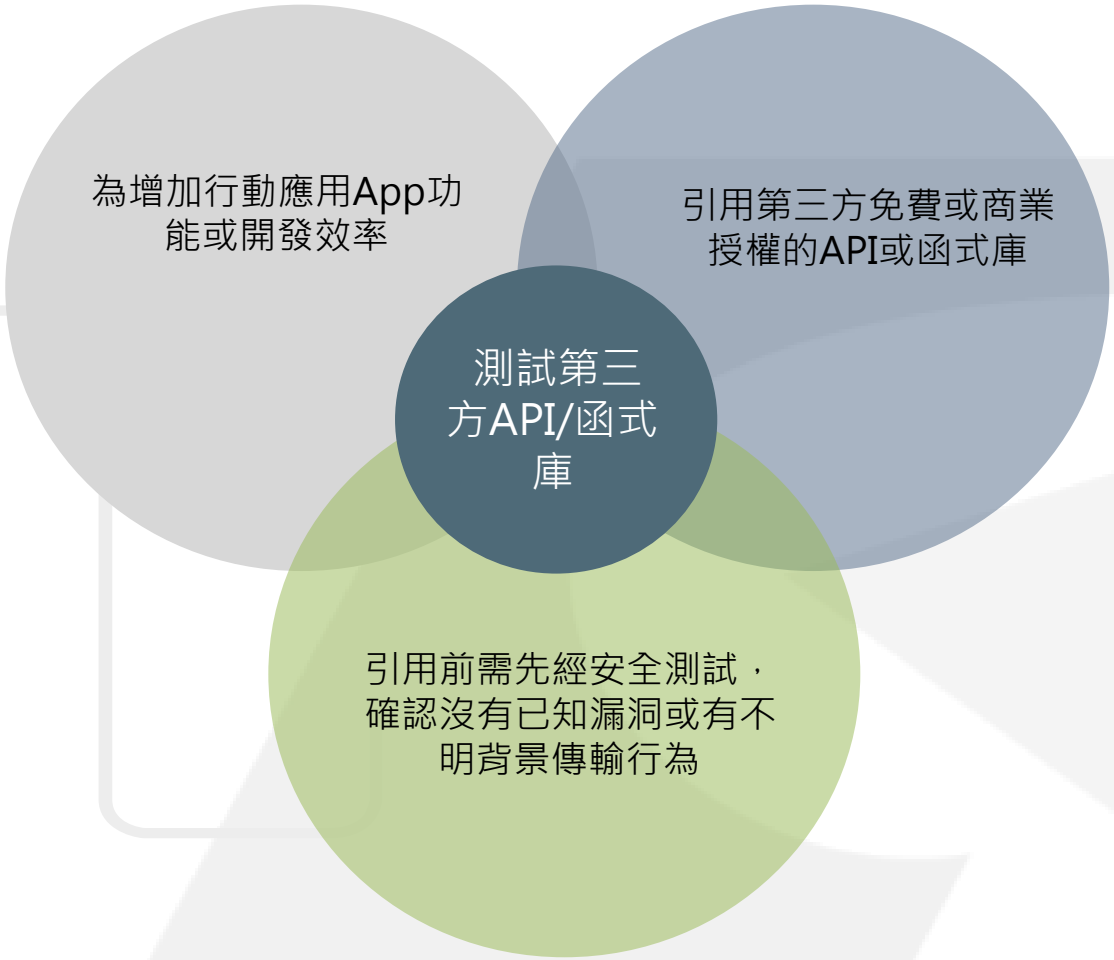
- 安全的軟體設計
- 安全的程式開發
- 安全的軟體測試
- 行動應用平台最新安全機制

技術要求 / SSDLC	4.1. 行動應用程式資訊安全技術要求事項					4.2. 伺服端資訊安全技術要求事項
	4.1.1. 行動應用程式發布安全	4.1.2. 敏感性資料保護	4.1.3. 付費資源控管安全	4.1.4. 身分認證、授權與連線管理安全	4.1.5. 行動應用程式碼安全	本規範旨在針對行動應用程式安全提出基本資訊安全要求，如行動應用程式涉及伺服端之資訊安全需求，建議應由業者自我宣告或切結其伺服端資訊安全防護與管理措施，或對於其伺服端服務之資訊安全防護與管理，出具第三方檢測通過證明。
A.需求階段	N/A	4.1.2.1. 敏感性資料蒐集 4.1.2.2. 敏感性資料利用 4.1.2.3. 敏感性資料儲存 4.1.2.4. 敏感性資料傳輸 4.1.2.5. 敏感性資料分享 4.1.2.6. 敏感性資料刪除	4.1.3.1. 付費資源使用 4.1.3.2. 付費資源控管	N/A	N/A	
B.設計階段	N/A			4.1.4.1. 使用者身分認證與授權 4.1.4.2. 連線管理機制	4.1.5.1. 防範惡意程式碼與避免資訊安全漏洞 4.1.5.2. 行動應用程式完整性 4.1.5.4. 使用者輸入驗證	
C.開發實作階段	N/A					
D.測試階段	N/A					
E.部署維運階段	4.1.1.1. 行動應用程式發布 4.1.1.2. 行動應用程式更新 4.1.1.3. 行動應用程式安全性問題回報	N/A	N/A	N/A	4.1.5.3. 函式庫引用安全	

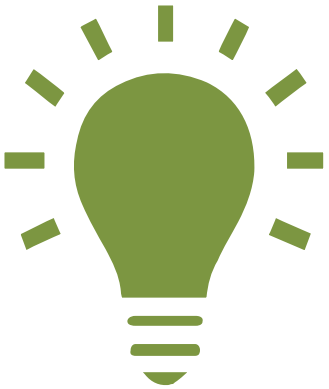
我國行動應用App基本資安規範第4章技術要求與SSDLC對應關係

C.測試第三方API/函式庫

引用不安全第三方API/函式庫，造成資安漏洞，像2014年被發現OpenSSL heart bleed漏洞，影響層面就非常廣泛。

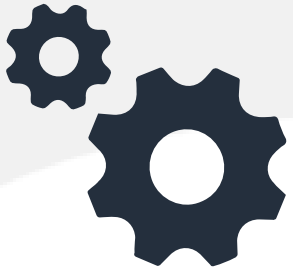


D.建立安全程式碼控管機制



經安全檢測的原始碼或API可以重複引用，以降低行動應用App漏洞發生機率。

下一步則要建立安全原始程式碼控管機制，如權限控管、簽入簽出流程、版本控管。



透過版本控管工具及變更管理機制，確保不會誤用不安全的程式碼。

2.需求階段

資安需求蒐集與分析

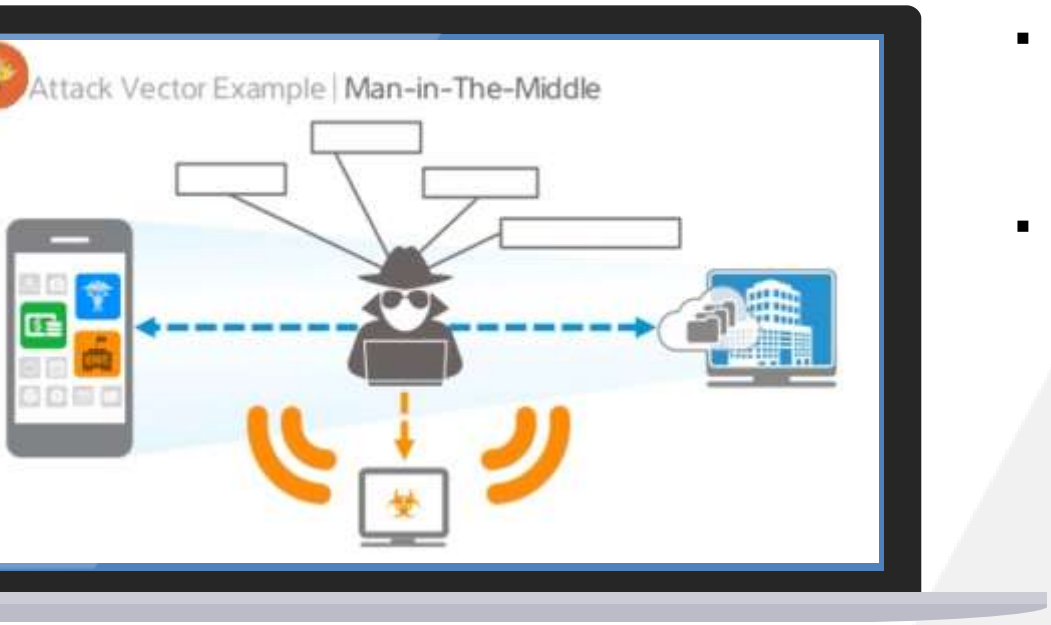
- 由PM及SA針對行動應用App整體安全需求進行評估與分析，包括識別作業系統及軟體可能面臨之安全風險，並完成安全與隱私風險評鑑，了解各種安全威脅與隱私風險狀況，定義正常與錯誤使用(濫用)案例。
- 行動應用App的資安需求的制訂，其重要性應等同功能及效能需求的制訂。
- 資安需求可包含：

- 為使開發人員(擔任PM/SA/IS角色)執行需求階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「需求階段安全檢核表」，詳見該指引附件2。

A.定義安全需求

- 是否有蒐集、處理及利用個人資料或其他敏感資料？
- 是否有來自產業的資料安全規範，如PCI DSS？
- 是否有上架當地國家資料隱私相關法規，如歐盟隱私資料保護指令？
- 軟體中敏感資料在其生命週期中的保護需求為何？
- 軟體中的敏感性資料是否有接收自或傳送至第三方的需求？
- 是否涉及付費或金流機制？
- 屬「行動應用App基本資安規範」的那一個安全分類？
- 使用對象為一般不特定消費者，還是企業內部員工？

B.風險分析



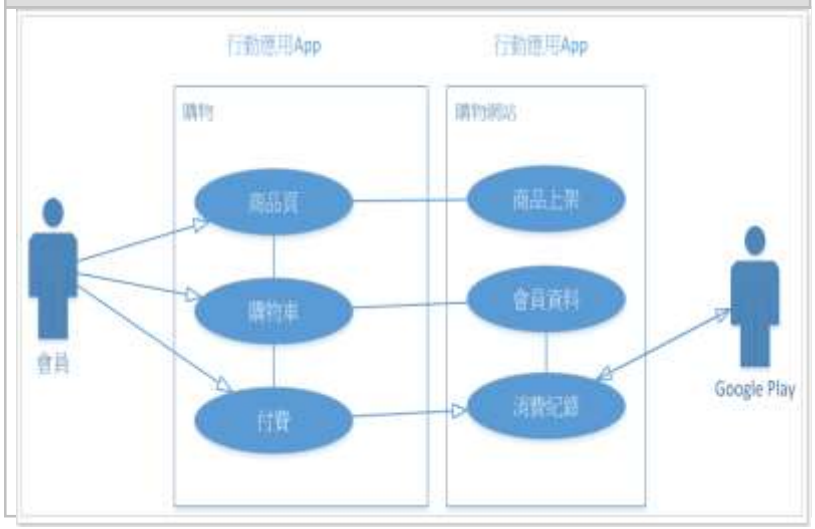
- 由PM/SA/IS與需求單位共同討論行動應用App的使用情境，以攻擊者的角度識別可能影響行動應用App與後端系統的威脅，並進行評估。
- 識別與評估威脅後，執行定性的風險分析。再對風險進行分級，對高風險事項選用適當的控制措施，以利快速形成行動應用App開發專案的安全需求重點。

