行動應用App安全開發說明會
Android設計實務

指導單位：經濟部工業局

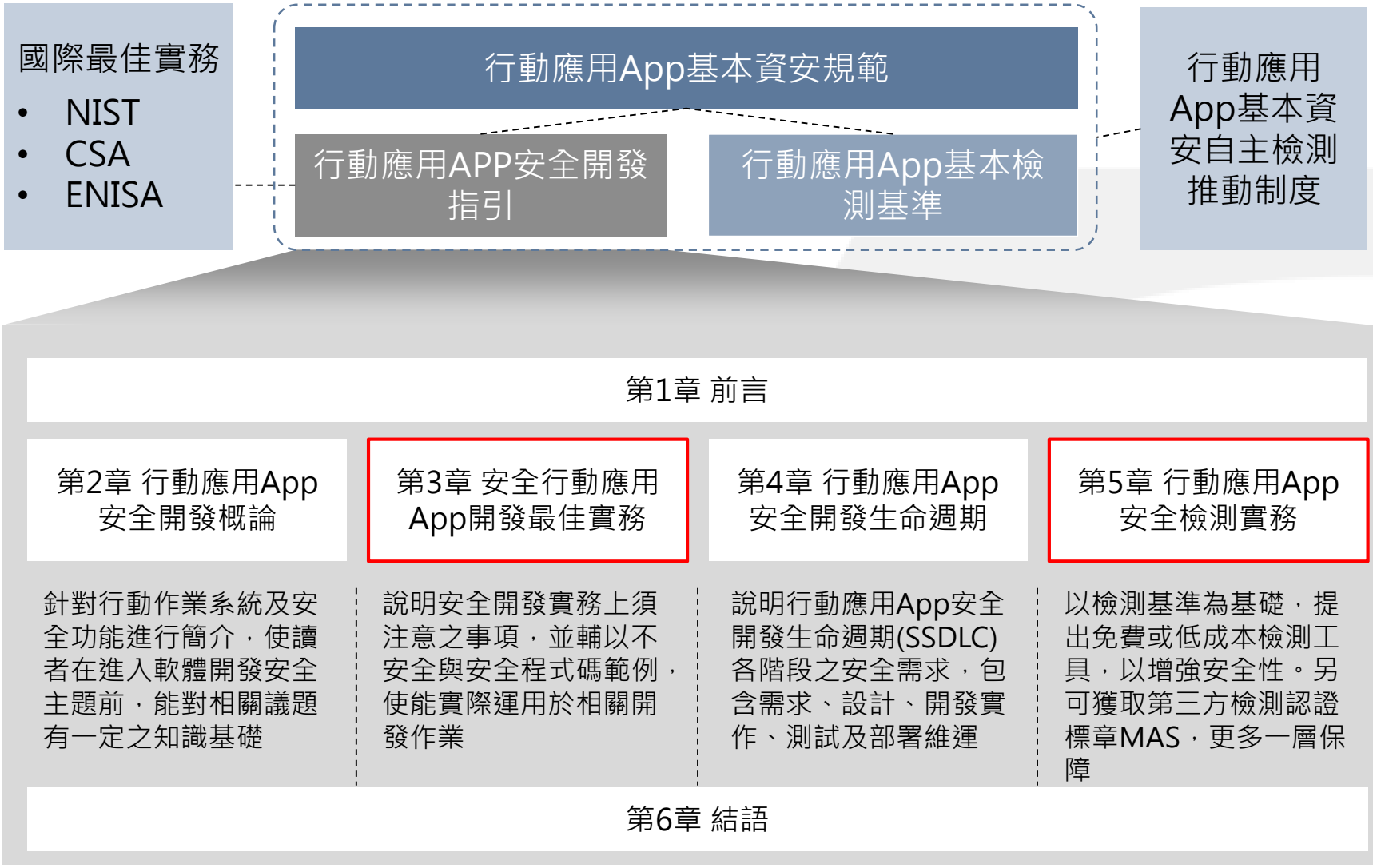
執行單位：財團法人資訊工業策進會、中華民國資訊軟體協會

協辦單位：台北市電腦公會、中華民國資訊安全學會

課程內容簡介

- Android安全行動應用設計最佳實務
  - 本單元將介紹在Android上開發行動應用App的安全開發注意事項，除使學員瞭解相關注意事項細節、解決方法，並輔以程式碼範例，使學員未來能實際運用於行動應用App開發作業中。
- 行動應用APP安全檢測流程與工具
  - 參考經濟部工業局所發展的「行動應用App基本資安規範」、「行動應用基本資安檢測基準」與「行動應用App基本資安自主檢測推動制度」等相關規範及OWASP/Cigital Touchpoint方法論。
  - 介紹行動應用App源碼檢測工具，協助開發人員得以在建構行動應用App時，從源頭貫徹資訊安全觀念作法，並能自行以低成本進行安全檢測。

行動應用APP安全開發指引架構簡介



課程大綱

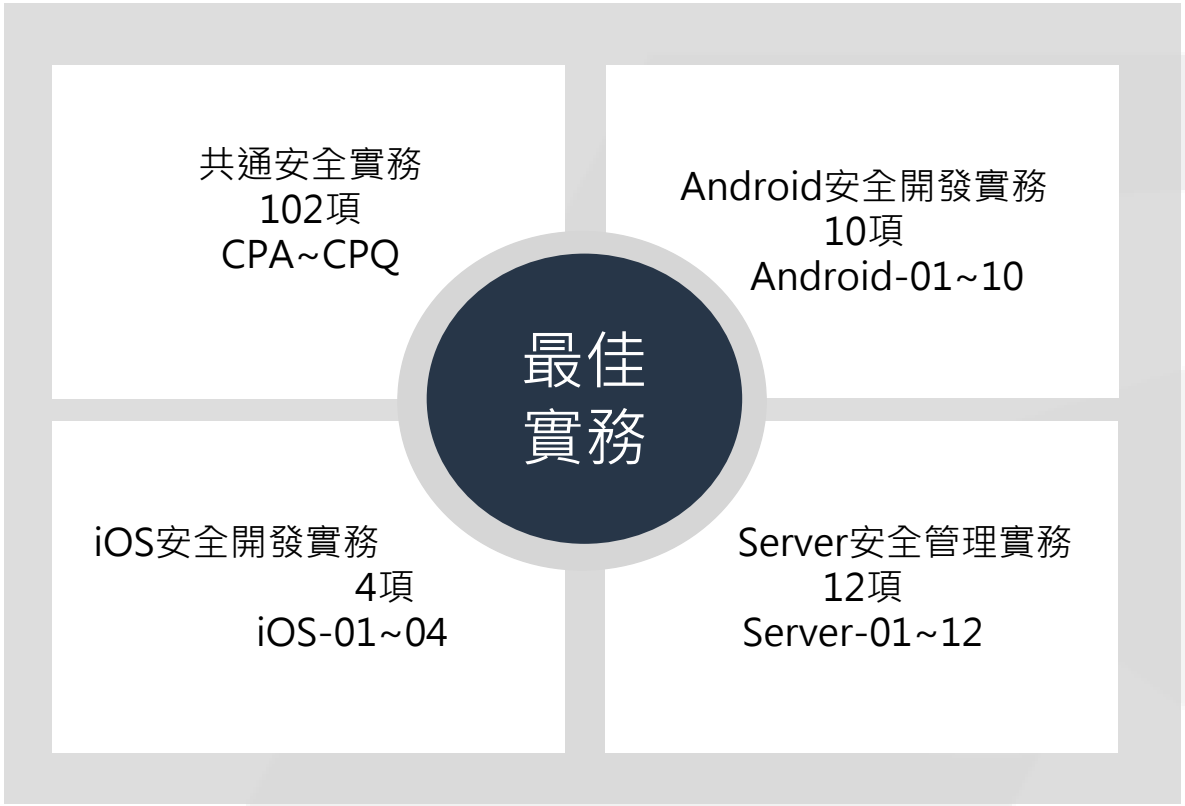
第1單元	Android安全行動應用設計最佳實務	2.5小時
第2單元	行動應用App安全檢測流程與工具	0.5小時

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。

課程大綱

第1單元	Android安全行動應用設計最佳實務	2.5小時
	共通實務(Common Practices)	
	Android實務	
	Server實務	
第2單元	行動應用App安全檢測流程與工具	0.5小時

敏感性資料蒐集(D) -CPD-01

CPD-01 需求階段需要識別計畫開發行動應用App蒐集、處理及利用敏應資料類別(例如密碼，個人資料、地理位置、財務資訊及錯誤日誌等)及行動應用App平台與發行國的隱私法令要求，定義安全性需求。

簡介

- 通常免費或無償提供行動應用App，多數負有蒐集、利用及分享行動智慧裝置擁有者個人資料有關識別與行為資料任務 。
- 個人資料的蒐集、處理及利用應遵循發行國之個資保護相關法令規定。

> 明確告知蒐集當事人的個人資料由那一個公務或非公務機關以何種目的蒐集個資當事人的何種類別資料，且會利用之期間、地區、對象及方式，及當事人對已被蒐集的個人資料有何權利，讓使用者在下載之前或是蒐集使用者本人個人資料前有判斷的依據，讓使用者可以選擇同意或是拒絕，如果選擇拒絕是否會影響公務或非公務機關對當事人服務的品質。

- 行動應用App專案經理與系統分析人員應充分識別需求單位處理個人資料在內的敏感資料類別，並定義出安全保護需求，由系統分析 / 設計人員進一步設計安控及隱私保護措施。

開發生命週期	需求階段		
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	4.1.2.1	行動應用App基本資安檢測基準	N/A

敏感性資料利用(E)-CPE-03

CPE-03 不得實作未經使用者同意，使行動應用App可擅自修改使用者資料的行為，包括在使用者無確認情況下刪除或修改使用者連絡人資料、通話記錄、簡訊資料和多媒體簡訊資料的行為。

簡介

- 使用者同意行動應用App開發人員蒐集敏感性資料不代表使用者已經授權可以任意處理這些敏感性資料，破壞資料完整性與正確性，如確有需要修改使用者資料，包括刪除或修改使用者連絡人資料、通話記錄、簡訊資料和多媒體簡訊資料的行為，請參考CPD-05及CPD-06實務取得使用者同意的意向再啟動。

CPD-05:蒐集敏感性資料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。

CPD-06:應依CPD-03實作行動應用App存取敏感性資料權限，不要授與過度蒐集不必要敏感性資料權限或於未告知使用者取得同意前於行動應用App背景運作蒐集敏感性資料活動。

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	4.1.2.2	行動應用App基本資安檢測基準	N/A

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。

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。

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

敏感性資料儲存(F)-CPF-08

CPF-08 需注意記憶體暫存的敏感性資料，如固定金鑰與密碼的安全性，當不需要時或於一定合理期間應強制清除或使用於一定期間就失效的可變量金鑰替代。

簡介

- 在行動應用App使用時，使用者或應用程式會將特定資料儲存在記憶體中，以方便使用者離開或逾時之後，下次能夠直接使用。
- 在Android上因為應用程式使用後停留在記憶體中，會被駭客利用除錯器竊取。
- 另一方面，針對敏感的密鑰、密碼，建議不要用字串儲存，改用Byte Array的方式儲存，以免容易被發現。而使用過之後，也建議應該儘快清除，或確定會被系統記憶體回收機制回收。

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	4.1.2.3	行動應用App基本資安檢測基準	N/A

敏感性資料儲存(F)-CPF-17

CPF-17 行動應用App有提供標準的設定參數檔函數，但由於駭客取得安裝檔之後，非常容易就可以修改這些標準的參數檔，於是為了加強安全層級，應用程式的設定參數，建議放在安全的地方，例如編譯至程式碼中，或者進行加密。

