Sicurezza degli Smartphone e i rischi dell’obsolescenza precoce

Date : 22 marzo 2018

Prefazione

Diciamocelo, aggiornare i nostri dispositivi è una vera seccatura!

Immaginiamo la giornata tipo di un utente:

- La mattina, all’accensione del cellulare:
  - notifiche di aggiornamento app
  - notifica di aggiornamento del sistema operativo
- Subito dopo, alla propria scrivania di lavoro:
  - quantità indefinita di aggiornamenti del SO
  - notifica di mancato aggiornamento dell’antivirus
- Nel pomeriggio, sul tapis roulant in palestra:
  - notifica di aggiornamento firmware della FitBand/SmartWatch
- La sera, prima di cena:
  - richiesta di supporto da parte della moglie, per similitudine, sui punti precedenti
  - ibidem, in caso di figli
  - ibidem, in caso di animali domestici aventi collare smart!

Stressante.

Tuttavia, l’utente in questione può ritenersi molto fortunato.

Di seguito vedremo il perché, ponendo l’attenzione ai soli dispositivi smartphone.

Le vulnerabilità

Quanto detto in precedenza, seppur in tono ironico, fa realmente parte del vissuto di quasi ognuno di noi.

Spesso si ignora che una notifica di aggiornamento, applicativa o di sistema, non comporta

solamente un abbellimento dell'interfaccia utente o l'aggiunta di ludiche funzionalità. In molti casi questi aggiornamenti volgono su tematiche di sicurezza.

Nonostante i singoli individui non siano tenuti a conoscere le principali minacce relative ai propri dispositivi, il contesto informativo attuale è comunque sufficiente a sensibilizzare l'utente riguardo questa tematica.

Di seguito una carrellata delle maggiori vulnerabilità da un anno a questa parte.

BlueBorne

Scoperta da Armis (Palo Alto), *BlueBorne* rende miliardi di device potenzialmente attaccabili nel raggio di 10 metri, sfruttando una vulnerabilità presente nella connettività Bluetooth (se attivo).

Si tratta di una *remote code execution* su *Bluez*, un componente del kernel che fornisce le funzionalità relative alla gestione di periferiche BT. La vulnerabilità, insita nell'implementazione del livello L2CAP dello stack, è causata dalla copia di un buffer senza previo controllo della dimensione (*buffer overflow*).

BlueBorne può servire qualsiasi obiettivo malevolo: cyber-spionaggio, furto di dati, diffusione di ransomware o creazione di *botnet* composte da dispositivi IoT.

Vulnerabili: Google Android Risolto: iPhone >= 5S

KRACK

Scoperta da due ricercatori belgi *Mathy Vanhoef* e *Frank Piessens*, è una vulnerabilità del protocollo di crittografia WPA2 che protegge la trasmissione dati sulle reti Wi-Fi.

Il metodo non tenta di decrittare il traffico Wi-Fi. Il *Key Reinstallation AttaCK* prende di mira il meccanismo che si viene a stabilire tra il supplicant (device) e l'authenticator (router): l'attaccante si intromette con un "man in the middle" nell'interscambio di messaggi, forzando l'authenticator a generare una chiave più semplice (composta di soli zero) di quella originaria. Tutto all'insaputa dell'utente. È chiaro quindi che con una simile chiave decrittare i dati è molto semplice.

Volendo semplificare, è come se un ladro decidesse che è inutile tentare di scovare la combinazione di una cassaforte, quando è più facile spiare chi la conosce e la sta utilizzando. Anzi, in questo caso è come se il ladro avesse il potere di convincere il malcapitato a reimpostare la combinazione con una sequenza di zeri.

Vulnerabili: Google Android Risolto: iPhone >= 5S

Meltdown e Spectre

Sono due vulnerabilità scoperte dai ricercatori di Google Project Zero.

Immaginiamo che tutti i dispositivi siano vulnerabili ad esfiltrazione di dati da parte di malintenzionati. Aggiungiamo che l'*exploit* non sia rilevabile dai software antivirus poiché sfrutta il normale funzionamento delle CPU, consentendo l'accesso non autorizzato ai registri di memoria dei processori.

Infatti, tutto parte dai dati: una CPU carica le informazioni, richieste da un processo, dalla RAM. Nel contempo verifica che tali dati non siano prerogativa del kernel, approvando o negando l'operazione.

Il problema nasce dal meccanismo di *esecuzione speculativa*: i core delle CPU usano *pipeline* multiple per processare le istruzioni parallelamente. In caso di *branch* (diramazione) di un'istruzione, si genera uno stallo che porta la pipeline a passare da una sequenza di istruzioni ad un'altra, causando una perdita di tempo in termini di cicli di clock, in attesa di nuovi dati dalla RAM.

Per evitare che ciò avvenga, ad ogni branch, un'*unità di predizione* prova a indovinare la sequenza di istruzioni di cui il processore avrà bisogno in seguito. Il problema risiede qui: nel modo in cui il sistema gestisce l'accesso alla memoria *cache* durante queste esecuzioni speculative.