C.定義正常使用與錯誤使用案例

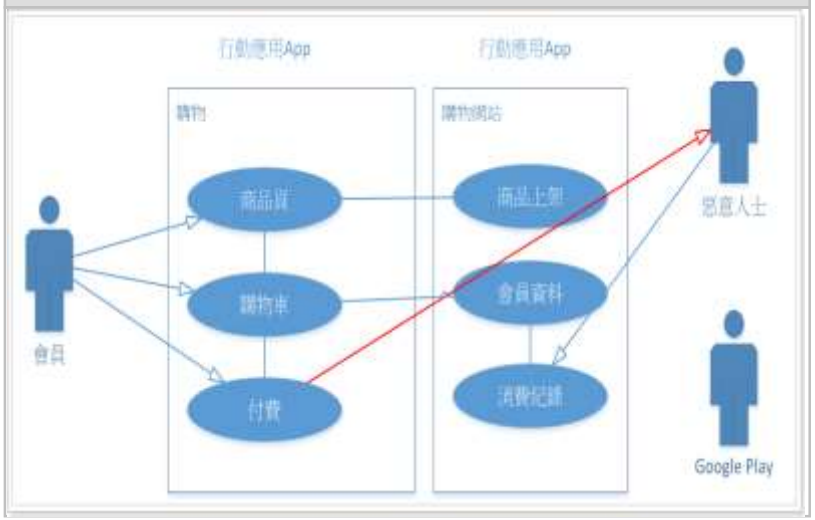
建立使用案例的目的在於將需求單位未於訪談時表達的潛在需求具體化，可透過：

- 需求(Requirements)
- 分析與設計(Analysis and Design)
- 實作(Implementation)
- 測試(Testing)

正常使用案例圖



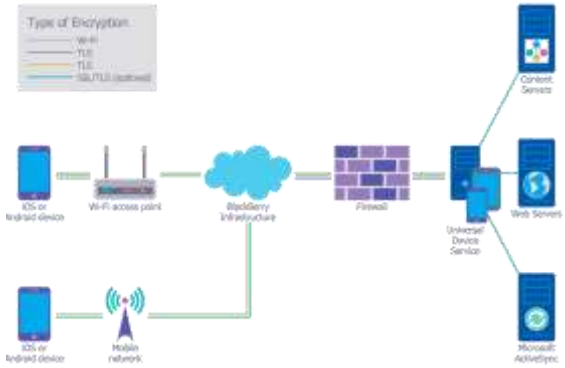
誤用案例圖



3.設計階段

- 安全與隱私防護功能應及早於系統設計初期納入，以避免於後期納入導致成本大幅的增加。
- 系統設計人員應詳細描述資訊安全的實作方法，設計過程中可與資料庫管理員及伺服器維運人員討論架構設計的可行性與安全性，包括：威脅建模、限制非必要服務、最小權限及縱深防禦等。
- 可依據「行動應用App基本資安檢測基準」建立查核點。
- 本階段應實行下列2項安全設計活動：

A.安全架構與設計



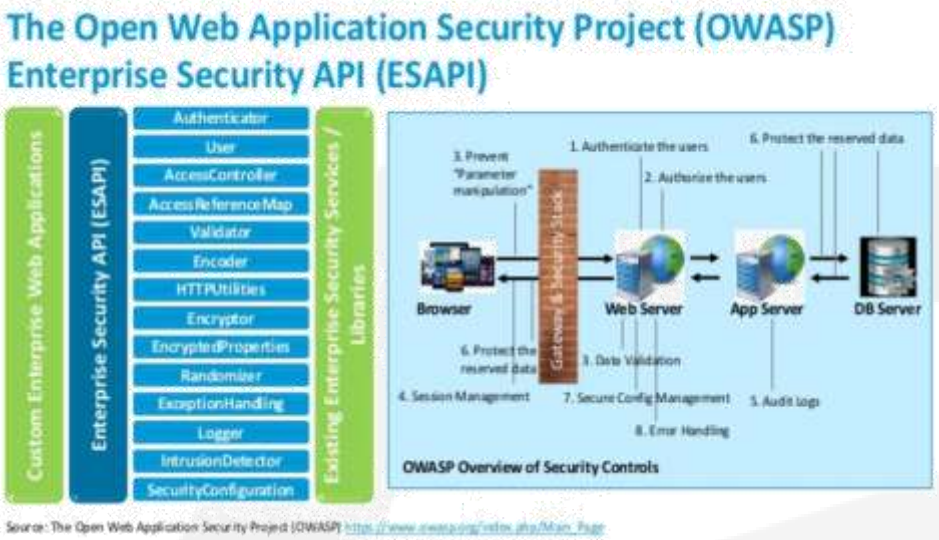
B.安全測試計畫

- 為使開發人員(擔任系統設計角色)執行設計階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「設計階段安全檢核表」，詳見該指引附件3。

A.安全架構與設計

1.安全架構師或系統設計人員依需求階段的產出， 設計系統的安全架構

- 應優先考量行動應用App作業系統的內建安全功能，例如：安全啟動鏈、Keychain、ATS及檔案加密系統。
- 其次參考iOS的安全開發指南建議及「行動應用App安全開發指引」的安全實務，進行App本地端及伺服器間安全設計。



Source: The Open Web Application Security Project (OWASP) https://www.owasp.org/index.php/Main_Page

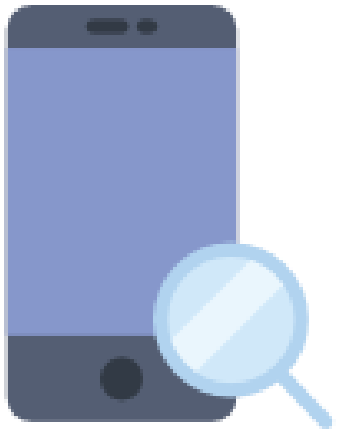
2.系統架構與安全設計，至少需包括：

- 系統架構圖
- 儲存區分配規劃
- 資料儲存安全設計
- 資料傳輸安全設計
- 安全界面設計
- 資料消除設計
- 權限設計
- App間安全通訊設計

3.對敏感性資料的保護，應以整個資料生命週期進行考量，選用符合法規及產業規範的控制措施。

B.安全測試計畫

在完成系統架構與安全設計後，應依其設計擬訂以風險為基礎的安全測試計畫，以利在開發實作完成後，進行安全需求驗證。

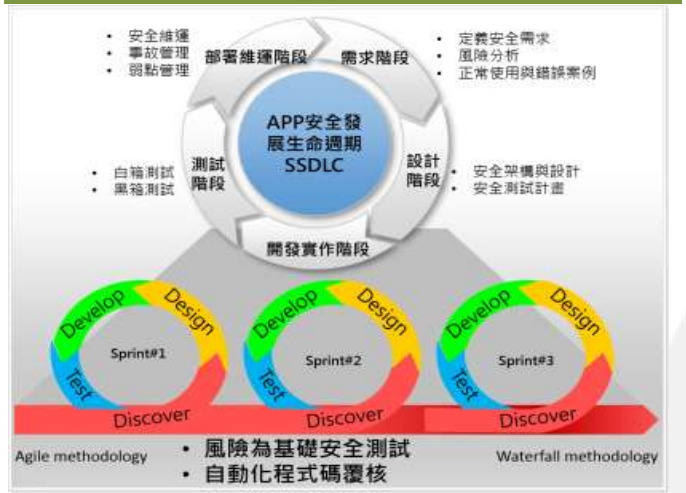


測試計畫內容	
測試目標	資源與時程規劃
測試策略 – 風險分析 – 測試方法 – 測試範圍 – 測試環境	測試報告格式
測試程序及接受標準	應變措施

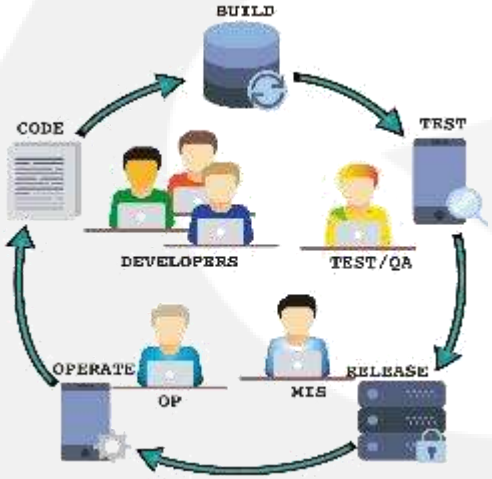
4.開發實作階段

- 建立經認可的開發工具清單，對於所有使用之函式庫皆審查其安全歷史，並研擬安全之替代方案。
- 程式開發人員需依行動應用App安全開發實務撰寫安全程式碼並實施單元測試，可依據「行動應用App基本資安檢測基準」建立查核點，同時應執行程式碼靜態分析，可使用靜態自動分析工具或由主管或不同開發人員實施必要之人工審核。
- 本階段簡介下列2項開發人員應該要知道的基礎知識：

瀑布式SSDLC與
敏捷式開發



行動應用App之
DevOps開發流程



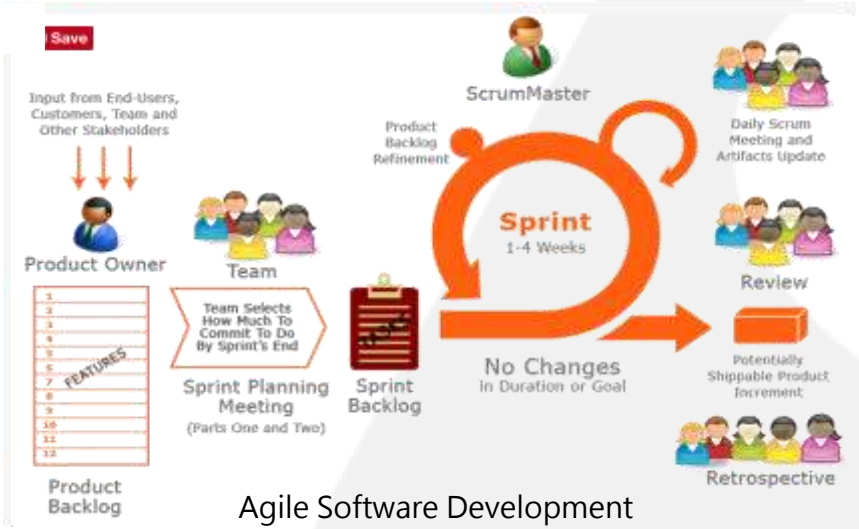
- 為使開發人員執行開發實作階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「開發實作階段安全檢核表」，詳見該指引附件4。

A.瀑布式SSDLC 與敏捷式開發

1.在敏捷式開發中，軟體專案的構建被切分成多個子項目，各個子專案的成果都經過測試，具備整合和可運行的特性。

2.由麻省理工學院史隆管理學院評論(MIT Sloan Management Review)所刊載的一篇為時2年對成功軟體開發專案的研究報告：

- 大多數的成功軟體開發專案是使用反覆循環的開發方式，而不是瀑布式過程。因在反覆循環的開發方式下，開發團隊原則上每天至少會有1次(以上)的機會，把新程式碼整併至既有系統中，並通過測試，如此頻繁的變更，可讓團隊更快的做出必要的反應。
- 開發團隊具備運作多個產品的工作經驗。
- 很早就致力於構建和提供內聚的架構。



Agile Software Development

資料來源：http://www.essentialsln.com/agile-software-development/

3.建議在進行敏捷式開發時，開發者應熟悉相關的安全實務，再輔以靜態自動分析工具，才能維持敏捷的特性。

B.行動應用App之DevOps開發流程

- DevOps一詞係來自Development和Operations的組合詞。
- DevOps是一種用來促進「軟體開發人員」和「IT運維技術人員」之間的溝通、合作與整合的過程及方法的統稱。
- 近年來不論大型或小型的開發團隊皆廣為採用。

參考Suzie Prince於2016年2月11日發表的「The Product Managers' Guide to Continuous Delivery and DevOps」一文中，說明下列3個重要概念。

持續整合 Continuous Integration	持續交付 Continuous Delivery	持續部署 Continuous Deployment

ithome DevOps專區：http://www.ithome.com.tw/devops

5.測試階段

- 為確保行動應用App之需求、規格及安全性功能如預期運作，需針對開發完成的App進行檢測。
- 本階段將帶領學員了解：

A.檢測目標

B.檢測流程

C.檢測方法

- 為使開發人員執行測試階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「測試階段安全檢核表」，詳見該指引附件5。