簡介

- 在iOS的行動應用App中，其應用程式的設定常儲存在某些情況下會受到影響的plist文件。同樣地，Android的開發商往往將App的設定儲存在共享的喜好XML文件或SQLite資料庫設定，預設不會加密，並可以讀取或甚至可用root權限進行修改。
- 在可行的情況下儘可能編譯設定到程式碼。有一點好處是於通過配置在iOS plist文件的應用程式，因為變更必須捆綁，且無論如何均需部署為新的應用程式。相反地攻擊者需要包括更多的時間和技能，以修改應用程式程式碼中的配置。不要存放在檔案目錄或其他文件中包含任何密鑰的設定，除非先進行加密。理想的情況下，使用由使用者提供的密碼的主密鑰，去加密所有的設定文件，或由遠端提供的一個密鑰。

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	N/A

敏感性資料傳輸(G)-CPG-01

CPG-01 行動應用App傳輸敏感性資料應規劃並實作傳輸全程全時使用TLS 1.1以上加密，維護敏感性資料機密性及完整性。

簡介

- 新的Web應用協定HTTP/2和SPDY並不支援SSL，只支援TLS。
- Google所有公共服務的加密都是使用TLS，故如果要整合Google的一些功能，只能使用TLS。
- PCI DSS規範SSL在2016年6月30日之後不得繼續使用。

Apple將在2017年1月強制所有iOS App開始使用ATS(App Transport Security, 支援TLS v1.2)，參考iOS-04。

Android支援標準TLS實作，在官方開發人員網站有完整的概念及實作介紹，並包含實例及常見問題。

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

敏感性資料刪除(I)-CPI-01

CPI-01 行動應用App儲存敏感性資料，應預先針對各國個人資料保護法或隱私要求及實務需求，定義資料於行動智慧裝置合理最長保存期限，預設App自使用者行動智慧裝置刪除或反安裝同時，針已儲存持久性敏感資料屆期可詢問是否需要刪除或是進行永久性加密。

簡介

- 資料刪除或銷毀是使用者敏感性資料在行動應用App生命週期的終點站。

- 非持久性資料(如地理位置、日誌檔等)應視需要設定留存時間。
- 持久性敏感性資料(如個人基本資料、聯絡人資料、付費資訊等)需考量各國個人資料保護法或隱私要求及實務需求，定義資料於行動智慧裝置合理最長保存期限，預設為App自使用者行動智慧裝置刪除或反安裝同時，並將敏感性資料保留期限，納入行動應用程式商店及行動應用App內隱私權政策聲明中，讓使用者了解。

開發生命週期	需求階段		
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	4.1.2.6	行動應用App基本資安檢測基準	N/A

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-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)

行動應用程式完整性(O)-CPO-03

CPO-03 行動應用App程式碼應運用人工或工具使之增加複雜度，並輔以限制除錯器使用、反追蹤、二進位剝離等措施，使惡意人士使用逆向工程方法分析程式碼難度增加。

簡介

- 攻擊者可藉由使用軟體逆向工程工具，逆向解開應用程式程式碼，並窺視程式的邏輯、破解程式保護，甚或盜取程式碼。另一方面，當手機端App被破解時，也會使得Server端的服務被窺伺或破解。
- 為了防止此問題，應適當加入保護措施，以保護程式碼內容與Server主機安全。
- 保護措施的建議方向有：

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	4.1.5.2	行動應用App基本資安檢測基準	N/A

使用者輸入驗證(Q)-CPQ-03

CPQ-03 行動應用App提供使用者輸入值儘量可以參數化(Query parameterization)。

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。請參照「行動應用App安全開發指引」CPQ-03程式碼範例所示，如何建構最常見的程式語言的參數化查詢。
  這些程式碼的目的是展示給Web開發人員如何避免在Web應用程式中建立資料庫查詢的時候遭受SQL注入攻擊。

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	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	N/A

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)

課程大綱

第1單元	Android安全行動應用設計最佳實務	2.5小時
	共通實務(Common Practices)	
	Android實務	
	Server實務	
第2單元	行動應用App安全檢測流程與工具	0.5小時

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-03

Android-03 Android版行動應用App實作廣播(Broadcast intent)應設定權限，避免被惡意行動應用App偽造元件。

簡介

- 當發送廣播意圖(Intent)，如果沒有設定權限，那麼任何非特權應用程式皆可以接收意圖(Intent)，除非它有一個明訂的目標. 若有一個惡意的開發者創建具有完全相同的元件名的應用程式假裝為一個合法的元件，只要該命名空間尚未使用，此惡意程序將可以安裝在裝置上。

注意：說定權限來保護應用程式的意圖(Intent)，並注意應通過廣播發送敏感資訊給第三方組件，因這個組件有可能會在惡意安裝中被替換而需注意。

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	4.1.5.1	行動應用App基本資安檢測基準	N/A

Android-07

Android-07 Android版行動應用App應謹慎實作Content Providers，避免因權限過高或未驗證資料目的地與來源，遭惡意程式注入式攻擊。

簡介

- 內容提供者 (Content Providers)是一個標準的方式讓應用程式可以使用URI地址定義來分享資料和關連式資料庫的模型。也可以用來當做使用URI地址定義來存取檔案的方式。

  注意：內容提供者 (Content Providers)可以宣告權限(Permissions)，並且可以限制讀和寫的權限。除非有其必要性否則請勿允許寫入權限。同理除非必要，請設定讀取限制，以免未授權的應用程式可以讀取內容提供者，以提高安全性。

- 內容提供者也允許資料紀錄等級(Record Level)的分享而不用整個資料庫分享，使用此特性來提供操作上需要的最低的存取限制。
- 傳遞給內容提供者的參數需視為不受信任的輸入，不可以在未經正規化過濾，就直接組合成SQL查詢。SQL程式碼可以在內容提供者的請求中被送出，並且如果此SQL程式碼是一個查詢，則內容提供者會返回結果資料或控制給攻擊者。內容提供者的服務是基於檔案層級時，要確保文件名稱的路徑遍歷有過濾掉。例如，如果攻擊者使用“../../../file”作為檔案名時，它可能導致程序讀取非預設的文件目錄，並且此資料將會被攻擊者得到。另外注意，如果攻擊者建立符號鏈接(Symbolic Link)也可能產生類似的結果。

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)

課程大綱

第1單元	Android安全行動應用設計最佳實務	2.5小時
	共通實務(Common Practices)	
	Android實務	
	Server實務	
第2單元	行動應用App安全檢測流程與工具	0.5小時

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

Server-06

Server-06 網頁伺服器需預防網頁掛馬及點擊劫持攻擊。

簡介

- 什麼是網頁掛馬？點擊劫持？

網頁掛馬(Framing)	點擊劫持(clickjacking)
係指使用iFrame來遞送一個Web/WAP網站連線。這種攻擊可以用 “包裝(wrapper)” 網站來執行點擊劫持攻擊。	係指利用熱門的網站服務(例如Facebook)，通過諸如跨網站指令碼(Cross-Site Scripting，常簡稱為XSS) ，於頁面上嵌入iFrame或程式碼，以獲取受影響主機的控制權。此一網頁頁框的主要目的是誘騙使用者點擊嵌在iFrame的隱藏連結，將使用者重新導向到攻擊者控制的網站以竊取資訊的一種威脅。

- 防止Framing的方法
  - 不使用網頁視圖(Web View)及停用JavaScript。
- 專門設計的WebView的API可以被濫用來破解的WebView中指定的Web內容的安全性。可以實作下列機制以避免此種攻擊：
  - 阻止上傳頁框託管於其他域名的請求內容的X-Frame-Option HTTP response header。然而，與已被入侵主機連線時，這種緩解方法並不適用。
  - 利用內部防禦機制，以確保所有UI元素在頂層框架上傳;這樣就避免了在較低的層級水準，通過不信任的頁框設置服務內容。

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	N/A	行動應用App基本資安檢測基準	N/A

Server-10

Server-10 參考CPG-01~CPG-03實作TLS伺服器端設定。

簡介

- 目前符合公認安全的版本為TLS 1.2以上。

SSL/TLS更新歷程

定義	SSL 1.0	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
年份	NA	1995	1996	1999	2006	2008	待定

停用1024位元金鑰

美國NIST建議於2013之前，金鑰長度由1024位元，轉換成2048位元。因1024位元金鑰有可能會被破解，而2048位元金鑰於理論上並不會在有限的時間(數年內)被破解。

停用SSL 3.0

自從2014 SSL 3.0的CVE- 2014-3566漏洞會造成資料不安全，各大網站推廣停用 SSL3.0，改用TLS加密協定。

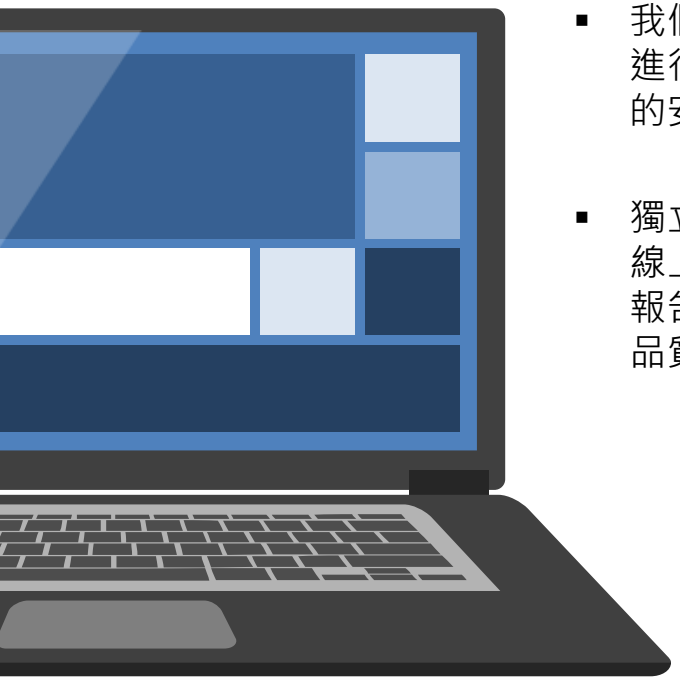
- OpenSSL heartbleed bug
  - OpenSSL大量被應用於網頁主機上，於2014/04/08發布1.0.1g修正此問題。此問題會影響包括SSL與TLS，會導致主機憑證私鑰被竊、主機資料被竊及客戶端資料被竊等議題。

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	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	Android安全行動應用設計最佳實務	2.5小時
第2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

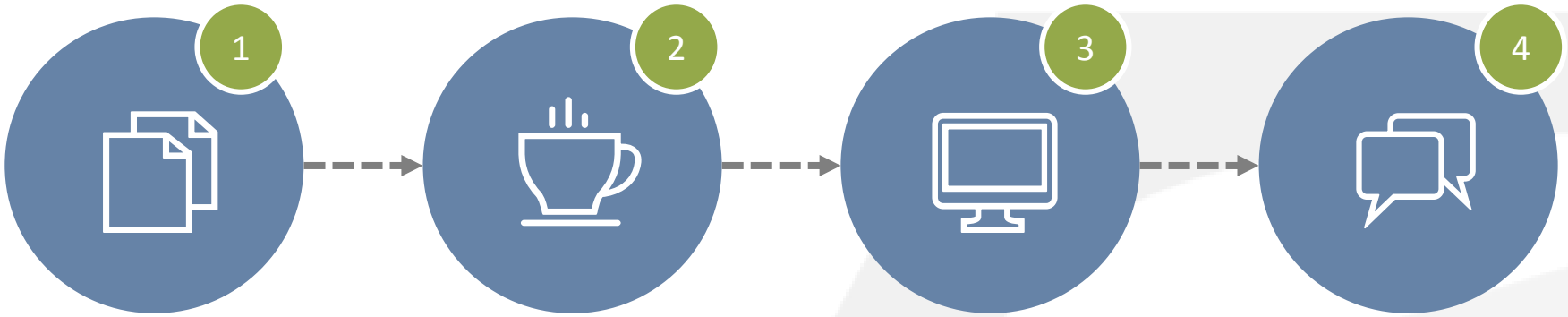


- 我們建議行動應用App安全檢測活動應由團隊測試人員或第三方單位人員進行檢測，透過靜、動態分析及黑、白箱等測試方式，針對行動應用App的安全性議題進行探討，有助於排除開發者自我開發之盲點。

