The Developer is the Enemy

R07944058 網媒碩一 陳鵬宇

1. Summary

這篇論文主要想解決 App Developer 在開發 app 時，不一定能夠完善的兼顧 security，一名 App Developer 在開發 app 時，可能會透過 API Developer 所創建的 library 來調用 API，App Develoepr 不是最上游的程式碼供應者，所以會有安全疑慮。這些疏忽造成的漏洞是很嚴重的。論文提到非常多種方法來解決安全問題：安全分析工具、建立安全的 library（從上游下手），帶讀者從 usability、security 和 end security 三個層面探討，並以開發者為第一視角，考量 usability 和 security 的取捨。

2. Strength(s)

論文假設不是所有開發者都是資安專家，有呼應到標題的「The Developer is the Enemy」，就是因為開發者對資安不夠熟稔，所以在開發 app 的過程中，就可能導致漏洞產生。另外論文中提到的流程圖也很棒，能夠讓讀者對 API Developer 向下傳遞 App Developer 再傳遞到 End User 有很明確的藍圖。分成多個 sections 能夠讓讀者針對每一個情況進行獨立思考，不會一次摻雜太多情境，而無法專注在一件單一因素上。

3. Weakness(es)

在 3.3 節中，有提到 security tools 是一個解決方法，但若開發者符合前提假設，不夠了解 security tools 也是白搭，因為他還是需要花時間去了解 security tools 如何運作。要一名開發者去了解資安運作機制也可能不是個好選擇，這樣會分散掉他們開發 app 的精力。4.3 節提到開發者可能會「刻意地」去忽略資安的重要性，而只是為了程式碼能跑快一點。本篇論文提到開發者不善於評估長期結果，但是沒有假設開發者只專注在功能性上，對其它細節不考量，因此我們無從得知是不是所有開發者都真的那麼不看重資安的重要性，更無法知道開發者是否都不評估長期結果。4.4 節更說測試不是優先考量的工作，就我所知，大公司在測試部分的程式碼肯定是少不了的。

4. Reflection

之前沒有想過原來 usability 和 security 可能是很難兼顧的。論文中沒有很明確指出個人資安、系統安全的定義，也沒有相關的實驗數據，這裡可以再多做補充。若能讓資安問題「盡可能地」獨立於功能面，會對開發者、資安專家都好，軟體工程常會有分工的需求，就像論文中有提到一個很棒的例子：開發網頁就讓藝術家專注在介面，而不需要再要求他具備程式設計的能力。

論文有提到 PHP 將 register_globals 移除的例子，因為這會導致資安問題，但這需要開發者重新更改他們的程式碼，我覺得這不是一個很妥當的方式，若能根據舊的資安漏洞去做些防護性的調整，會比直接連根拔起好，當然這是建立在有其它方式能夠修復的前提下。

若能讓開發者了解一定程度的資安知識，讓他們在開發 app 時，就有一定程度意識到「若這樣寫，會不會造成漏洞？是否有更好的寫法？」，論文不斷強調不是每位 App Developer 都是資安專家，一名 App Developer，該花費多少心力去補充資安知識是很值得探討的。若能了解某一程式語言，有可能在何時會產生漏洞，是很有幫助的。

API Developers 可能來自各地，而不是一個團隊，若 API Developers 在創建 API、libraries 時，就能周詳地考量資安防護機制，這樣就能從「源頭」改善資安問題，如此一來調用 API、libraries 的「App Developer」就可以專心在「App」上，而不用再多操心是否會有漏洞、資安問題產生。