A.行動應用App檢測目標

進行「驗證設計規格」與「確認符合需求」

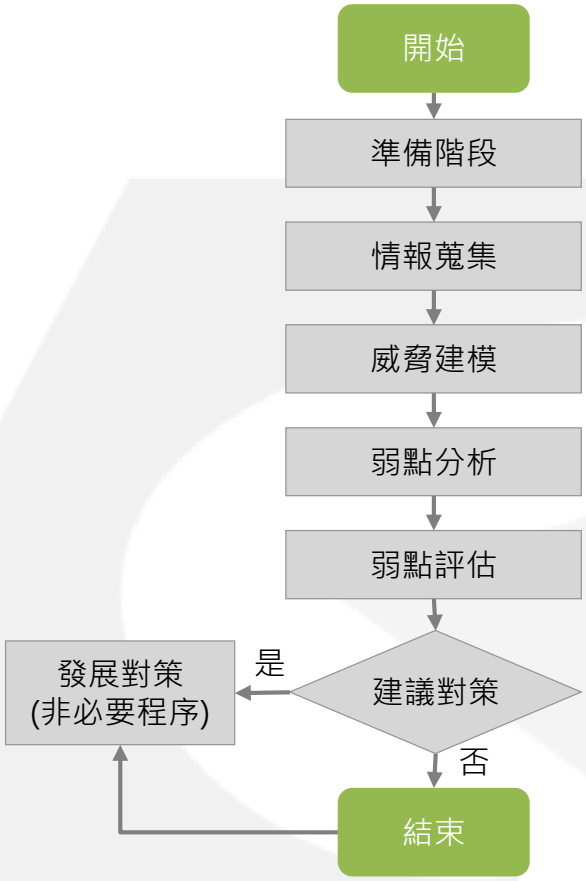
除確保其符合客戶之功能或非功能性要求，安全規格亦是當前政府與使用者重視的要點。

	驗證	確認
ISO/IEC 12207	• 檢查軟體產品並提出客觀證據，以證實達成訂定的規格要求。 • 判斷某發展階段的軟體，達成前項發展階段訂定的需求或限制。	• 檢查軟體產品並提出客觀證據，以證實達成某一特定預期功用的需求 • 確定依據需求規格製作的最終軟體產品，是否滿足特定使用目的。
CMMI相關指引	確保工作產品符合其指定需求的規格。	在展示最終軟體產品或產品元件在需求環境中，實現客戶需要的產品。

B.行動應用App檢測流程

OWASP的行動應用App的安全測試指引

參考採用之基本App核心檢測流程，透過App資料擷取、威脅建模及弱點分析等流程，最後以指引說明或工具檢測等方式，獲得自我檢測結果與建議。



資料來源：The OWASP Foundation

C.行動應用App檢測方法(1/2)

- 參考McGraw所提出的TouchPoint Model。
- 透過「需求制定與使用者案例」、「架構設計」、「檢測計畫」、「程式碼」、「測試與結果」、「檢測回饋」等面向，提出7種最佳檢測實務。
  - 源碼檢測(以使用工具為主)
  - 風險分析
  - 滲透測試
  - 以風險為基礎之安全測試
  - 濫用案例(Abuse cases)
  - 安全需求制定
  - 安全操作

TouchPoint Model方法論

資料來源：https://www.cigital.com/presentations/ARA10.pdf

C.行動應用App檢測方法(2/2)

3種行動應用App弱點分析技術

靜態方法

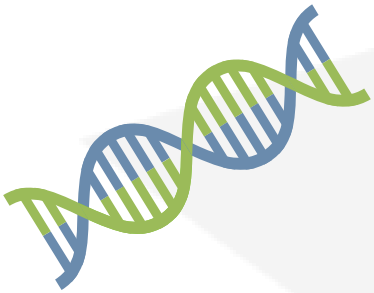
- 在不需要執行App的情況下，確認需求與規格是否符合預期。
- 常見的執行方法為檢視(Inspection)、結構化逐步審查(Structured Walkthrough)與主動審查(Active Review) 。
- 可針對標的為App產品計畫、需求規格文件與程式碼等內容。
- 使用工具如dex2jar、otool、androwarn、Flawfinder等。

動態方法

- 觀察App的活動細節是否有異常。
- 常見的執行方法為黑箱測試(Black Box)與白箱測試(White Box)。
- 針對App的各項介面、功能與非功能需求，以及相關執行邏輯等正確性。

鑑識方法

- 透過時間軸及Log分析，驗證與確認App之安全規格與行為合理性。



6.部署維護階段

- 行動應用App進入部署維運階段時
  - MIS人員應先對作業系統、伺服器、資料庫及軟體本身等部署環境進行安全檢視工作，包括使用正確的安全設定、關閉非必要之服務與網路埠、移除測試帳號與資料，以及使用最低權限執行等。
  - 遇有軟體或作業系統環境變動時，應做好變動管理工作，包括識別可能變動及分析變動對安全性之影響，另應進行持續性之監測，以驗證軟體運作安全性。

- 為使開發人員執行部署維運階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「部署維運階段安全檢核表」，詳見該指引附件6。

A.安全維運

安全組態管理

行動應用App以雲端服務模式呈現時，應強化伺服端系統管理。

特別強化下列安全管理領域：

- 系統管理：系統更新、關閉不必要服務、限制軟體安裝。
- 網路管理：網路配置、防火牆設定。
- 模組元件管理：版本控管、取得來源管理。
- 伺服器管理：帳號權限管理、加密連線設置、監控與稽核。
- 應用程式：帳號權限管理、開發與測試過程所使用的功能應移除或關閉。

安全部署管理

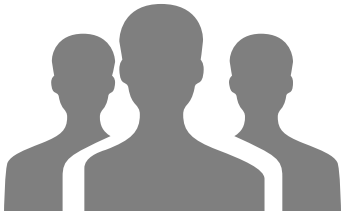
根據「系統管理」、「網路管理」、「伺服器管理」、「模組元件管理」及「伺服應用程式管理」等項目類別，進行各個類別的部署管理並具體採用前述之工具與檢核表進行安全項目自我檢測，以完成伺服端之安全部署管理。

亦可採用滲透測試進行檢測，可參考：

- OWASP
- IECOM
- SANS

B.事故管理

儘管已執行相關的資安檢核與設置，仍有發生資安事故之可能



組織應考量：

- 資訊安全通報及應變程序
- 資訊安全弱點回報分析措施
- 資訊安全改善與矯正措施

詳細資料亦可參考ISO/IEC 27001與ISO/IEC 20000等標準

C.弱點管理

1 管理規範	2 技術強化	3 持續落實
應制定整體弱點管理政策、程序落實及執行與後續追蹤及改善等，並進行相關弱點稽核與紀錄。	在弱點管理上可建置或安裝相關管理系統，如防毒軟體、防火牆與入侵偵測防禦系統等，協助相關弱點之即時偵測防禦與追蹤紀錄管理。	當弱點被偵測且分析後，應進行相關後續修補與更新動作。

課程大綱

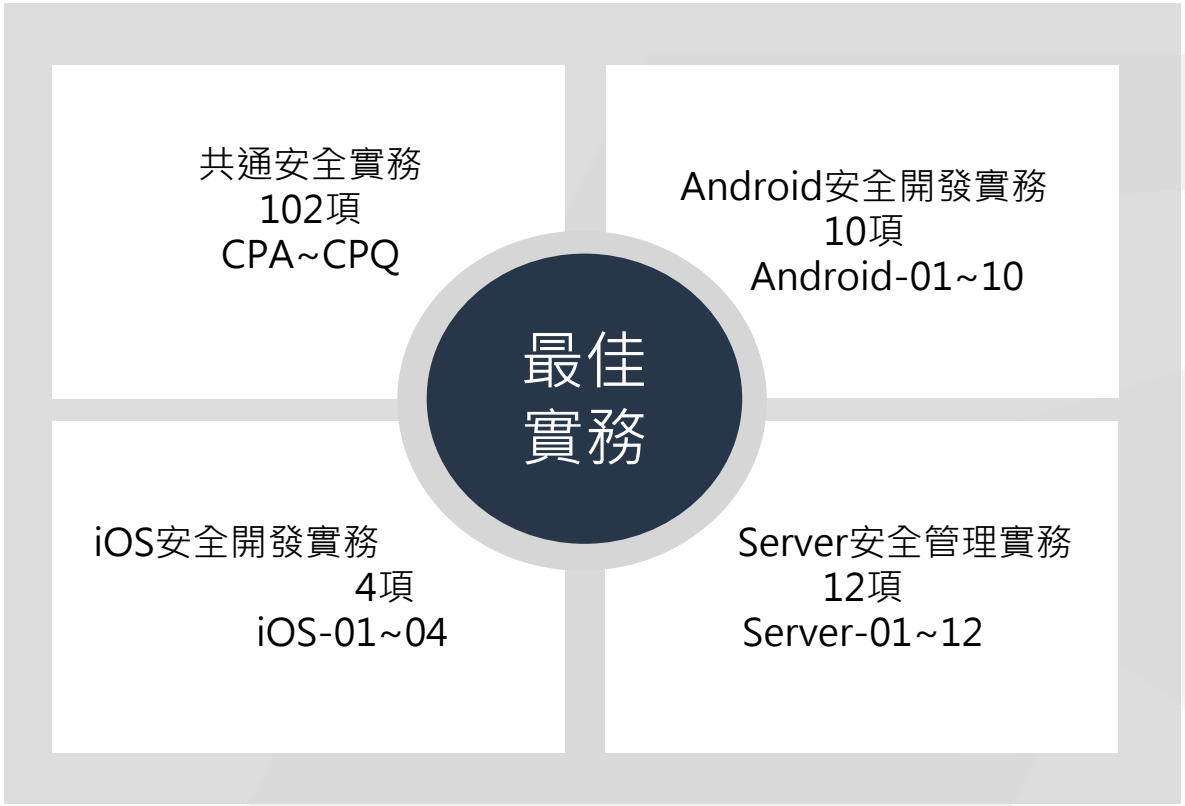
第1單元	行動應用App安全基礎概論	0.5小時
第2單元	安全軟體開發生命週期	0.75小時
第3單元	**安全行動應用設計最佳實務**	**1.25小時**
第4單元	行動應用App安全檢測流程與工具	0.5小時

行動應用App安全開發最佳實務

依據我國經濟部工業局「行動應用App基本資安規範」及「行動應用App基本資安檢測基準」及彙整中國大陸、歐盟、日本、美國及CSA行動應用App安全開發實務要點。

4
個實務類別

128
項實務準則



實務準則一覽表(以Android之App開發為主)

102 項共通實務

10 項Android實務

12 項Server實務

共通安全開發實務準則(類別)	
A	行動應用程式發布
B	行動應用程式更新
C	行動應用程式安全性問題回報
D	敏感性資料蒐集
E	敏感性資料利用
F	敏感性資料儲存
G	敏感性資料傳輸
H	敏感性資料分享
I	敏感性資料刪除
J	付費資源使用
K	付費資源控管
L	使用者身分認證與授權
M	連線管理機制
N	防範惡意程式碼與避免資訊安全漏洞
O	行動應用程式完整性
P	函式庫引用安全
Q	使用者輸入驗證

Android安全開發實務 (類別)	
1	N.防範惡意程式碼與避免資訊安全漏洞
2	N.防範惡意程式碼與避免資訊安全漏洞
3	N.防範惡意程式碼與避免資訊安全漏洞
4	N.防範惡意程式碼與避免資訊安全漏洞
5	N.防範惡意程式碼與避免資訊安全漏洞
6	N.防範惡意程式碼與避免資訊安全漏洞
7	N.防範惡意程式碼與避免資訊安全漏洞
8	N.防範惡意程式碼與避免資訊安全漏洞
9	Q.使用者輸入驗證
10	Q.使用者輸入驗證

伺服器安全實務	
1	作業系統強化與記錄留存
2	作業系統強化與記錄留存
3	網頁服務安全
4	網頁服務安全
5	網頁服務安全
6	網頁服務安全
7	網頁服務安全
8	網頁服務安全
9	網路安全防護
10	網路安全防護
11	網路安全防護
12	網路安全防護