I normali controlli di sicurezza che separano lo spazio utente e la memoria kernel non si compiono con la necessaria *rapidità*. Di conseguenza il processore può momentaneamente recuperare, ed eseguire, dati dalla memoria cache a cui di norma non dovrebbe avere accesso.

Vulnerabili: Google Android Risolto: iPhone >= 5S

chaiOS

Il bug è stato reso noto da Abraham Masri tramite twitter.

In sostanza è stato scoperto che se si inserisce una gran quantità di caratteri nei metadati di una pagina Web *Safari* va in *crash*, portandosi dietro tutto il SO con il conseguente blocco del dispositivo.

Ciò deriva da un mancato controllo del buffer utilizzato per creare il *preview* delle pagine all'interno delle app (*buffer overflow*).

Risolto: iPhone >= 5S

Carattere indiano

La vulnerabilità riguarda un carattere della lingua Telugu. Se visualizzato in una qualsiasi app di messaggistica su iOS ne provoca un crash, bloccando quest'ultima fino a quando non avviene l'eliminazione dello stesso dal messaggio che lo contiene.

Peggio ancora, se iOS tentasse di mostrare il carattere in un messaggio di notifica, il crash potrebbe coinvolgere l'intera *Springboard*[1]. In questo caso iOS entrerebbe in uno stato di *bootloop*, rendendo inutilizzabile il dispositivo fino ad un successivo reset in modalità DFU[2], causando la perdita di tutti i dati memorizzati.

Risolto: iPhone >= 5S

Cloak & Dagger

Scoperta da un team di ricercatori della *University of California* e del *Georgia Institute of Technology*, la vulnerabilità colpisce il SO Android.

Questa si basa sull'autorizzazione "draw on top" ("disegna sopra altre applicazioni") che viene concessa come impostazione predefinita alle app scaricate da Google Play. In questo scenario, un'app malevola può sfruttare questa autorizzazione per indurre l'utente ad abilitare l'autorizzazione "a11y" (contrazione di "Accessibility") mediante clickjacking (cattura dei clic sfruttando elementi di interfaccia ingannevoli).

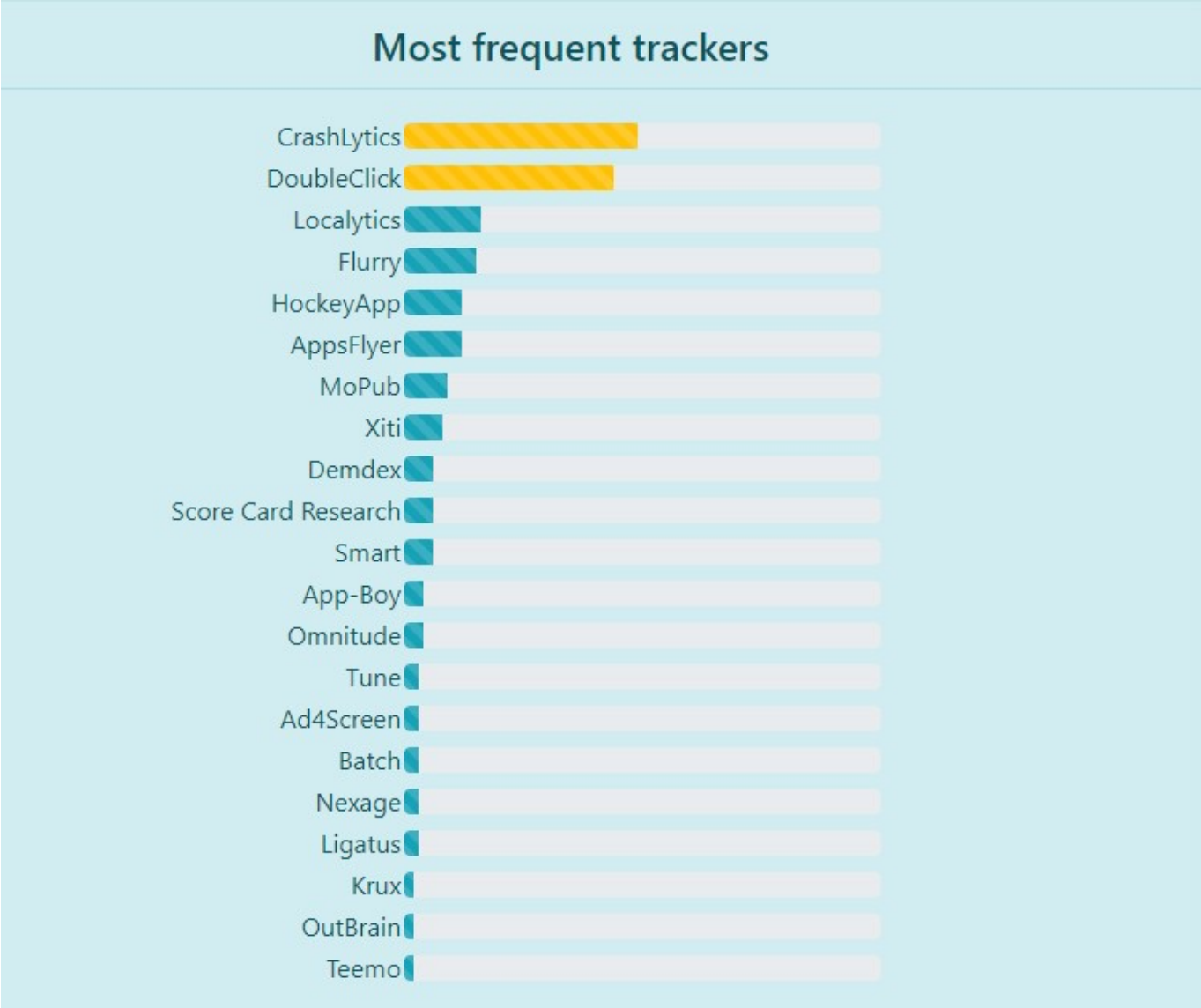
Se condotti con successo, gli attacchi consentono ad un'app di interferire nell'interazione con l'interfaccia utente ed ottenere il controllo completo del dispositivo all'insaputa del proprietario.