- 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。

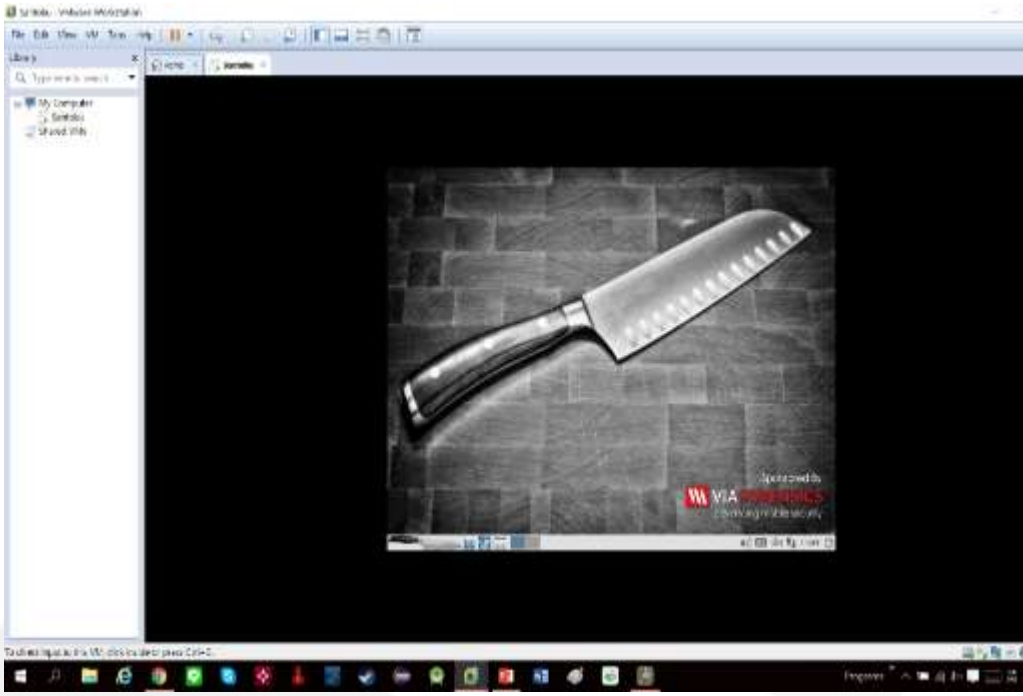
HockeyApp提供多樣性的行動開發工具，並具跨平台的行動應用程式測試功能，HockeyApp的功能包含錯誤報告(Crash Report)、App發布和回報等。

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。



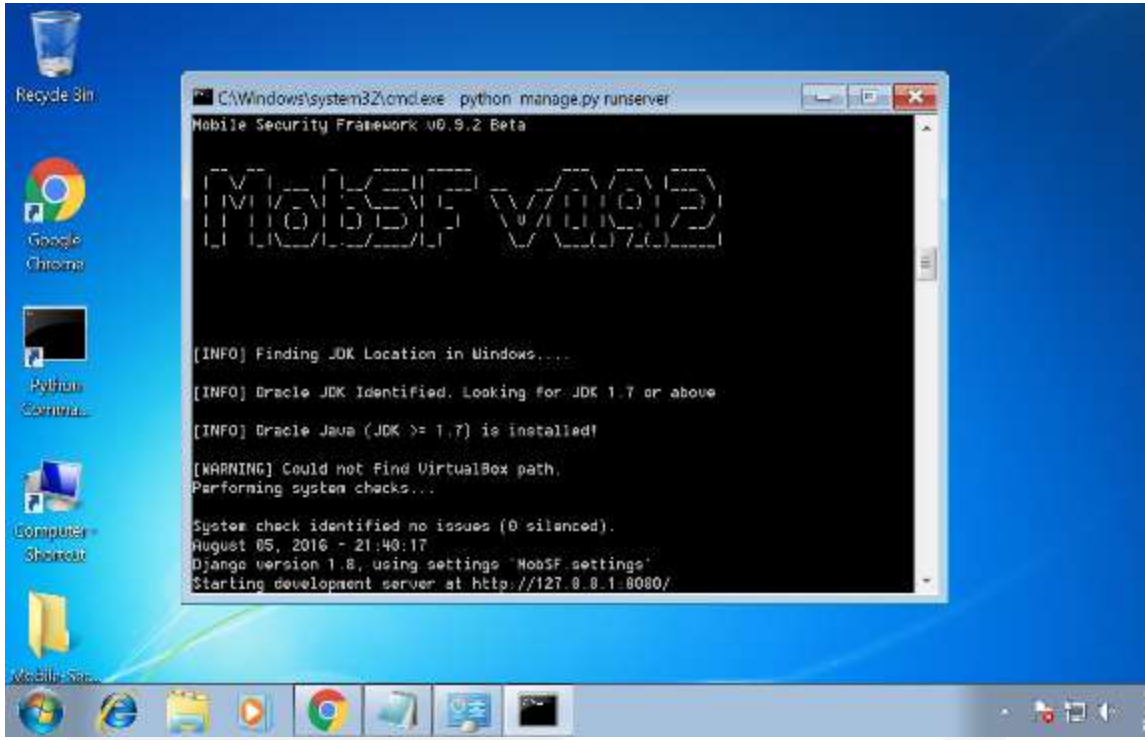
Santoku操作環境

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

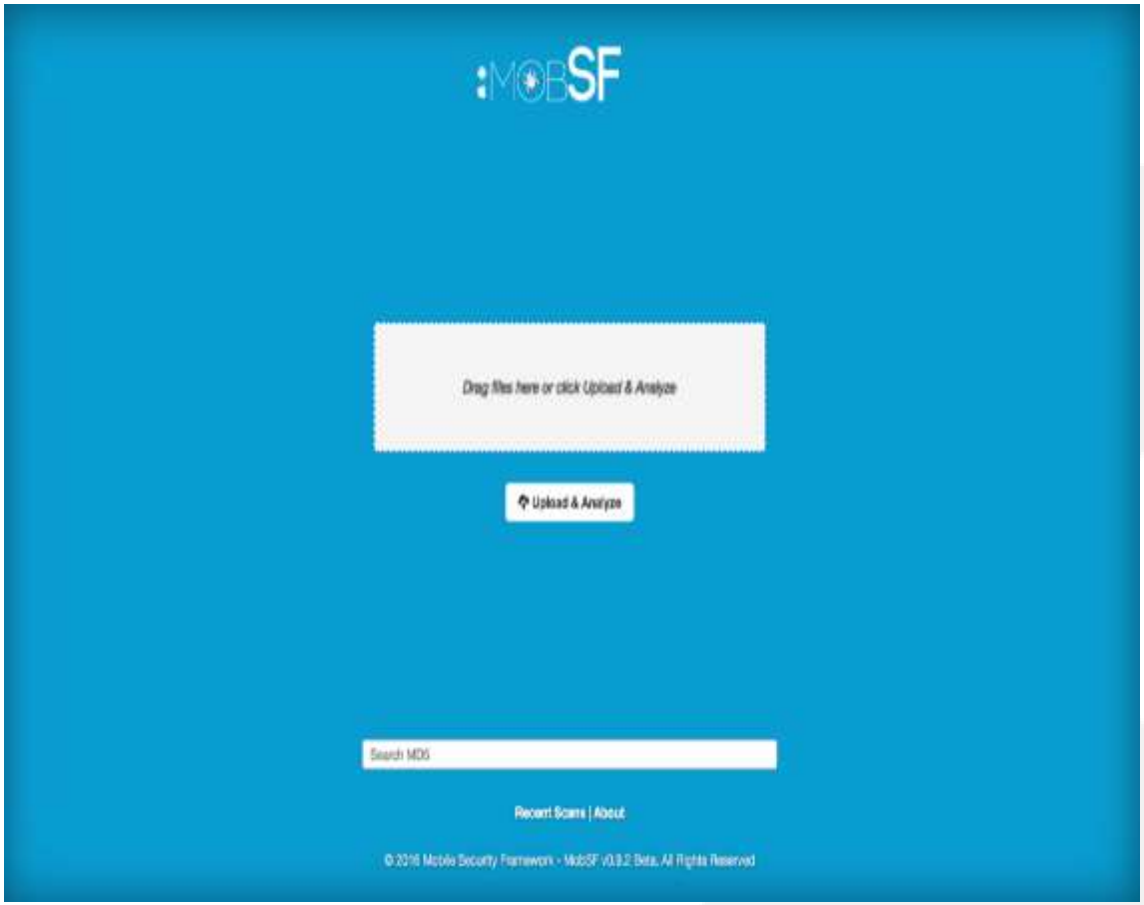
檢測工具/環境

- 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)



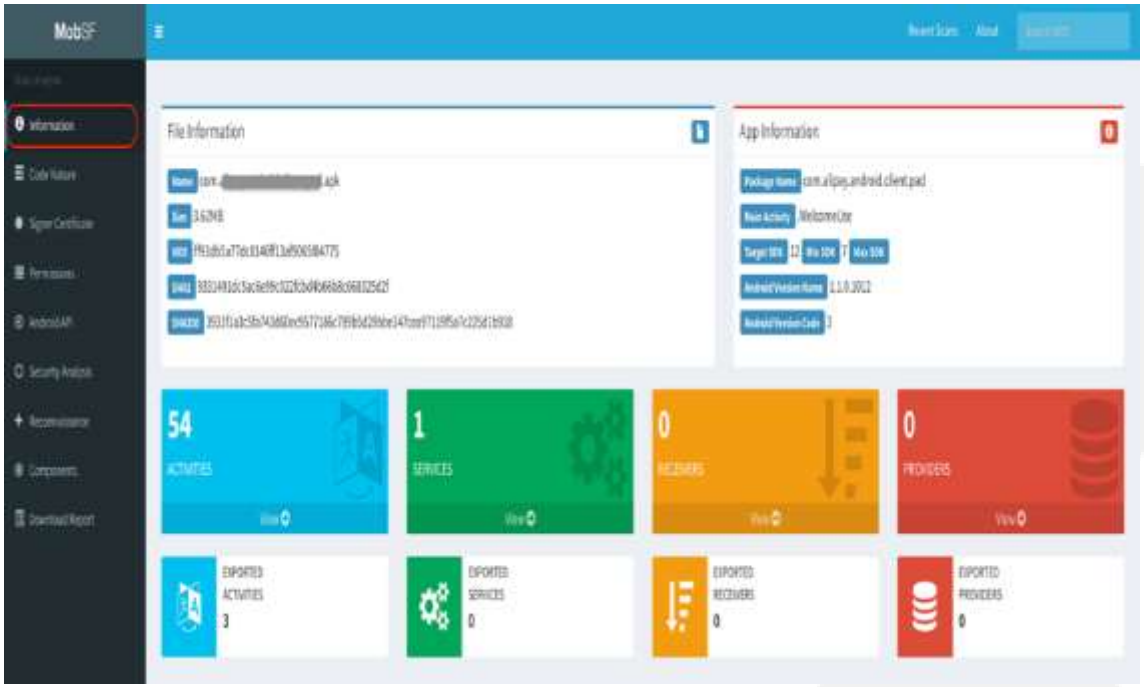
MobSF 開啟畫面

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)