由於時間因素，以下將挑選幾項最佳實務進行說明。

行動應用程式發布(A)-CPA-01

CPA-01 於行動應用程式商店提供及應用程式內實作提供隱私權政策說明連結，說明欲存取之敏感性資料、行動裝置資源及宣告權限用途。

簡介

- 使用者下載安裝行動應用App時，除了功能及價格因素，還會考量隱私的議題。
- 事前規劃並撰寫此行動應用App的隱私權政策。
- 可參考Center for Democracy & Technology (CDT)的行動應用程式開發最佳實務的隱私權宣告範本(https://www.cdt.org/files/pdfs/Apps%20Best%20Practices%20v%20beta.pdf) 。

隱私權政策的參考範本如下：

- 隱私權保護政策的適用範圍
- 個人資料的蒐集、處理及利用方式
- 資料之保護
- 網站對外的相關連結
- 與第三人共用個人資料之政策
- Cookie之使用
- 隱私權保護政策之修正

開發生命週期	需求階段、開發實作階段、部署維運階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.1.1	行動應用App基本資安檢測基準	4.1.1.1.2

行動應用程式安全性問題回報(C)-CPC-01

CPC-01 於可信任之應用程式商店或行動應用程式內，提供聯絡網頁、電子郵件、電話或其他類型聯絡方式。

簡介

- Google Play的商店資訊及iTunes Connect App屬性的版本資訊中，必需填寫開發人員的聯絡網頁、電子郵件、電話或其他類型聯絡方式，並注意此資訊應定期維護。
- 如果是中大型開發商，建議使用客服專線或是公務用的固定聯絡方式，避免以登記個別開發人員之聯絡資訊，以避免當該開發人員職位異動時，開發商無法接收到客訴或安全性通報資訊。

開發生命週期	部署維運階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.1.3	行動應用App基本資安檢測基準	4.1.1.3.1

敏感性資料儲存(F)-CPF-01

CPF-01 行動應用App如需要儲存敏感性資料需依CPD-02規劃於可信任之應用程式商店或行動應用程式內聲明。

簡介

- 行動應用App如需要儲存”敏感性資料”參考CPA-01將行動應用App需要儲存敏感性資料於Google Play及App Store及行動應用App的隱私權政策中聲明。

敏感性資料(Sensitive Data)
指依使用者行為或行動App之運作，建立或儲存於行動裝置及其附屬 儲存媒介之資訊，而該資訊之洩漏有對使用者造成損害之虞，包括但不限 於個人資料、通行碼、即時通訊訊息、筆記、備忘錄、通訊錄、地理位置、行事曆、通話紀錄及簡訊。

開發生命週期	設計階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.3	行動應用App基本資安檢測基準	4.1.2.3.1

敏感性資料儲存(F)-CPF-07

CPF-07 敏感性資料或包含敏感性資料的日誌檔，除非已加密，應避免儲存於與其他行動應用App共用或全域可讀寫儲存區域或外部儲存媒體。

簡介

- 其他行動應用App共用或全域可讀寫儲存區域或外部儲存媒體均是未設任何權限且明文儲存資料區域，將敏感性資料或相關日誌儲存前述區域極易造成敏感性資料外洩。
- Android預設是將資料儲存於行動應用App的私有儲存區域，其他的App無法存取它們，當App被卸載時會被一併刪除，呼叫openFileOutput()與該文件的名稱和操作模式，MODE_PRIVATE將創建文件，並使其專用於您的應用程式。其他可用的模式有：MODE_APPEND，MODE_WORLD_READABLE，和MODE_WORLD_WRITEABLE。

  注意：參數MODE_WORLD_READABLE和MODE_WORLD_WRITEABLE自API等級17以後已不能用於Android N，使用它們開始將導致SecurityException錯誤訊息被拋出。這意味著Android N和更高版本不能通過檔案名稱共享的私有檔案，並試圖共享一個“file：//”URI將導致 FileUriExposedException錯誤訊息被拋出。如果您的應用程式需要與其他應用程式共享的私有檔案，它可以使用FileProvider在FLAG_GRANT_READ_URI_PERMISSION申請權限。另請參閱共享檔案。

- iOS現階段並無法將沙箱的私有資料像Android儲存在沙箱之外在全域可讀寫的區域。

開發生命週期	開發實作階段		
不安全程式碼範例	參考補充講義	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.2.3	行動應用App基本資安檢測基準	4.1.2.3.6

敏感性資料分享(H)-CPH-01

CPH-01 敏感性資料分享給不同行動應用App應實作使用者同意或拒絕選項，提示使用者選擇時機可於(1)安裝時(2)當敏感資料被儲存或傳送前(3)預設設定為關閉同意，需使用者自行開啟。

簡介

- 參考CPA-01將行動應用App需要分享敏感性資料類別於Google Play及App Store及行動應用程式的隱私權政策中聲明。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.5	行動應用App基本資安檢測基準	4.1.2.5.1(1) 4.1.2.5.1(2)

使用者身分認證與授權(L)-CPL-01

CPL-01 行動應用App應設計並實作適當身分認證機制，並依使用者身分授權，以防止敏感資料被非授權人員存取。

簡介

- 開發實作人員需以「使用者意願」及「最小權限」為主要安全原則。
- 在App運行先後順序上以「1.識別使用者身分」、「2.徵詢使用者意願(企業存取政策)」及「3.取得權限」，所以使用者身分認證是行動應用App取用敏感性資料權限的必要機制。
- 行動應用App設計及開發人員應視法規及實務需求，設計且實作適當身分認證機制。

常見行動應用App身分認證方式

- 行動智慧裝置解鎖PIN碼
- 行動智慧裝置內建生物辨識(指紋、虹膜、面部、聲紋)
- 雲端服務認證：Apple ID、Google ID、Microsoft Live ID
- 開放式認證：OAuth 2.0
- App自建私有身分認證
- 企業整合認證：Microsoft AD、Novell LDAP

常見實作的技術

- 帳號 / 密碼
- 觸控螢幕滑動手勢
- PKI
- 多因素認證
- 生物辨識

現行最常運用的認證機制為OAuth 2.0，使用者可以使用Google、Facebook及Windows Live。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.4.1	行動應用App基本資安檢測基準	4.1.4.1.1 4.1.4.1.2

連線管理機制(M)-CPM-01

CPM-01 行動應用App實作CPG-02 TLS連線，需使用編碼長度為128位元(含)以上之交談識別碼(Session ID) 。

簡介

- 攻擊者可以在經過攫取分析一系列的ID值，破解交談識別碼。
- 交談識別碼必須足夠長，以防止暴力攻擊，並驗證有效交談的存在。
- 交談識別碼長度必須至少為128位元(16位元組)，但該數目不應該被認為是絕對的最小值，作為其他實施因素可能影響其強度。例如，有公認良好的實作方式，如Microsoft ASP.NET中，利用120位元的隨機數為它的交談識別碼，可以提供很好的有效熵，和(20字符的字符串表示)其結果，可以認為足夠長的時間，以避免猜測或暴力攻擊。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.4.2	行動應用App基本資安檢測基準	4.1.4.2.1(1)

連線管理機制(M)-CPM-03

CPM-03 行動應用App連線使用交談識別碼，應實作具備逾時失效(Session time-out)機制。

簡介

- 由於行動智慧裝置經常丟失或被盜，並且攻擊者可能利用一個應用程式來存取敏感資料，執行交易或研究設備所有者的帳戶。尤其是銀行或交易類的應用程式。建議行動應用App，在登入後也進行時間控管，以加強安全性。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.4.2	行動應用App基本資安檢測基準	4.1.4.2.1(3)

連線管理機制(M)-CPM-05

CPM-05 行動應用程式應使用憑證綁定(Certificate Pinning)方式驗證並確保連線之伺服器為行動應用程式開發人員所指定。

簡介

- 在全球漏洞資料庫網站揭露一個CVE(Common Vulnerabilities and Exposures)漏洞，漏洞編號為CVE-2014-6693。CVE漏洞報告指出，因為行動應用App沒有驗證x.509的CA(Certificate Authority)憑證，駭客可以藉此發動中間人攻擊，以竊取使用者敏感性資料。
- 開發人員應使用憑證綁定(Certificate Pinning)的方式，把需要比對的信任發行者發行憑證預先存放在App裡，指定特定Domain就只能使用特定憑證，等到要進行SSL Handshake的時候，再與伺服器的憑證進行比對。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.4.2	行動應用App基本資安檢測基準	4.1.4.2.2(2)

防範惡意程式碼與避免資訊安全漏洞(N)-CPN-01

CPN-01 如有程式碼，使用可檢測行動應用App原始碼安全檢測工具或人工進行靜態分析，檢視權限並比對是否與CPD-03安全設計及使用者設定權限相符(permission mapping analysis)。

簡介

- 行動應用程式的安全性審查可透過程式碼檢測工具或人工程式碼檢視進行審查，此過程需按照ISO/IEC 17025標準進行，且需審查之項目應考量相關對應之標準或規範施行檢測。
- 建議提供安全需求項目與理想檢核結果並加以說明，在檢核過程中不強制規範使用靜態或動態方法或兩者並用，作為履行審核項目要求的方法組合。

- 測試人員執行行動應用App原始碼安全檢測，並以工具或人工進行靜態分析時，應檢視權限並比對是否與CPD-03安全設計及使用者設定權限相符。

開發生命週期	測試階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.5.1	行動應用App基本資安檢測基準	4.1.5.1.1(5)

函式庫引用安全(P)-CPP-01

CPP-01 行動應用App使用第三方函式庫前，需先確認其是來自可靠來源、有持續更新並經測試沒有漏洞、後端木馬及不明傳送目的地。

簡介

- 行動應用App當引用第三方函式庫時：
  - 需了解其來源是否可靠性。
  - 確認更新到最新版本的並進行使用前測試。
  - 需注意漏洞及檢測是否有不明伺服器端連線。
- 若自行編譯函式庫時，也需注意元件組成是否安全，是否有惡意程式碼被夾帶進來。
- 著名的XCodeGhost事件，或者更多的駭客、病毒的注入，都是類似的作法。包含Android及iOS的編譯紀錄均應詳細檢查。在Unix-like的作業系統，需注意LD_PRELOAD的使用，並可透過set -x的方式，檢查編譯紀錄是否有惡意的注入。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.5.3	行動應用App基本資安檢測基準	4.1.5.3.1

使用者輸入驗證(Q)-CPQ-05

CPQ-05 行動應用App應實作過濾使用者輸入及伺服器端傳入資料中易導致SQL injection之字串。

簡介

- 由於SQL命令，若欄位參數採用字串組合，遇到特殊字元或惡意攻擊，會有安全漏洞，必須改用元件提供之欄位參數輸入方式。
- 於CPQ-03也說明相關解決方式：使用Query parameterization。

開發生命週期	開發實作階段		
不安全程式碼範例	參考補充講義	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.5.4	行動應用App基本資安檢測基準	4.1.5.4.2(1)

使用者輸入驗證(Q)-CPQ-08

CPQ-08 行動應用App應實作過濾使用者輸入及伺服器端傳入資料中易導致XML Injection之字串。

簡介

- XML注入攻擊者會嘗試在SOAP訊息中插入各種字串，目標在對XML結構注入各種XML標籤。通常一個成功的XML注入攻擊將導致限制操作的執行。根據各種執行的操作，而衍生各種安全問題。
- 由於XML的應用，在存取XML資料的時候，也須避免如SQL inject的處理方式而造成的漏洞。或基於XML的衍生功能與漏洞進行限制，例如XML External Entity (XXE)。

開發生命週期	開發實作階段		
不安全程式碼範例	參考補充講義	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.5.4	行動應用App基本資安檢測基準	4.1.5.4.2(1)

Android-01

Android-01 Android版本行動應用App應謹慎實作Intents，避免資訊不慎外洩遭惡意運用。

簡介

- 意圖元件(Intents)是使用在內部元件信號傳遞及以下功能:
  1.開啟另一個使用者界面，啟動另一個Activity。
  2.作為廣播事件，以通知系統和應用程式特定狀態的改變。
  3.啟動、停止背景程式及與其溝通。
  4.經由ContenetProviders來存取資料。
  5.扮演回調(Callbacks)來處理相關連的事件。