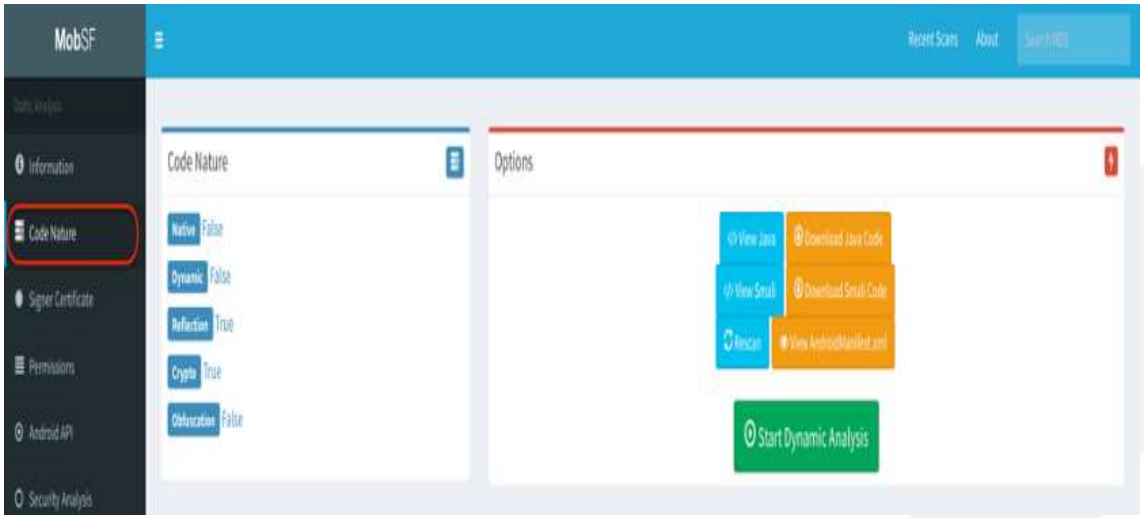
靜態分析- Information(Dashboard)

步驟二(1/8)

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 。

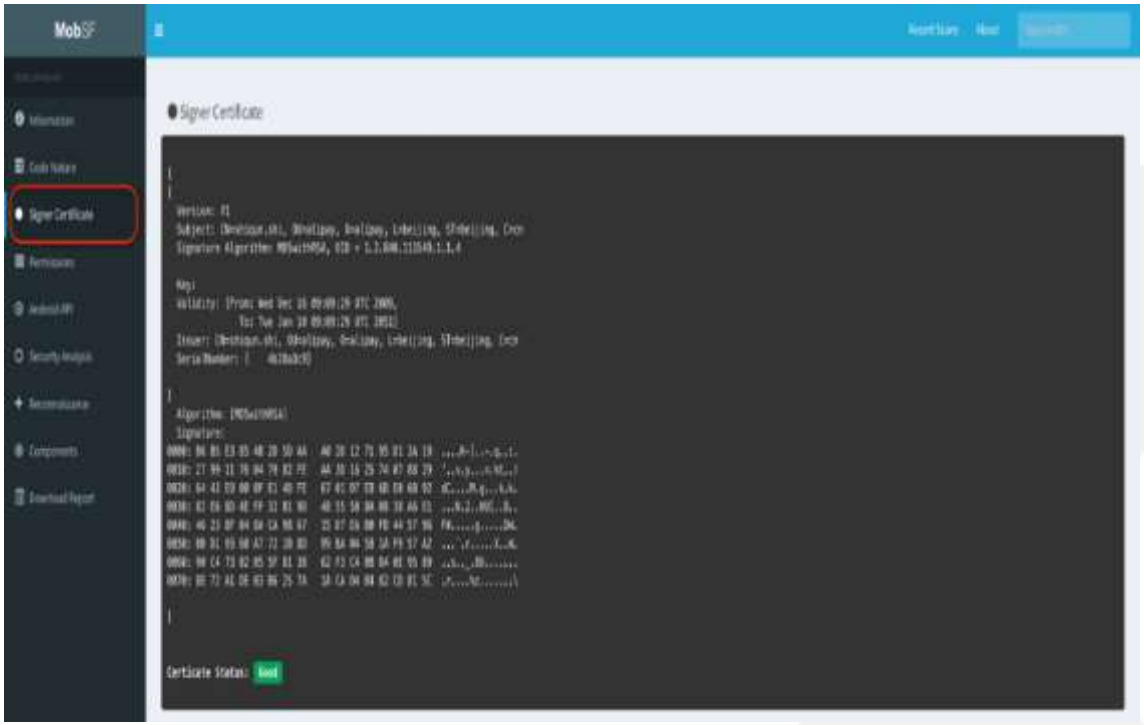
檢測操作步驟-靜態分析(MobSF)



靜態分析- Code Nature

步驟二(2/8)

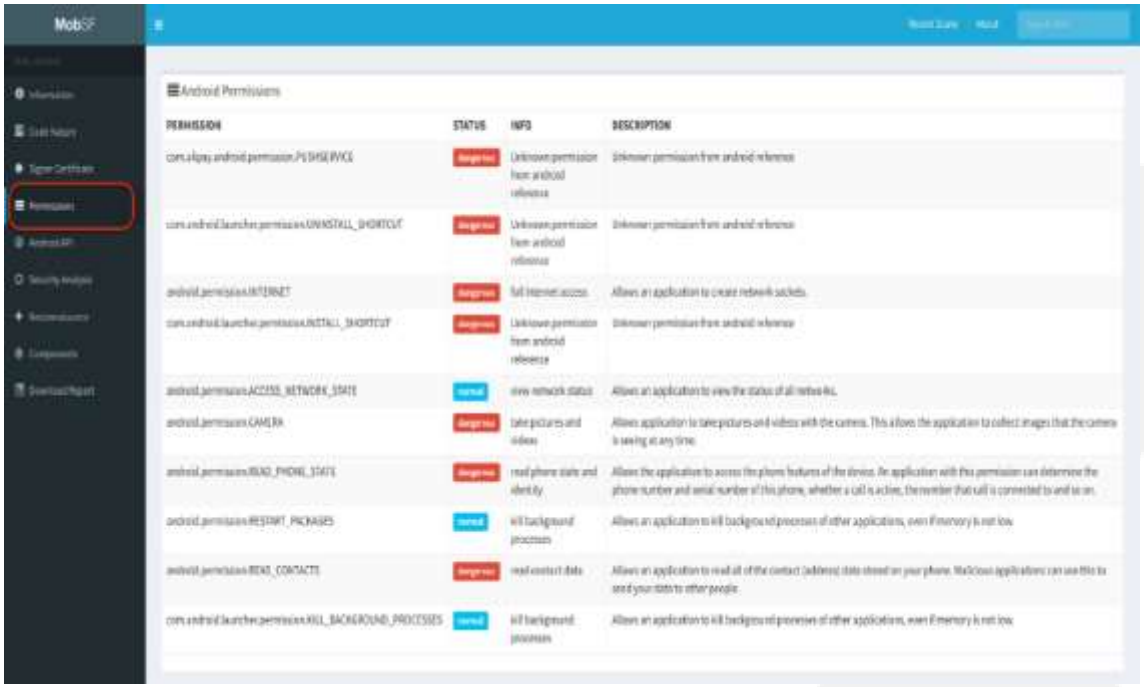
檢測操作步驟-靜態分析(MobSF)



靜態分析- Signer Certificate

步驟二(3/8)

檢測操作步驟-靜態分析(MobSF)



靜態分析- Permissions

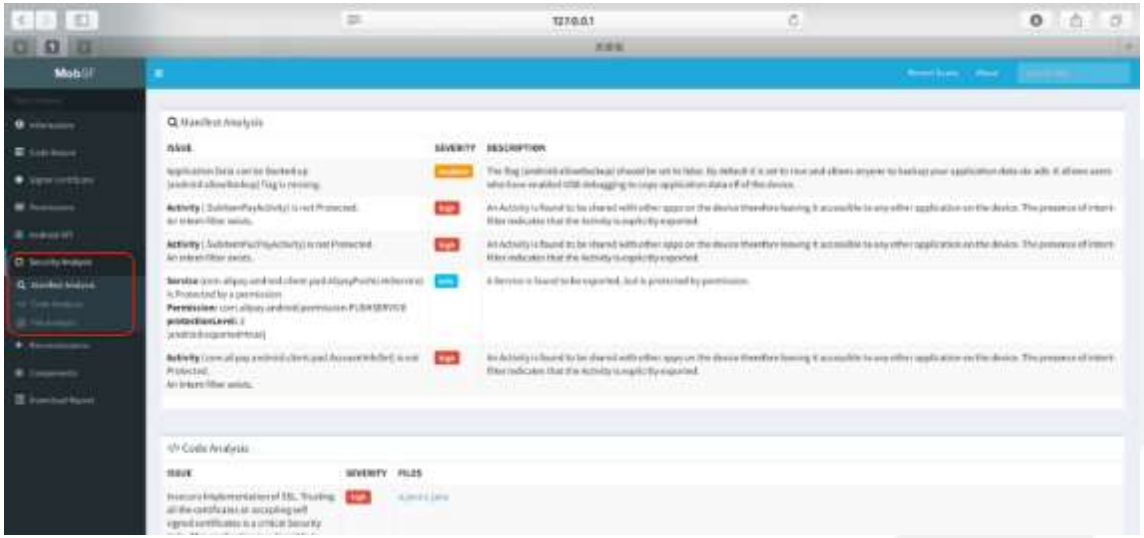
步驟二(4/8)

檢測操作步驟-靜態分析(MobSF)

靜態分析-Android API

步驟二(5/8)

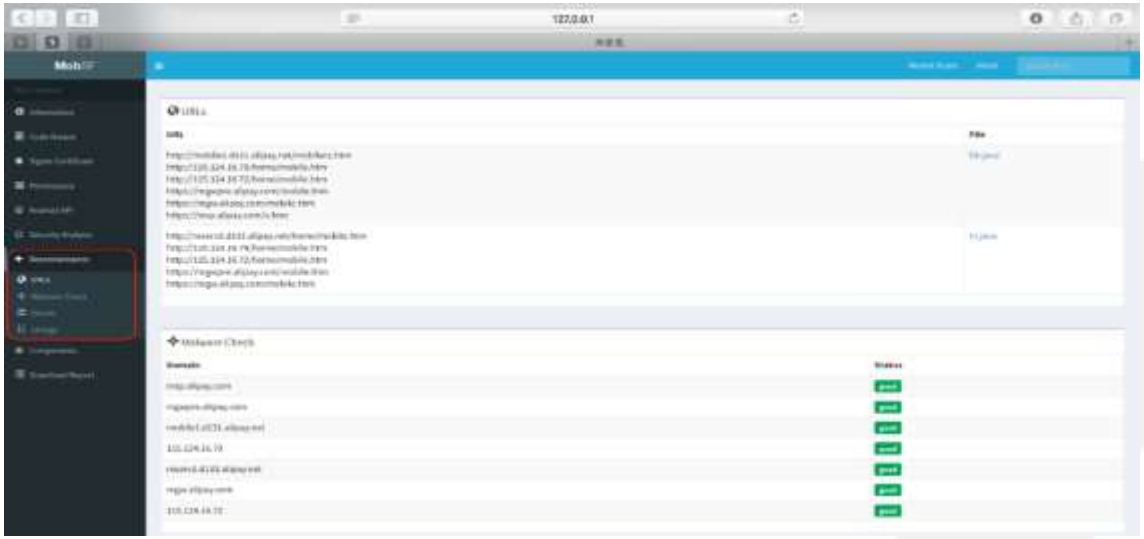
檢測操作步驟-靜態分析(MobSF)



靜態分析- Security Analysis

步驟二(6/8)

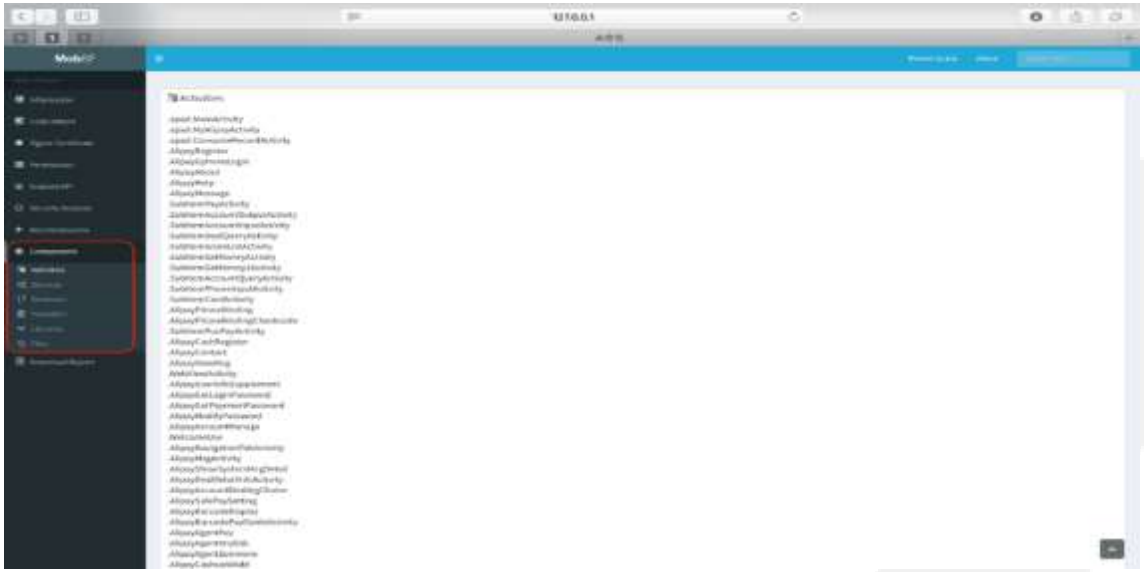
檢測操作步驟-靜態分析(MobSF)