- 不適當的實作有可能會造成資料洩密、限制使用的功能被不當使用與程式執行流程被調整或繞過。

開發生命週期	開發實作階段		
不安全程式碼範例	參考補充講義	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.5.1	行動應用App基本資安檢測基準	N/A

Android-09

Android-09 行動應用App應實作過濾使用者輸入及伺服器端傳入資料中易導致Intent Injection之字串。

簡介

- 防止意圖欺騙：
  1. 限制通訊：Android平台有兩個控制元件限制可以控制權限使用相對應權限的應用程式，以防止與應用程式通信惡意應用，以及意圖類型使用隱式意圖產生更多的安全問題。
  2. 驗證輸入並假定輸入是不一定是來自可信賴的來源以及驗證每個應用程式的輸入。
  3. 設定值在外部的Android的AndroidManifest設定值：避免接收到惡意的意圖出口組件。

開發生命週期	開發實作階段		
不安全程式碼範例	參考補充講義	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.5.4	行動應用App基本資安檢測基準	4.1.5.4.2(7)

Server-01

Server-01 與行動應用App連接之所有後端服務伺服器(包含網頁、資料庫及中介等)作業系統應有效強化及進行安全設定配置，並持續進行安全性程式修補。

簡介

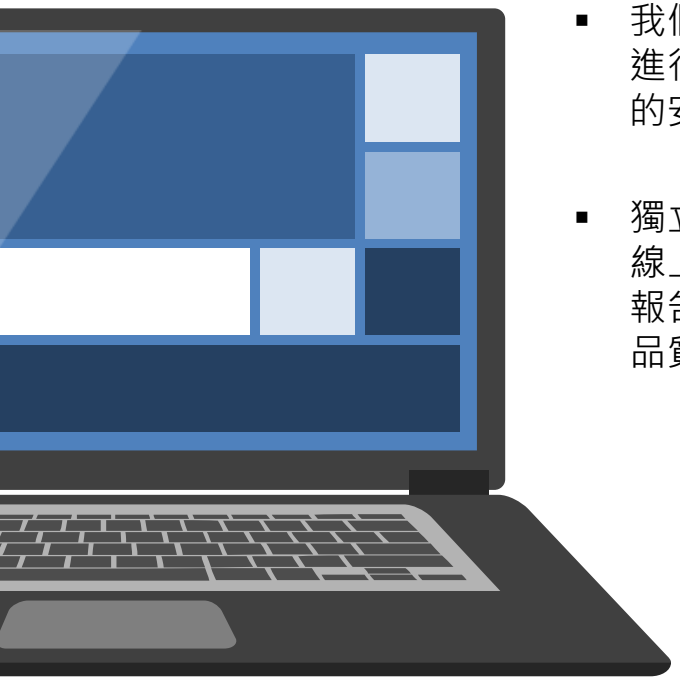
- 減少資訊洩露
  - 攻擊者可因為獲得片段有用的伺服器資訊，提高攻擊成功的機率。
  - 正式環境中應避免揭露過於詳細的錯誤訊息，例如網頁的元件、版本、作業系統等。
  - 改善建議：降低Apache的版本資料、刪除某些預設的路徑或特別的安裝路徑。此外，管理者功能路徑，除非必要，否則不應提供公開存取(加入安全限制)。
- 強制使用HTTPS機制
  - 於網頁主機使用HTTPS機制(在header加上“Strict-Transport-Security”)，以保護連線。

開發生命週期	部署維運階段		
不安全程式碼範例	參考補充講義	安全程式碼範例	參考補充講義
行動應用App基本資安規範	N/A	行動應用App基本資安檢測基準	N/A

課程大綱

第1單元	行動應用App安全基礎概論	0.5小時
第2單元	安全軟體開發生命週期	0.75小時
第3單元	安全行動應用設計最佳實務	1.25小時
第4單元	行動應用App安全檢測流程與工具	0.5小時

行動應用App資安檢測實務

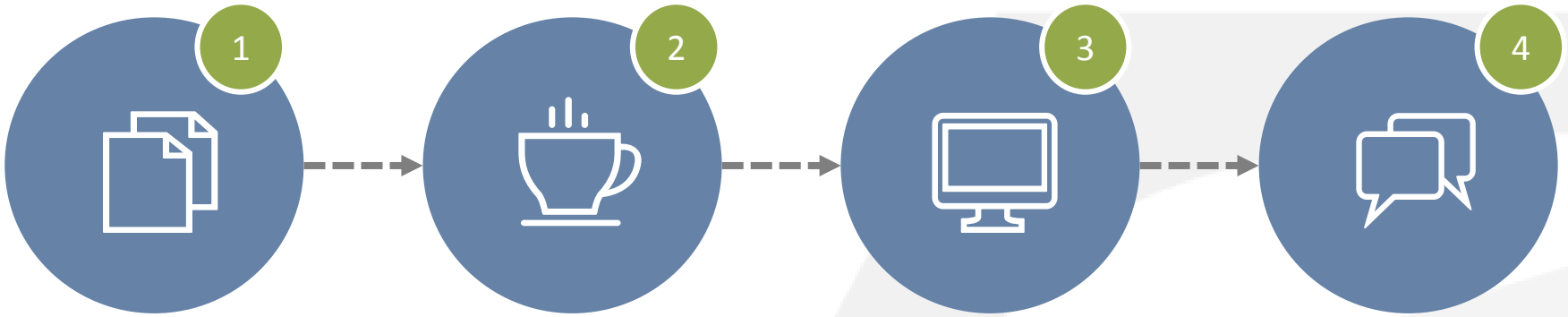


- 我們建議行動應用App安全檢測活動應由團隊測試人員或第三方單位人員進行檢測，透過靜、動態分析及黑、白箱等測試方式，針對行動應用App的安全性議題進行探討，有助於排除開發者自我開發之盲點。

- 獨立測試人員可由測試工程方法蒐集與探討相關惡意或可疑之函式庫、線上服務接口或相關第三方程式模組等，並且研擬相關測試方案、表單、報告與改善方法，有助於團隊之分工合作及提升行動應用App改版效率與品質。

行動應用App安全檢測流程

測試程序之4步驟



測試規劃

開發初期應規劃擬定相關測試計畫書，並載明測試目標與範圍及其相關方法。

測試準備

準備測試工具、環境及參考文獻，以順利執行測試計畫

App測試

執行App測試並記錄相關過程與回報問題。

評估控制

監控與檢討測試計畫，並評估其執行成效。

行動應用App安全檢測工具(1/3)

建議使用以下2個主要工具，可對Android系統進行黑白箱測試。

Santoku

MobSF

當開發人員與測試發布人員有溝通協作上的困難時，可考慮納入DevOps的開發概念，並可額外採用如HockeyApp這類的工具。

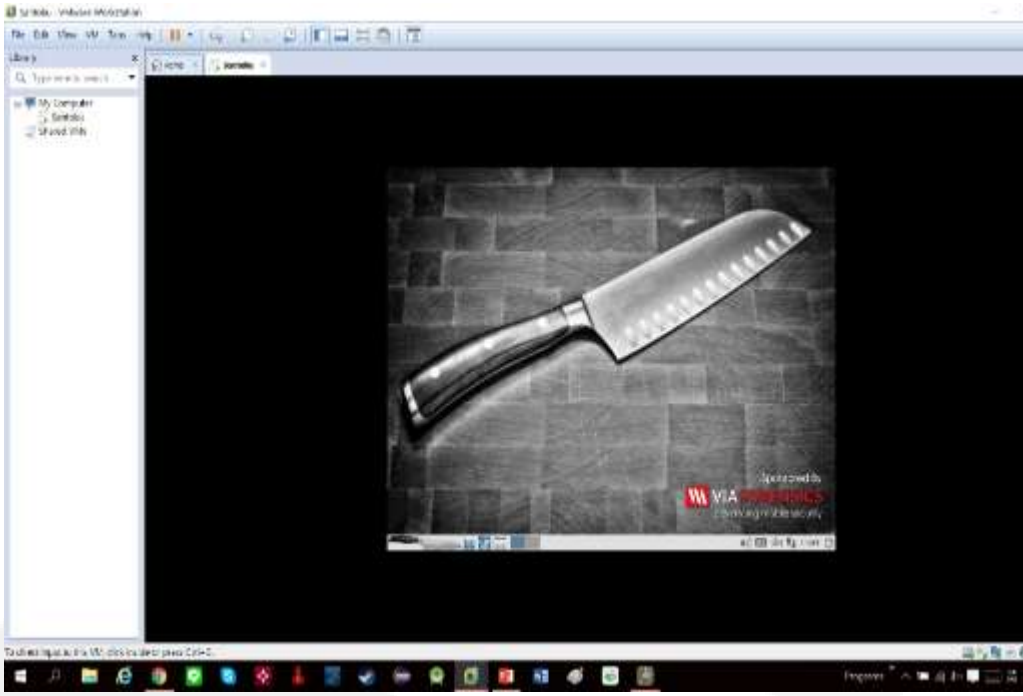
HockeyApp提供多樣性的行動開發工具，並具跨平台的行動應用程式測試功能，HockeyApp的功能包含錯誤報告(Crash Report)、App發布和回報等。

行動應用App安全檢測工具(2/3)

Santoku

- 適用於iOS及Android平台。
- 以Ubuntu Linux為基礎的整合工具環境作業系統，以提供使用者對手機或App進行檢測或分析。
- 主要功能
  - 行動裝置App鑑識
  - 行動裝置App惡意軟體分析
  - 行動應用安全檢測

主要採用其中BurpSuite進行行動應用App黑箱之網路檢測，探討其在網路資料傳輸或交換之安全項目。



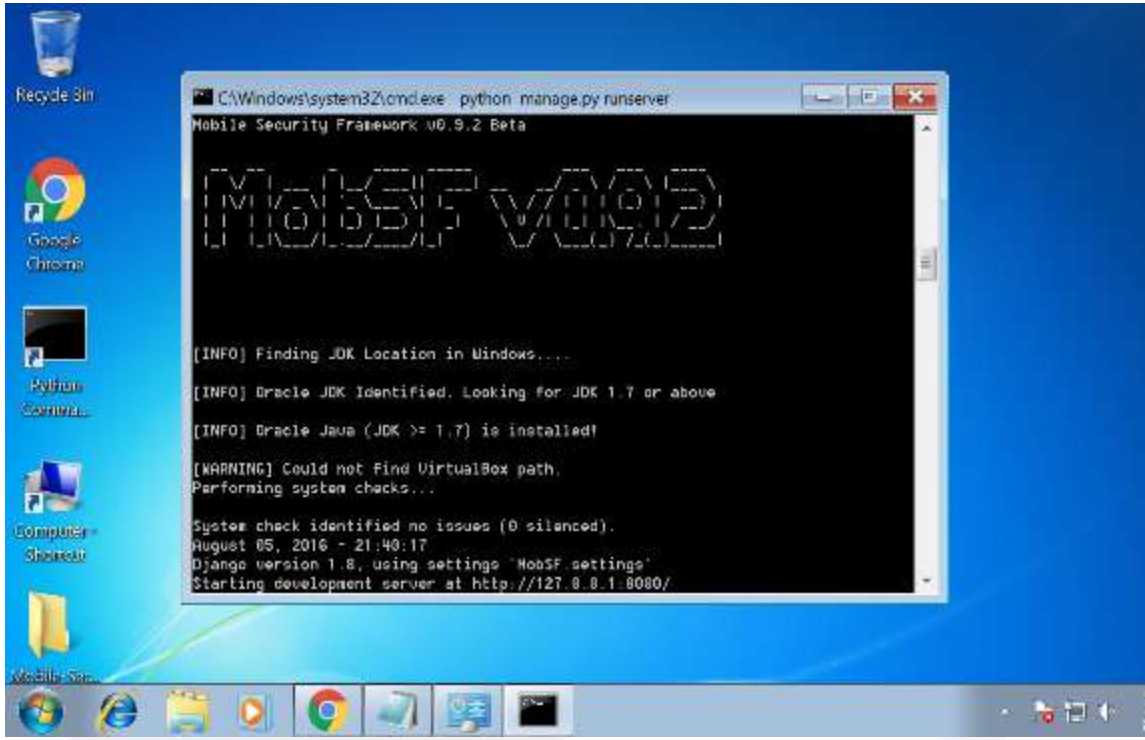
Santoku操作環境

行動應用App安全檢測工具(3/3)

MobSF

- 適用於iOS及Android平台。
- 為一個行動App分析安全框架，具備多種功能可針對行動應用App(Android版/ iOS版)進行分析，這個測試框架能夠執行靜態和動態(僅適用於Android平台)的分析。

建議開發者可採用由ajinabraham所發展的MobSF工具，有助於多數開發者可以較快上手進行使用。



MobSF執行畫面

行動應用App安全檢測實務

本單元以採用MobSF自動化工具檢測環境，並以某電子支付App為檢測範例。

APP基本資訊

- 檔案名稱：testsamlpe01.apk
- 主要功能：行動支付
- 取得權限：
  - 聯絡人
  - 電話
  - 相機裝置 ID 和通話資訊
  - 其它

本單元以透過取得該App的apk檔案，進行實務操作分析。

自我基本檢測目標與流程

提供開發人員於行動App開發完成後的初步基本安全檢測，藉以善盡開發者安全開發的基本責任。

- 檢測開始前針對待測之行動應用App進行基本版本、權限等公開資訊分析。
- 透過自動化檢測工具/環境，分別進行靜、動態分析及Web API安全性檢測。
- 根據其所產生之報告中，發掘問題點的特定項目檢測。針對問題警訊，依據檢測項目不同，其選擇適用之檢測工具及方法。

檢測環境及檢測方式

工具運行作業系統

- Mac OS X EI Capitan v.10.11.6

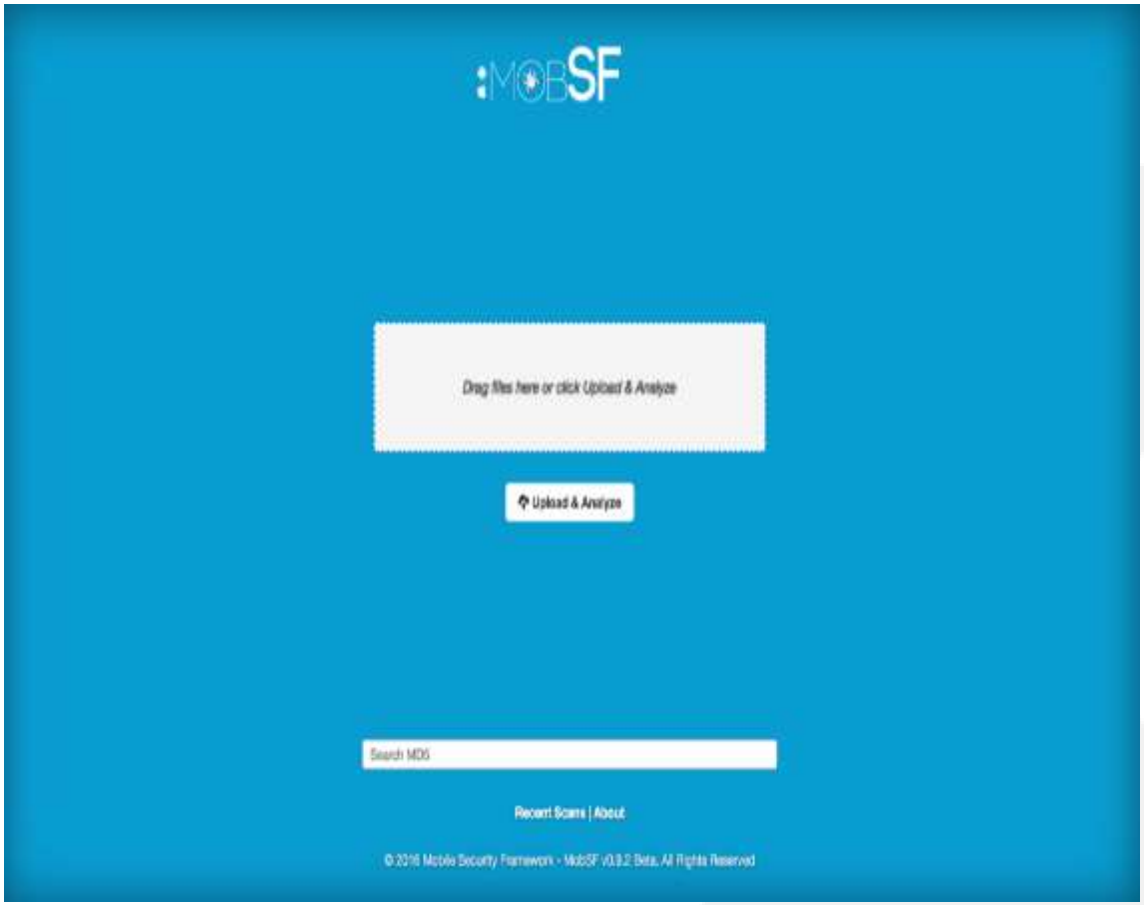
檢測工具/環境

- Mobile Security Framework(MobSF) v0.9.2 Beta(依工具要求環境建置)
- MobSF_VM_0.2.vbox (Samsung Galaxy S4, Android OS v.4.4.2)

檢測方法

- 黑箱測試
- 靜態分析及動態分析

檢測操作步驟-靜態分析(MobSF)



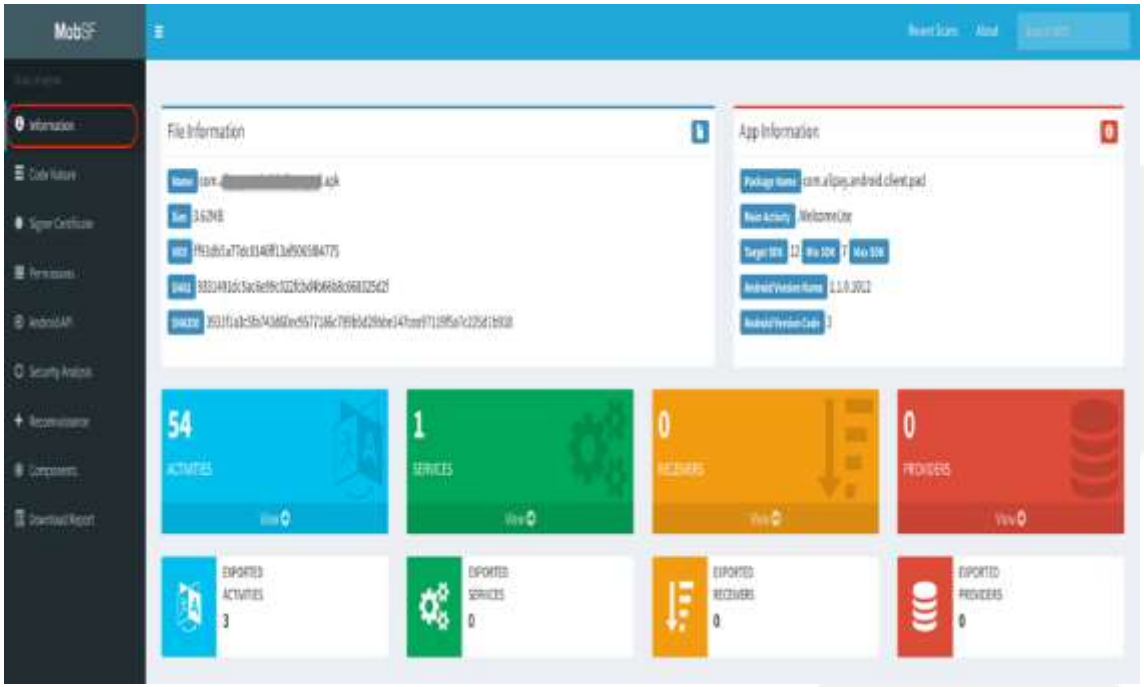
MobSF 開啟畫面

步驟一

上傳待測行動應用App

- 將欲檢測的行動App(App)上傳或拖曳至該測試平台。
- 支援檔案格式為apk、ipa檔。

檢測操作步驟-靜態分析(MobSF)



靜態分析- Information(Dashboard)

步驟二

靜態分析-產生測試結果

- App上傳後，系統即會自動開始進行靜態分析，其檢測項目有:
  - ✓ Information
  - ✓ Code Nature
  - ✓ Signer Certificate
  - ✓ Permissions
  - ✓ Android API
  - ✓ Security Analysis
  - ✓ Reconnaissance
  - ✓ Components
- 另可下載Java Code, Smali Code及AbdroidMainfest.xml檔檢視，以進行進階分析 。

檢測操作步驟-動態分析(MobSF)

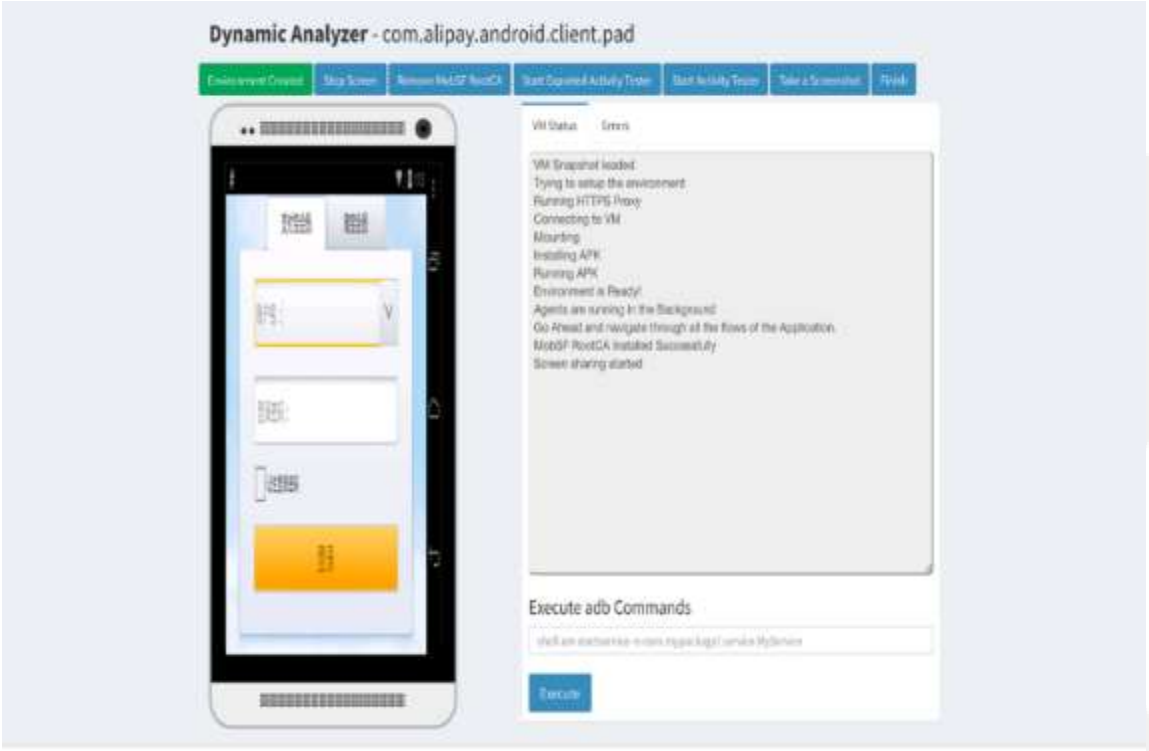
動態分析- 開啟畫面

步驟三

動態分析-載入測試平台

- 完成靜態分析後，點擊”Star Dynamic Analysis”進行動態分析(需事先完成Android模擬器，及VM IP, Host/Proxy IP, VM UUID及Snapshot UUID等環境設定)，進入動態分析頁面，並同時載入Android模擬器的快照。

檢測操作步驟-動態分析(MobSF)