靜態分析- Reconnaissance

步驟二(7/8)

檢測操作步驟-靜態分析(MobSF)



步驟二(8/8)

靜態分析- Components

檢測操作步驟-動態分析(MobSF)

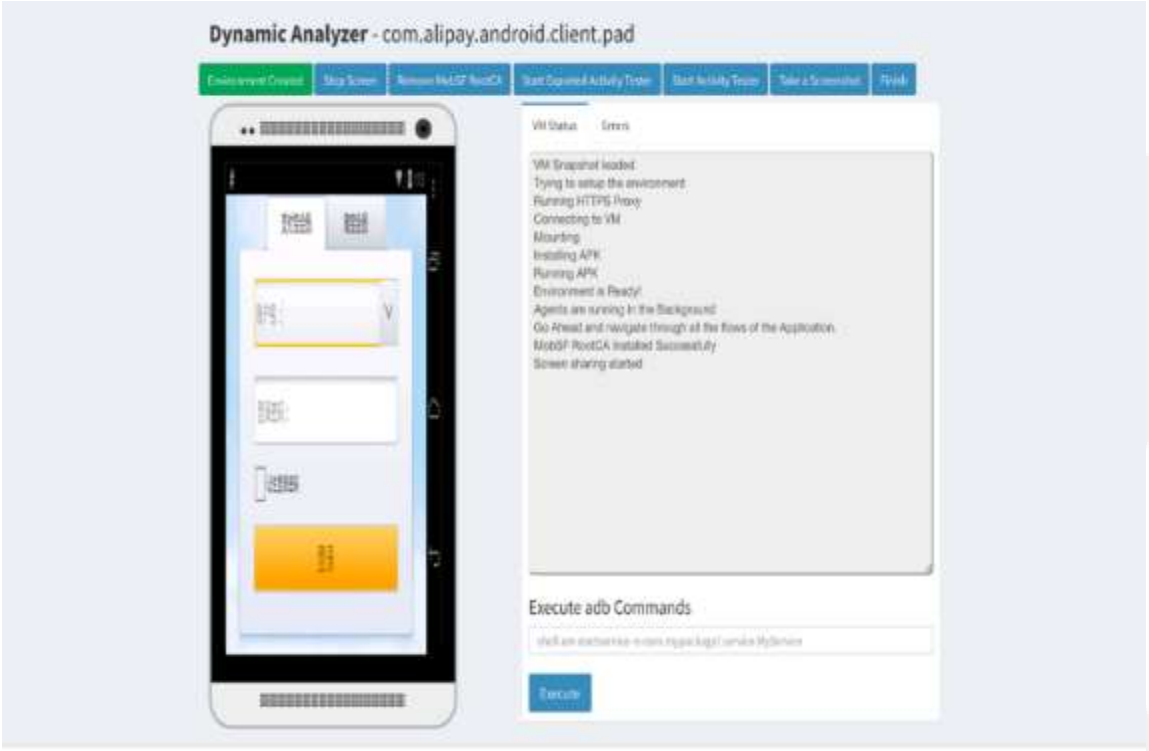
動態分析- 開啟畫面

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)



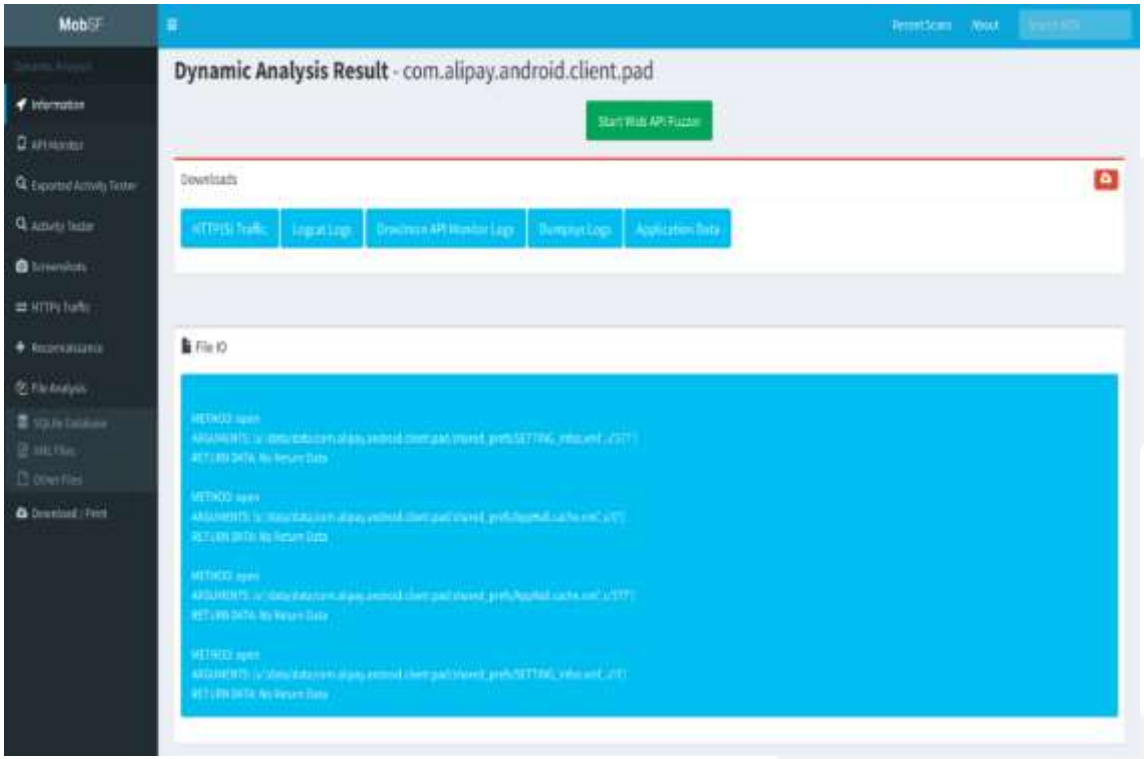
動態分析-測試環境

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)



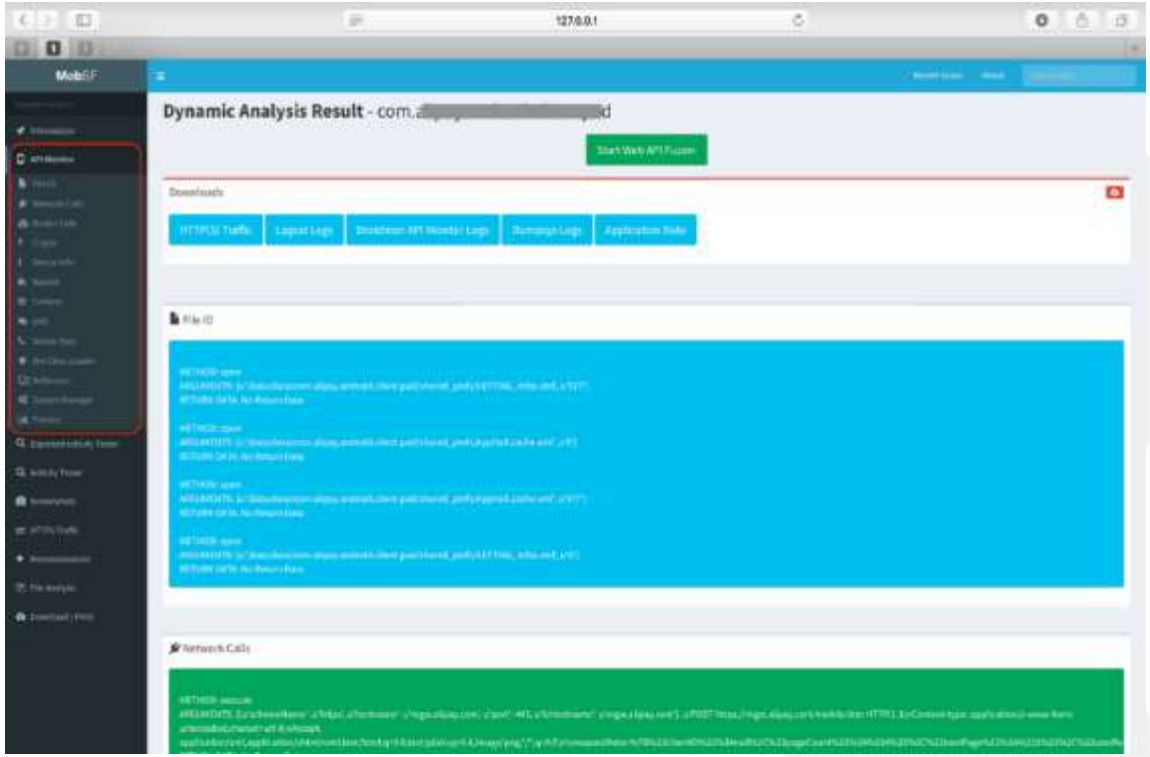
動態分析- Information

步驟五(1/7)

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 。

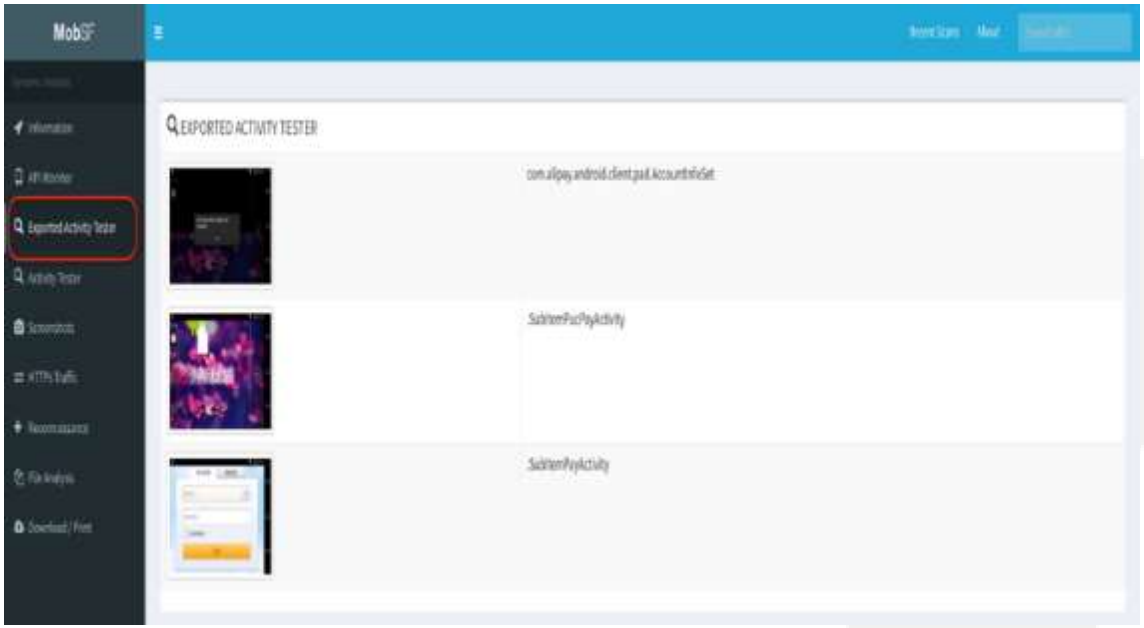
檢測操作步驟-動態分析(MobSF)



動態分析- API Monitor

步驟五(2/7)

檢測操作步驟-動態分析(MobSF)



步驟五(3/7)

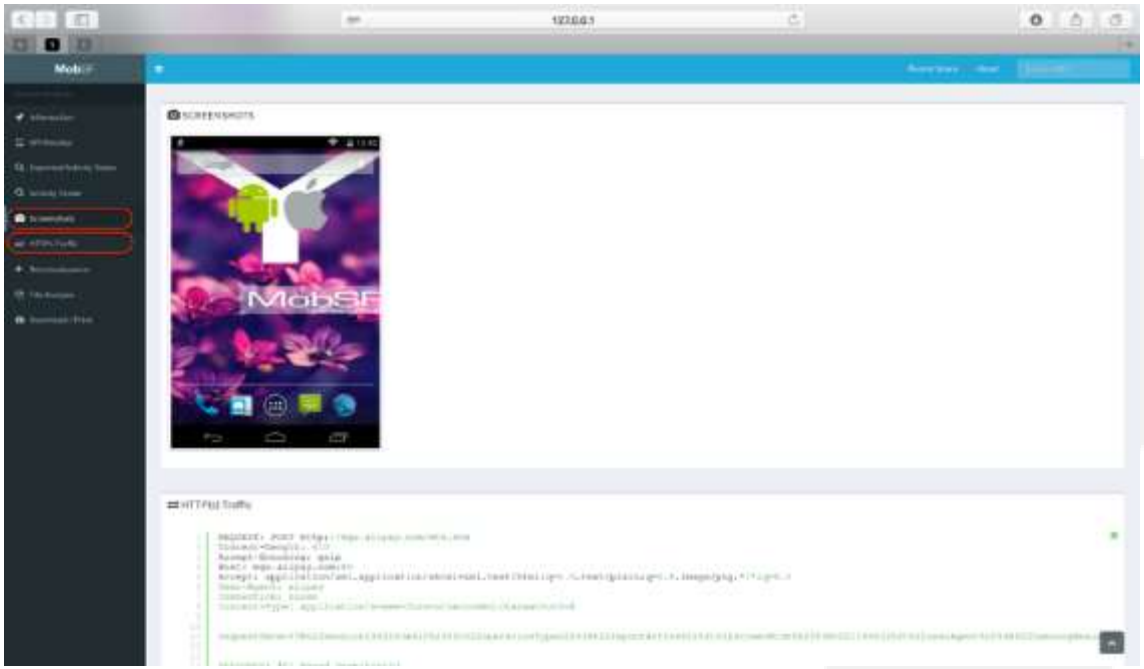
動態分析- Exported Activity Tester

檢測操作步驟-動態分析(MobSF)

動態分析- Activity Tester

步驟五(4/7)

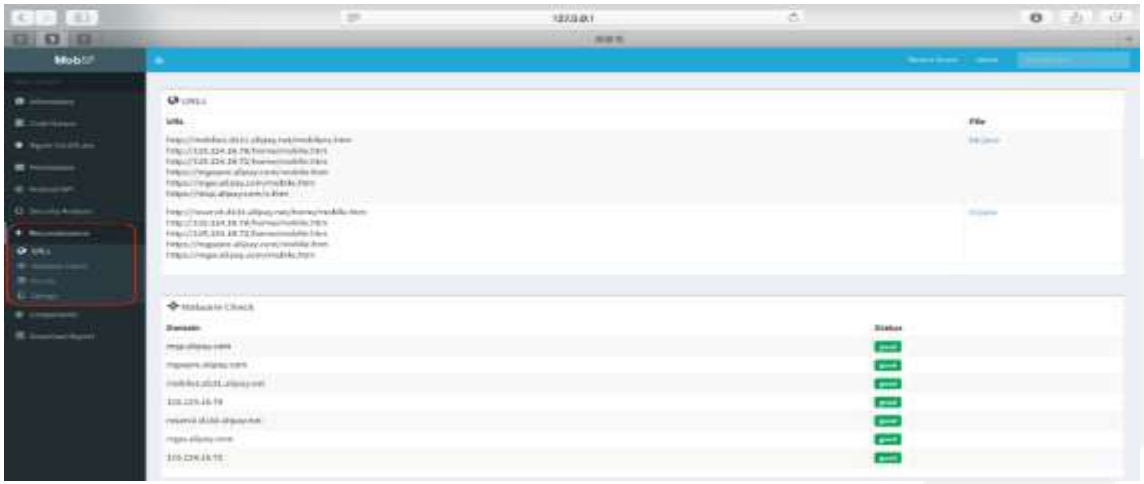
檢測操作步驟-動態分析(MobSF)



步驟五(5/7)

動態分析- Screenshots、HTTPs Traffic

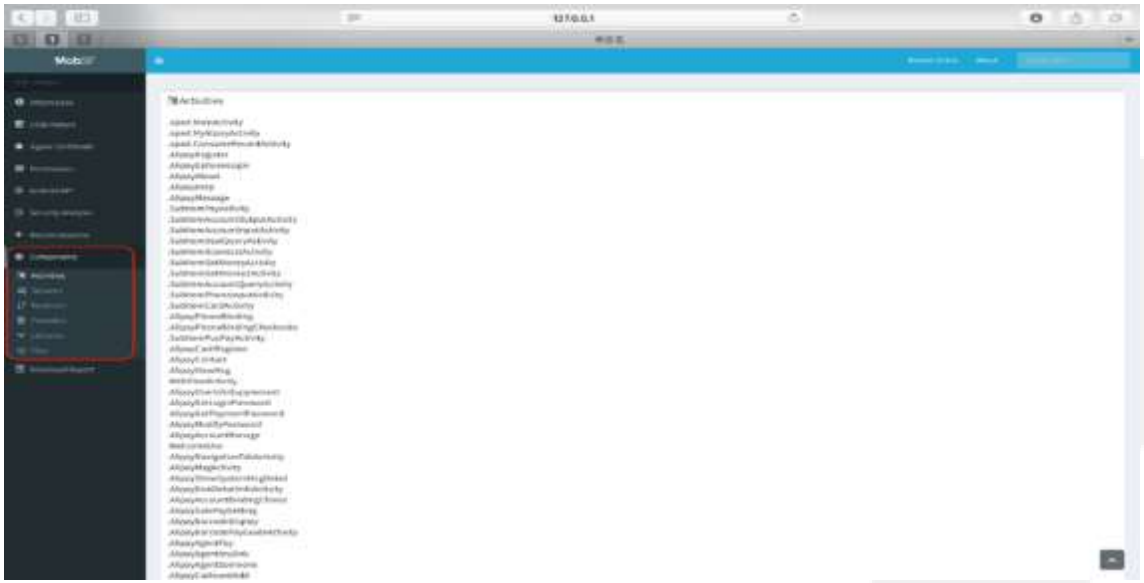
檢測操作步驟-動態分析(MobSF)



動態分析- Reconnaissance

步驟五(6/7)

檢測操作步驟-動態分析(MobSF)



步驟五(7/7)

靜態分析- Components

檢測操作步驟-動態分析(MobSF)

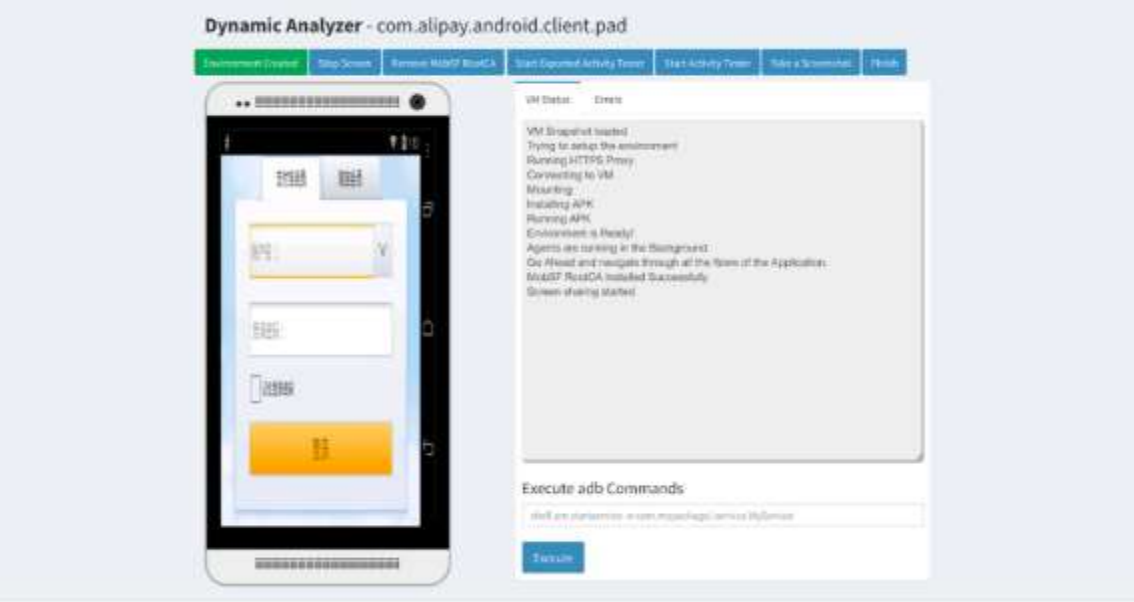
動態分析- 開啟畫面

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)



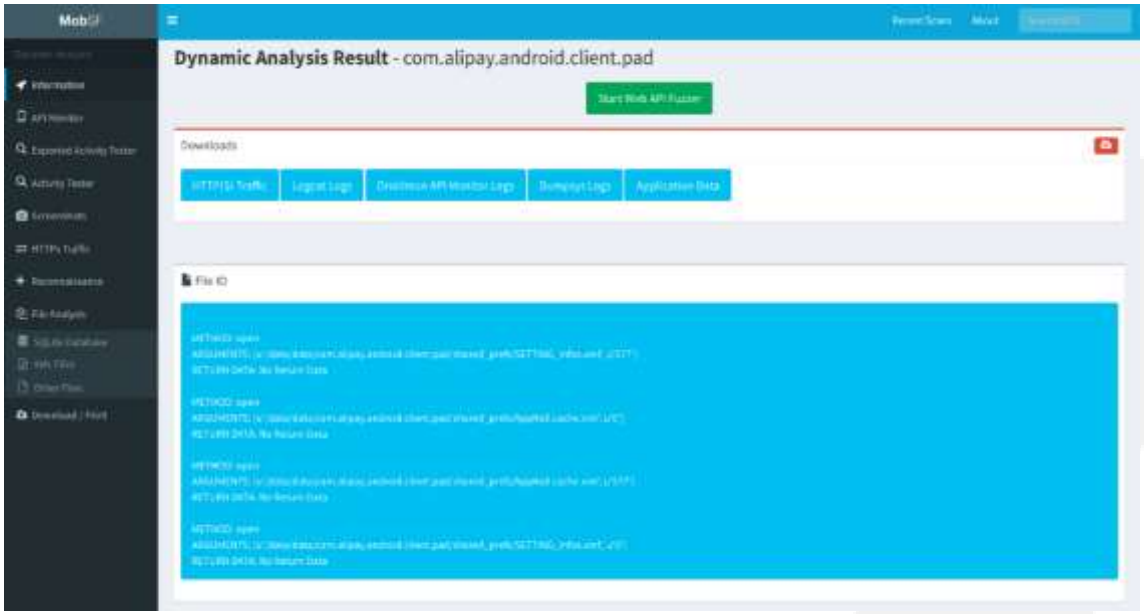
動態分析- 測試環境

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)