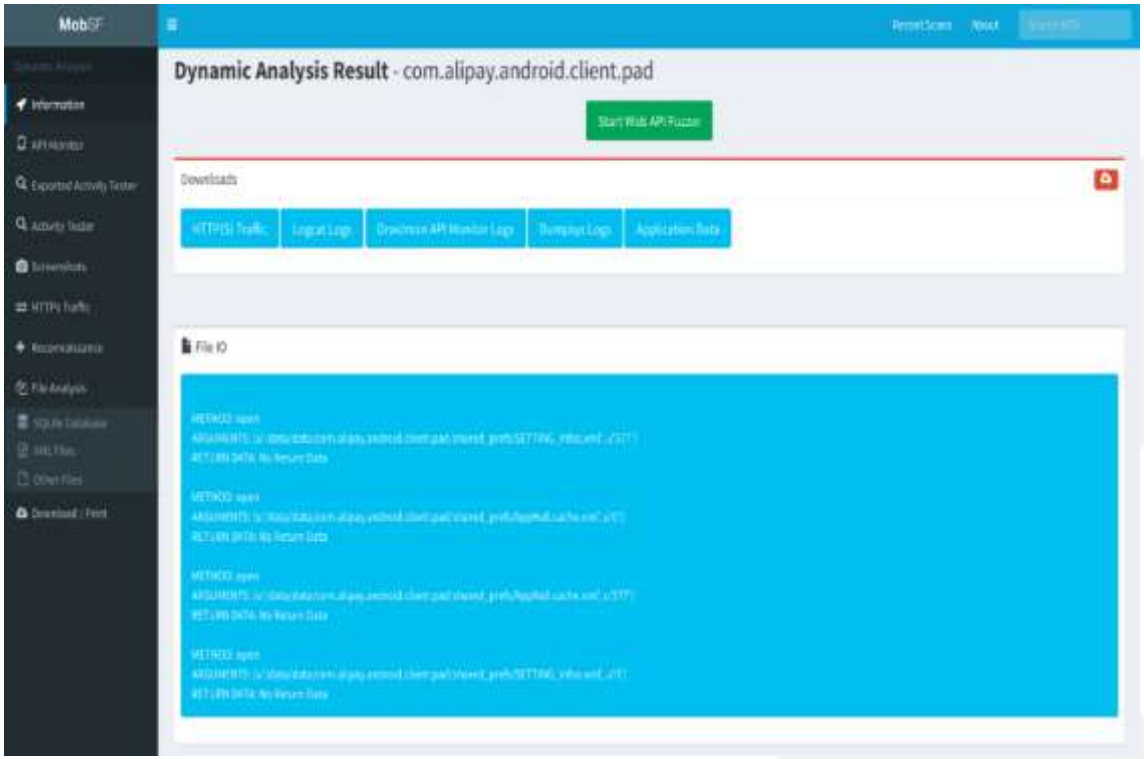
動態分析-測試環境

步驟四

動態分析-測試環境創建

- 完成載入後，點擊”Create Environment”建立動態測試環境。此時會載入並同時執行行動應用App。此時可依序進行動態測試項目 ：
  - ✓ Exported Activity Tester
  - ✓ Activity Tester
- 另動態分析時，同時可針對Android模擬器中行動應用App測試情形進行畫面擷取。

檢測操作步驟-動態分析(MobSF)



動態分析- Information

步驟五

動態分析-產生測試結果

- 檢測項目結果有 ：
  - ✓ Information
  - ✓ API Monitor
  - ✓ Exported Activity Tester
  - ✓ Activity Tester
  - ✓ Screenshots
  - ✓ HTTPs Traffic
  - ✓ Reconnaissance
  - ✓ File Analysis
- 並可下載HTTPs Traffic、Logcat Log、Droidmon API Monitor、Dumpsys Logs及Application Data等原始資料進行進階分析 。

檢測操作步驟-動態分析(MobSF)

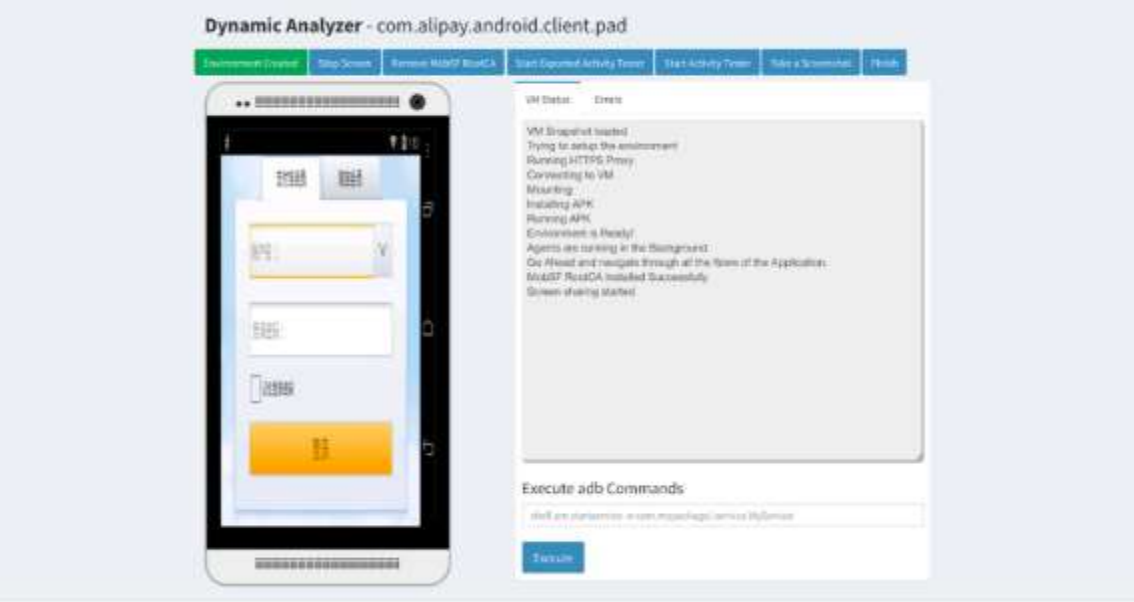
動態分析- 開啟畫面

步驟三

動態分析-載入測試平台

- 完成靜態分析後，點擊”Star Dynamic Analysis”進行動態分析(需事先完成Android模擬器，及VM IP, Host/Proxy IP, VM UUID及Snapshot UUID等環境設定)，進入動態分析頁面，並同時載入Android模擬器的快照。

檢測操作步驟-動態分析(MobSF)



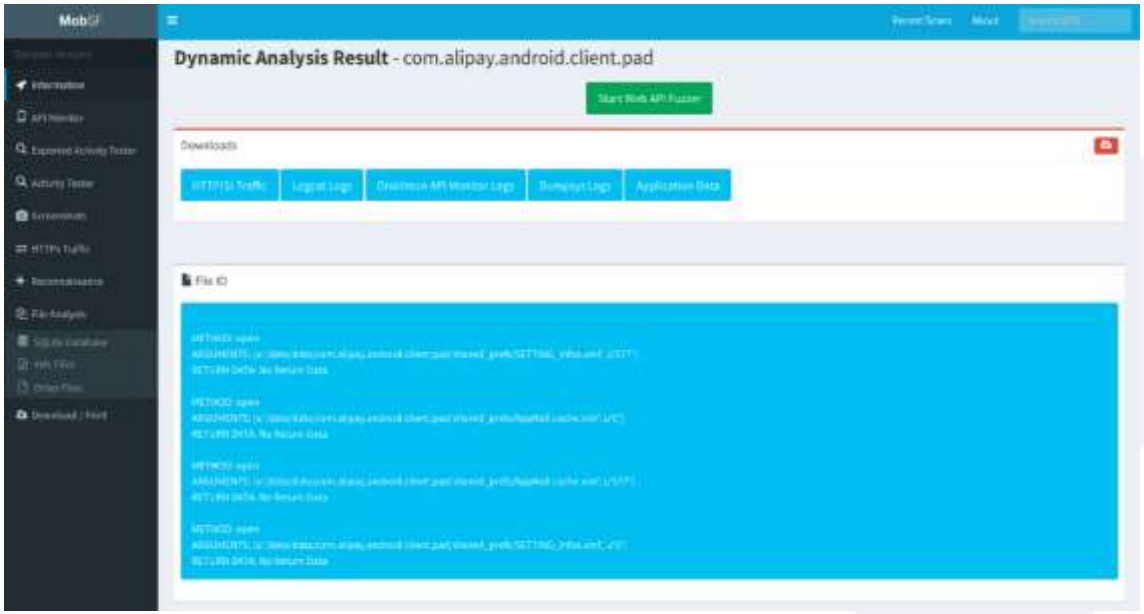
動態分析- 測試環境

步驟四

動態分析-測試環境創建

- 完成載入後，點擊”Create Environment”建立動態測試環境。此時會載入並同時執行行動應用App。此時可依序進行動態測試項目：
  - ✓ Exported Activity Tester
  - ✓ Activity Tester
- 另動態分析時，同時可針對Android模擬器中行動應用App測試情形進行畫面擷取。

檢測操作步驟-動態分析(MobSF)



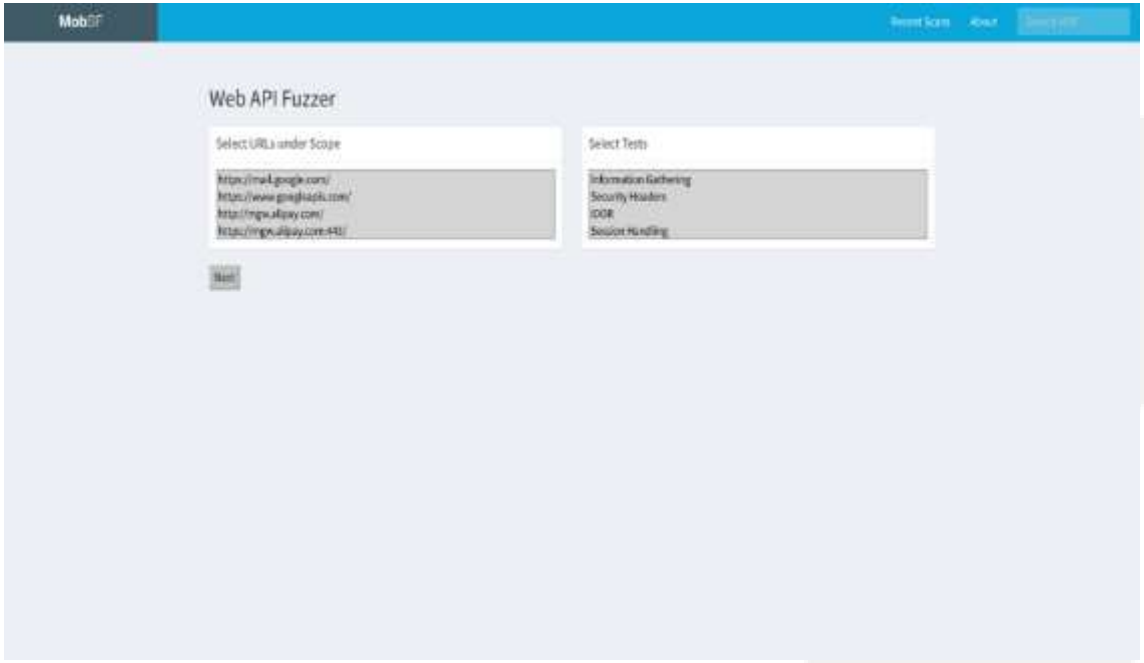
動態分析- Information

步驟五

動態分析-產生測試結果

- 檢測項目結果有:
  - ✓ Information
  - ✓ API Monitor
  - ✓ Exported Activity Tester
  - ✓ Activity Tester
  - ✓ Screenshots
  - ✓ HTTPs Traffic
  - ✓ Reconnaissance
  - ✓ File Analysis
- 並可下載HTTPs Traffic、Logcat Log、Droidmon API Monitor、Dumpsys Logs及Application Data等原始資料進行進階分析。。

檢測操作步驟-Web API 分析(MobSF)