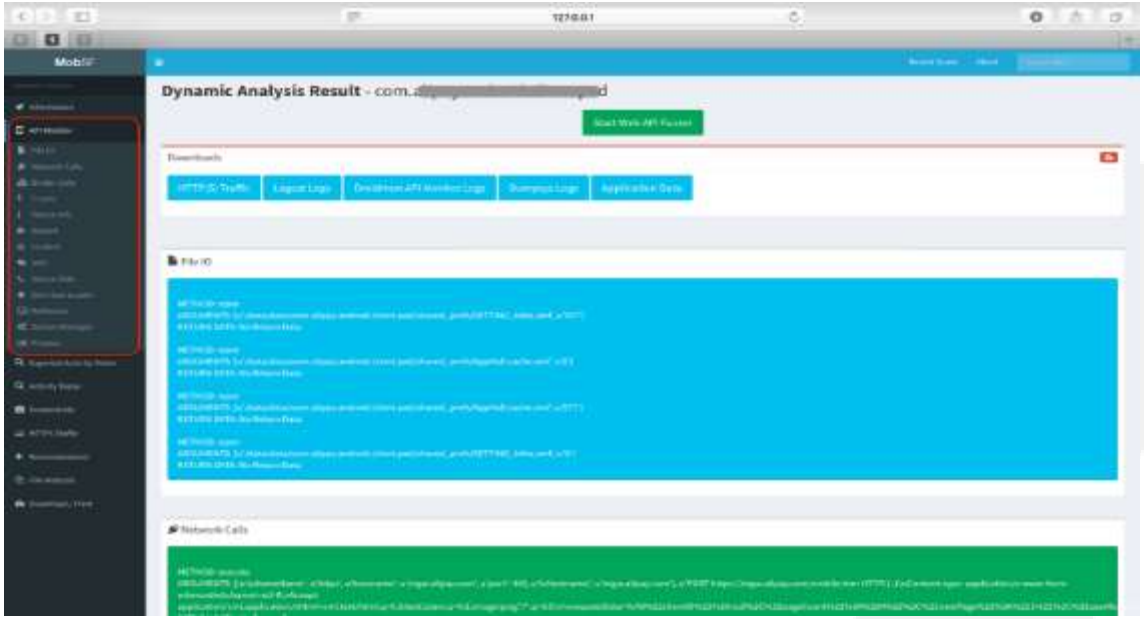
動態分析- Information

步驟五(1/7)

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。。

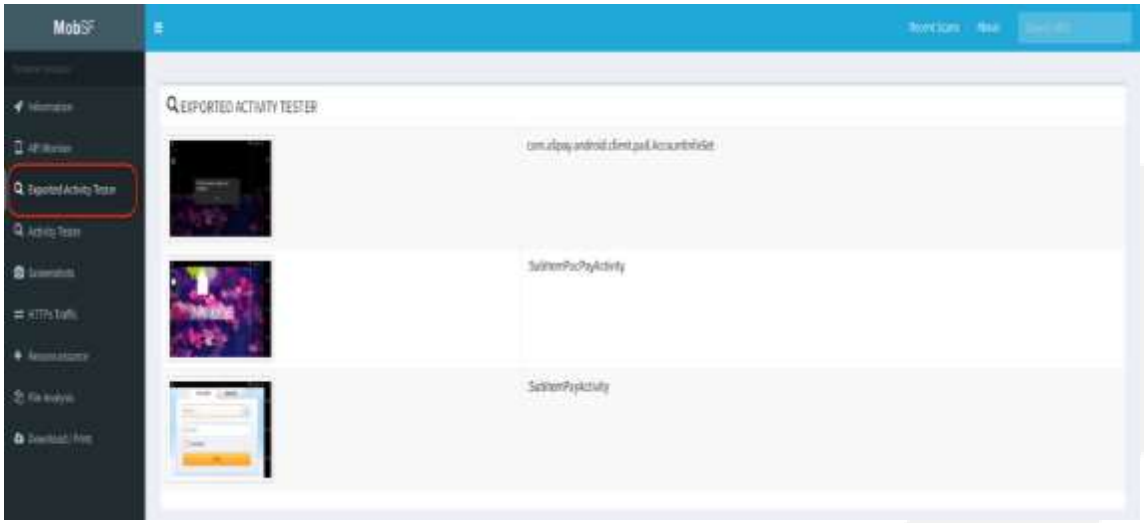
檢測操作步驟-動態分析(MobSF)



動態分析- API Monitor

步驟五(2/7)

檢測操作步驟-動態分析(MobSF)



步驟五(3/7)

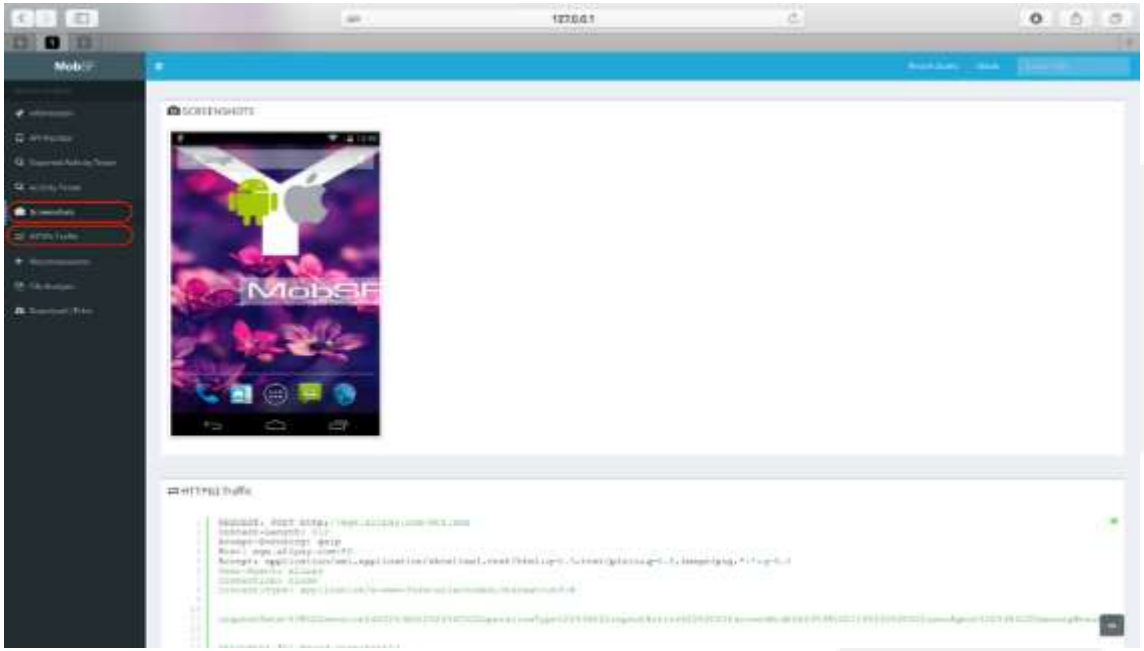
動態分析- Exported Activity Tester

檢測操作步驟-動態分析(MobSF)

步驟五(4/7)

動態分析- Activity Tester

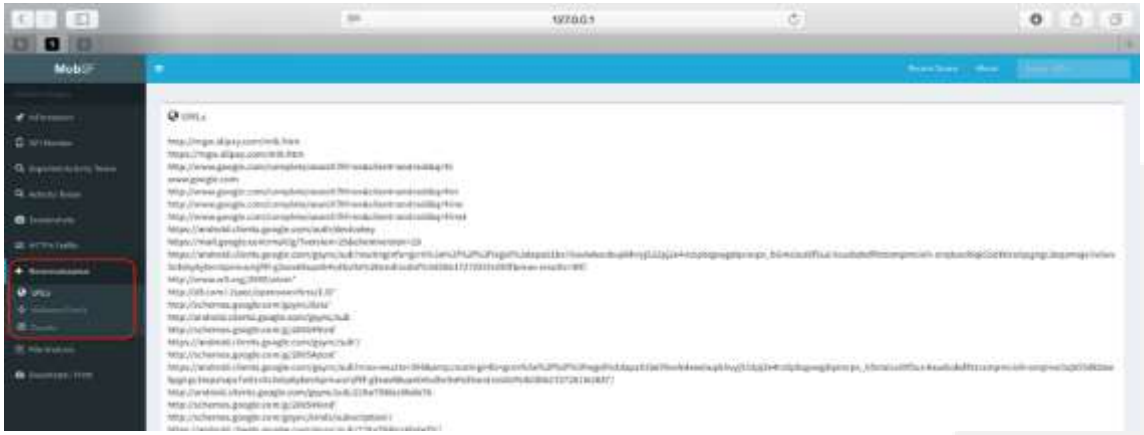
檢測操作步驟-動態分析(MobSF)



步驟五(5/7)

動態分析- Screenshots、HTTPs Traffic

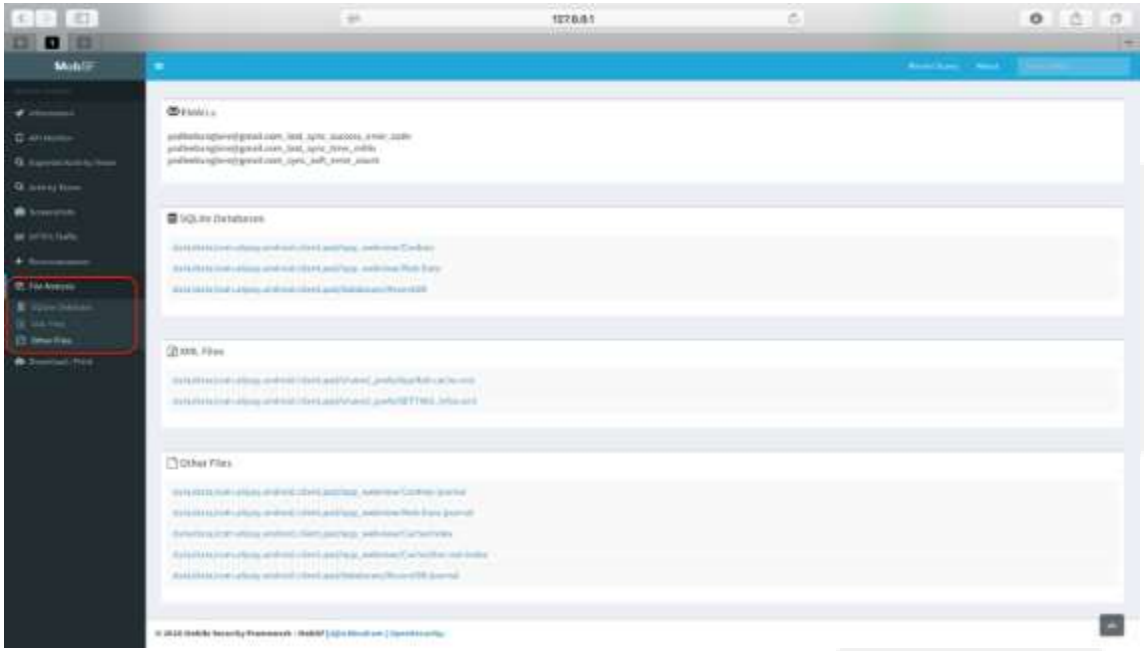
檢測操作步驟-動態分析(MobSF)



動態分析- Reconnaissance

步驟五(6/7)