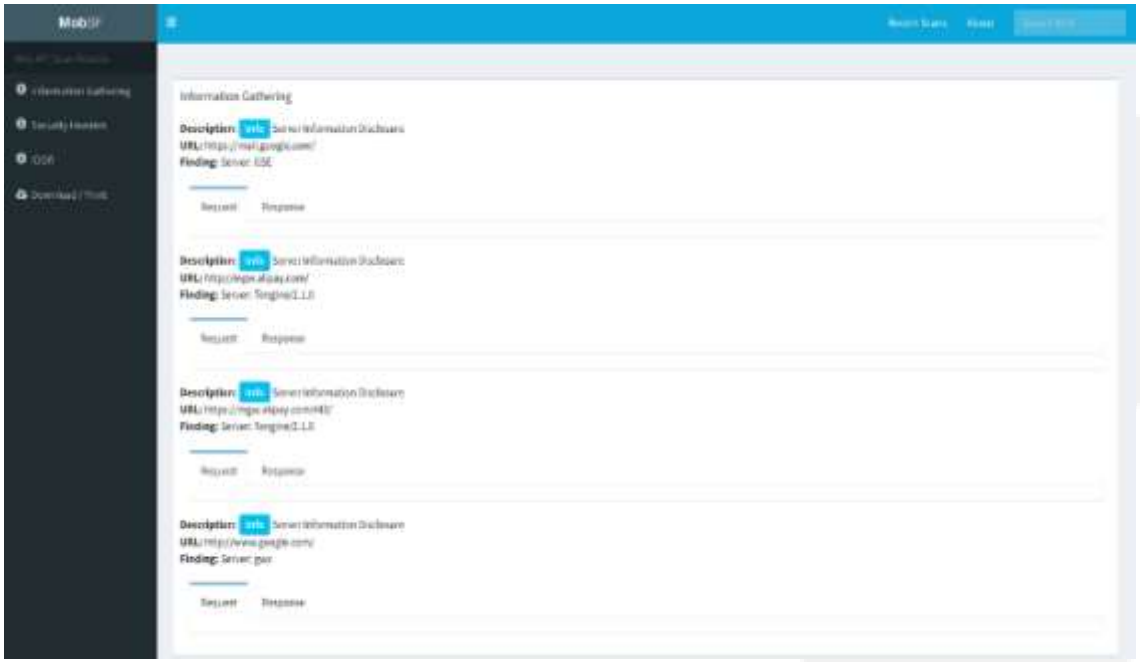
Web API分析- 開啟畫面

步驟六

Web API分析-選取測試項目

- 點擊動態分析完成畫面之”Star Web API Fuzzer”執行行動應用App之Web API應用架構的進階分析，測試之項目有:
  - ✓ Information Gatering
  - ✓ Security Headers
  - ✓ IDOR
  - ✓ Session Handling
  - ✓ SSRF
  - ✓ XXE
  - ✓ Path Travrsal
  - ✓ Rate Limit Check

檢測操作步驟-Web API 分析(MobSF)



Web API分析- 測試結果

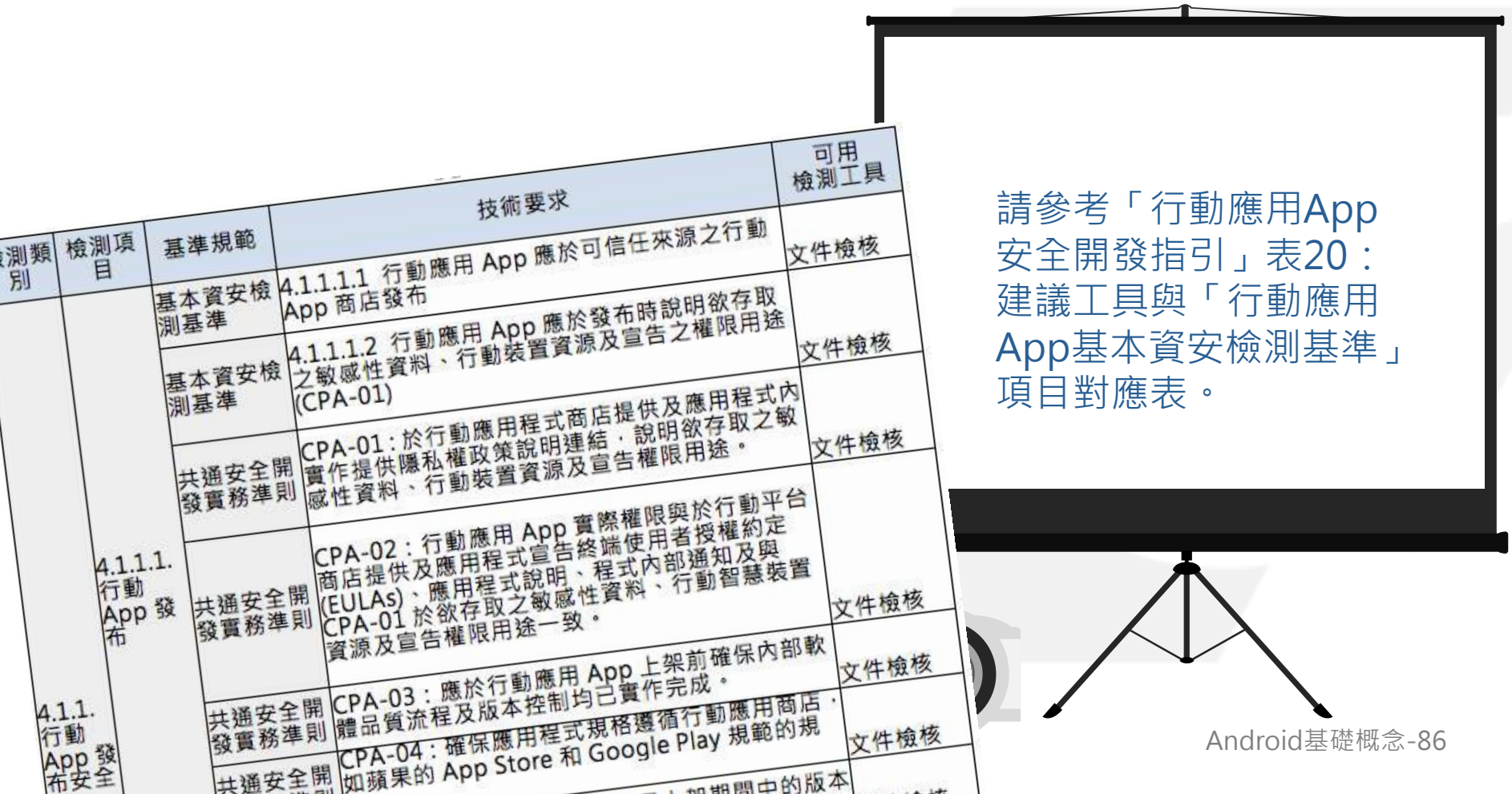
步驟七

Web API分析-產生測試結果

- 根據測試產生的結果報告，進行檢測項目的解讀與判斷。

「行動應用App基本資安檢測基準」各構面與開發最佳實務工具對應

以工業局公告的「行動應用App基本資安檢測基準」與相關安全規範等之相關安全項目，對應本單元提出可應用之檢測工具，彙整成表，提供開發者參考。

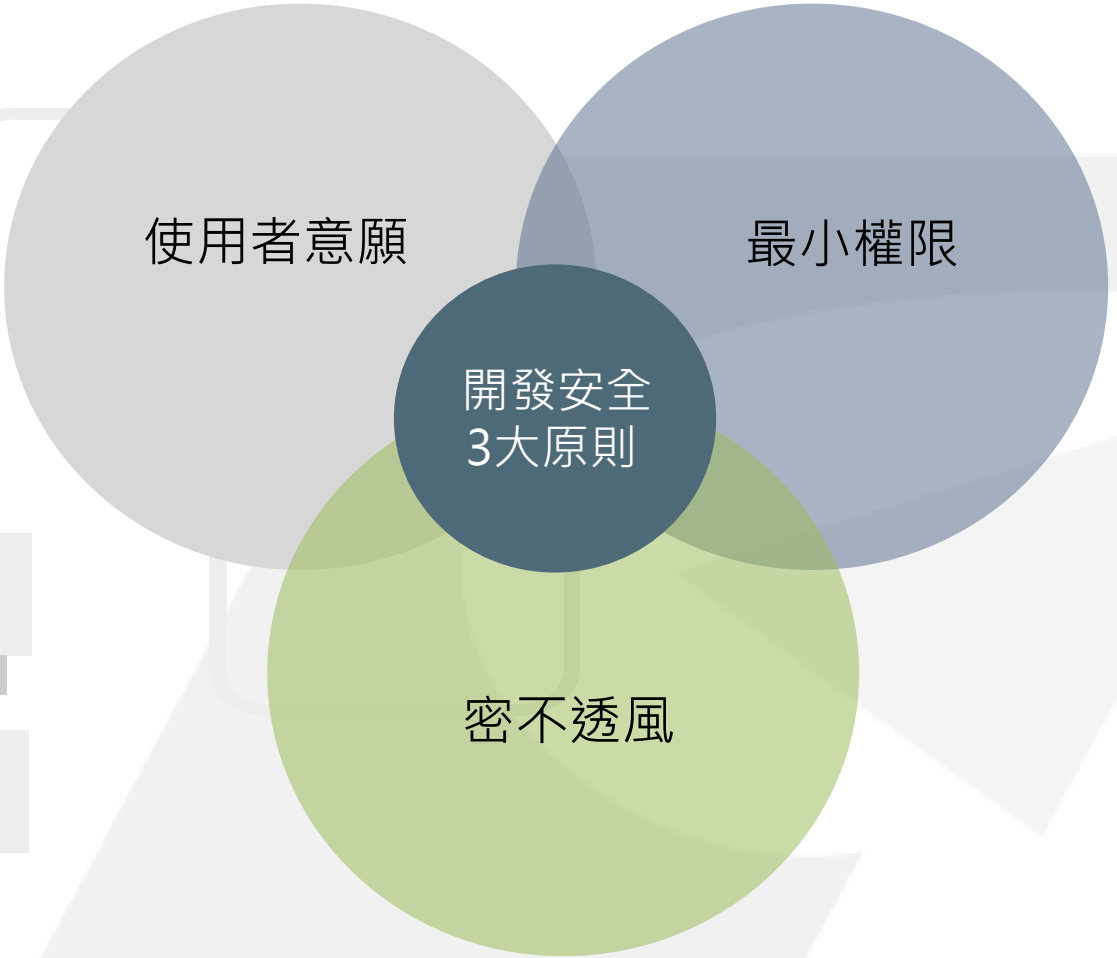


檢測類別	檢測項目	基準規範	技術要求	可用檢測工具
	4.1.1.1. 行動App 發布	基本資安檢測基準	4.1.1.1.1 行動應用 App 應於可信任來源之行動 App 商店發布	文件檢核
		基本資安檢測基準	4.1.1.1.2 行動應用 App 應於發布時說明欲存取之敏感性資料、行動裝置資源及宣告之權限用途(CPA-01)	文件檢核
		共通安全開發實務準則	CPA-01：於行動應用程式商店提供及應用程式內實作提供隱私權政策說明連結，說明欲存取之敏感性資料、行動裝置資源及宣告權限用途。	文件檢核
		共通安全開發實務準則	CPA-02：行動應用 App 實際權限與於行動平台商店提供及應用程式宣告終端使用者授權約定(EULAs)、應用程式說明、程式內部通知及與CPA-01 於欲存取之敏感性資料、行動智慧裝置資源及宣告權限用途一致。	文件檢核
4.1.1. 行動App 發布安全		共通安全開發實務準則	CPA-03：應於行動應用 App 上架前確保內部軟體品質流程及版本控制均已實作完成。	文件檢核
		共通安全開發實務準則	CPA-04：確保應用程式規格遵循行動應用商店，如蘋果的 App Store 和 Google Play 規範的規	文件檢核
		共通安全開		

請參考「行動應用App安全開發指引」表20：建議工具與「行動應用App基本資安檢測基準」項目對應表。

行動應用App開發3大原則

需要有一個安全開發流程 - 安全軟體開發生命週期(SSDLC)，協助開發者藉由安全需求、安全設計、實作安全、測試安全及上線安全，開發出不會濫取使用者資料、能對抗惡意攻擊的行動應用App。



問題與討論