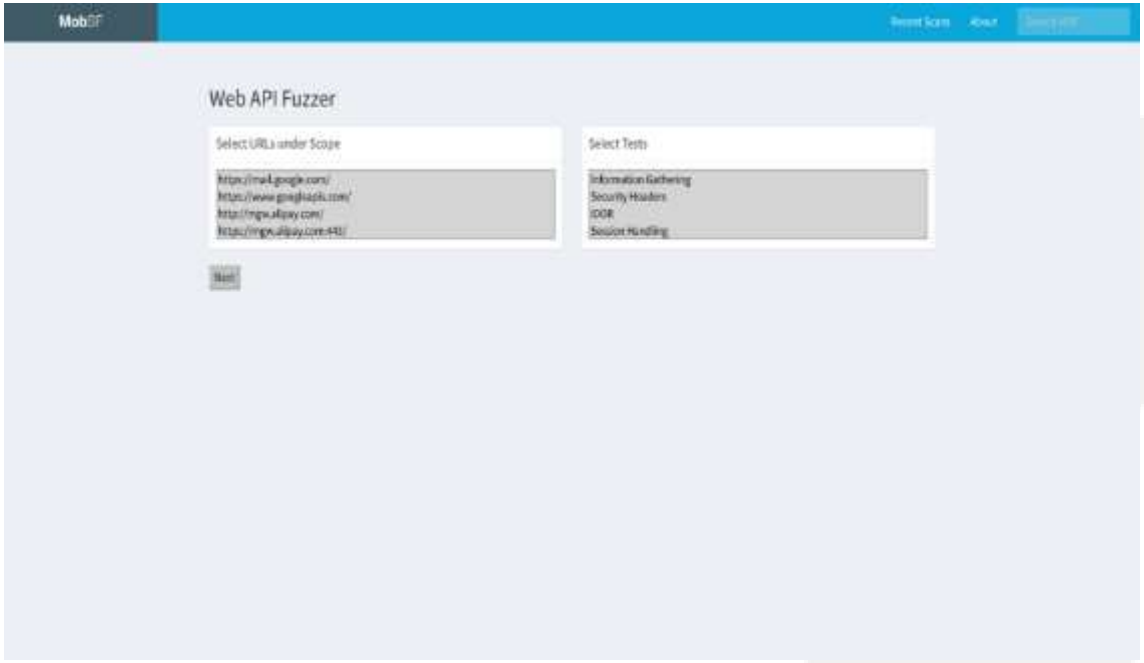
檢測操作步驟-動態分析(MobSF)



動態分析- File Analysis

步驟五(7/7)

檢測操作步驟-Web API 分析(MobSF)



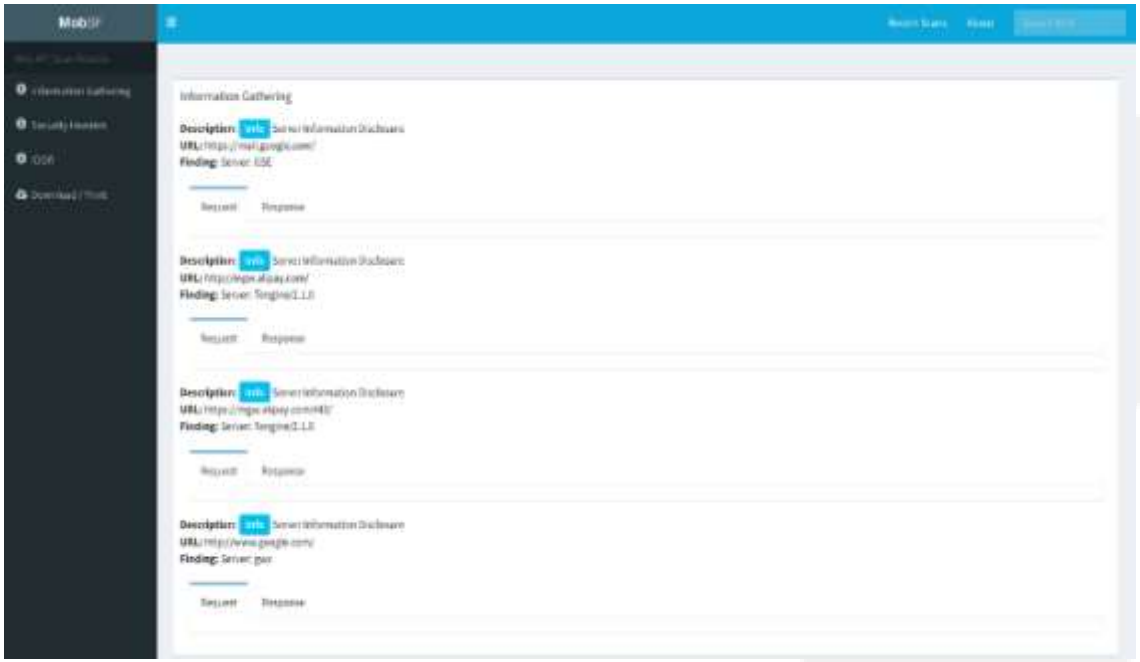
Web API分析- 開啟畫面

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

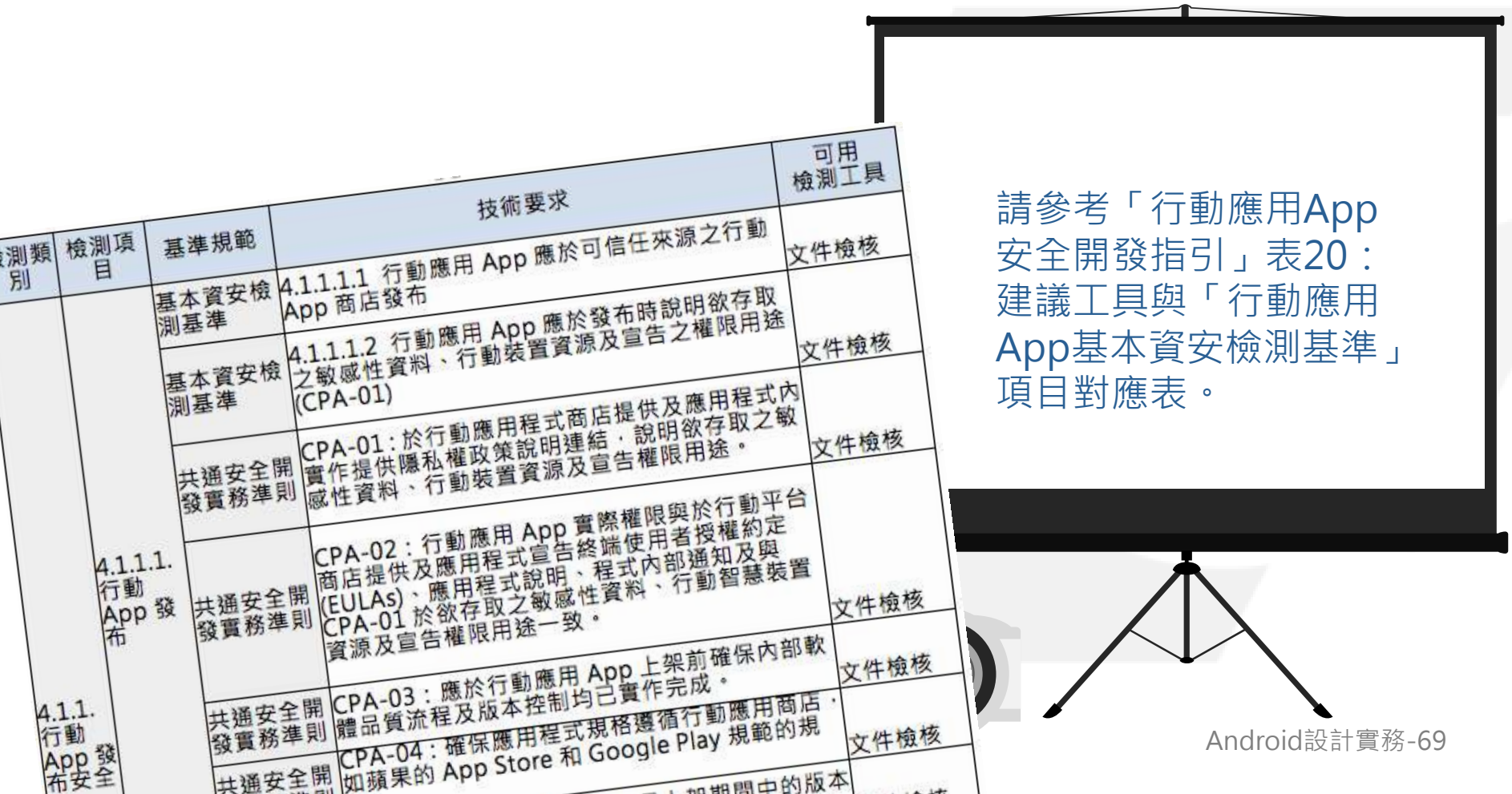
步驟七

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App開發者參考。

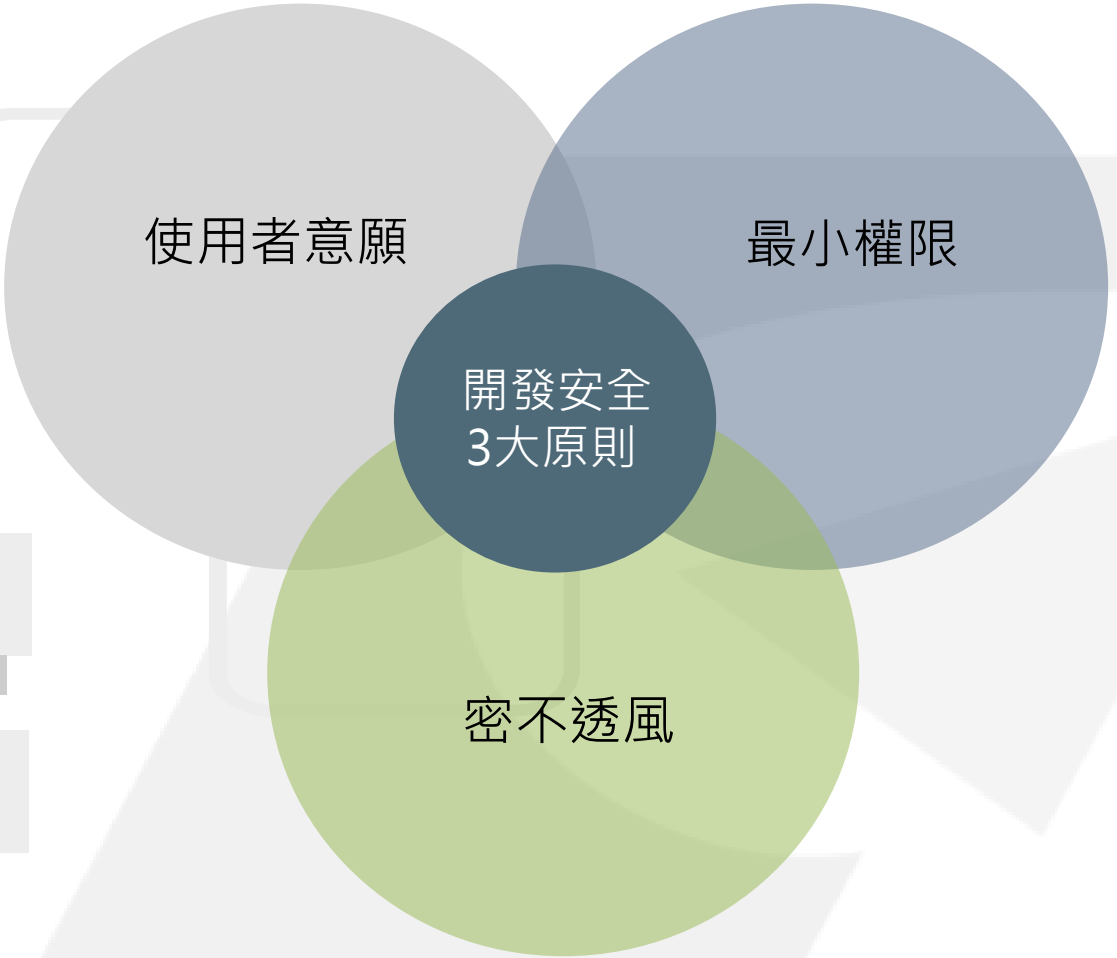


檢測類別	檢測項目	基準規範	技術要求	可用檢測工具
	4.1.1.1. 行動App發布	基本資安檢測基準	4.1.1.1.1 行動應用 App 應於可信任來源之行動 App 商店發布	文件檢核
		基本資安檢測基準	4.1.1.1.2 行動應用 App 應於發布時說明欲存取之敏感性資料、行動裝置資源及宣告之權限用途(CPA-01)	文件檢核
		共通安全開發實務準則	CPA-01：於行動應用程式商店提供及應用程式內實作提供隱私權政策說明連結，說明欲存取之敏感性資料、行動裝置資源及宣告權限用途。	文件檢核
		共通安全開發實務準則	CPA-02：行動應用 App 實際權限與於行動平台商店提供及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