Vulnerabili: Google Android

Google Play

In conclusione, non si possono non menzionare le problematiche legate al Play Store di Google.

Una ricerca condotta dalla Princeton University mette in evidenza un gigantesco sistema di sorveglianza che sfrutta una serie di JavaScript inseriti nelle app, volti a registrare tutto quello che facciamo: dai dati inseriti ai tocchi sullo schermo fino ai clic sulla pagine.



Senza contare che tali app, spesso inserite sullo Store da sviluppatori di origine sconosciuta, possono contenere non solo codice malevolo, ma anche pericolosi bug.

In tal senso, poniamo l'attenzione su app critiche come quelle sviluppate per i sistemi SCADA[3]. Da un recente report, si evince come tali app sul Play Store si siano rivelate un vero colabrodo in quanto a sicurezza.

L'obsolescenza precoce

Le vulnerabilità descritte in precedenza sono tempestivamente risolte dagli sviluppatori nel caso in cui i bug riguardino le app. Viceversa, la questione è più complicata se tali criticità riguardano i SO.

Quanto detto nell'articolo riguardante la memorizzazione dei dati è estendibile agli aggiornamenti del SO: analizziamo ora gli approcci dei maggiori produttori.

Allo stato dell'arte, per quanto riguarda iOS, possiamo constatare che device fino all'iPhone 5S, commercializzato nel 2013, continuano a ricevere aggiornamenti di sistema e di sicurezza con l'ultima versione di iOS (11). Abbiamo quindi circa 5 anni di finestra di supporto al device.

Per quanto riguarda Android, i dispositivi marcati Google (Pixel) hanno una finestra di supporto al device di circa 3 anni; per i produttori di terze parti, tale finestra non supera mediamente i due anni dalla data di uscita del dispositivo.

A conferma, in questo** articolo **sono riportati i dispositivi che continueranno ad essere supportati tramite l'ultimo SO Android. Una rapida occhiata rivela come dispositivi quali Sony Xperia Z5 o Huawei P9, con caratteristiche HW degne di nota, non compaiano in tale lista.

Di conseguenza, dispositivi con simili caratteristiche non riceveranno mai gli ultimi aggiornamenti di sicurezza, rimanendo esposti a gran parte delle vulnerabilità scoperte nel recente passato.

È inutile, anche in questo caso, approfondire la scena Windows Phone, poiché il progetto è stato dichiarato** concluso**.

Conclusioni

In base a quanto detto sarebbe buona prassi, nel momento in cui si sceglie uno smartphone, aver cura di valutare non solo gli attributi puramente quantitativi (CPU, RAM, GB di memoria, megapixel, ergonomia, ecc.), ma anche gli attributi qualitativi relativi alla sicurezza ed alla longevità del progetto.

È totalmente sconsigliato farsi trarre in inganno da dispositivi "simulacri" delle marche più blasonate. Di solito questi ultimi sono progettati in paesi dell'est ed hanno caratteristiche hardware apparentemente di livello. In realtà, un occhio ben attento e un pizzico di esperienza, possono portare in risalto l'arretratezza (o peggio, considerando quanto riportato** qui**) del SO installato e la totale mancanza di supporto ed aggiornamenti di sicurezza da parte della casa produttrice.

In concreto, si vuole solamente sottolineare cosa vi è dietro al prezzo di un dispositivo certificato dalle aziende più blasonate: ingenuamente considerati "apparecchi che fanno le stesse cose che fanno gli altri", in realtà questi ultimi sono il risultato di una serie di studi, accortezze, qualità del supporto, che altri competitor non garantiscono.

Acquistare un dispositivo che rientra in quest'ultima casistica, riduce notevolmente il rischio di ritrovarsi tra le mani un prodotto precocemente obsoleto e di essere esposti ad un elevato numero di vulnerabilità che non saranno mai risolte.

Note

[1] ***La SpringBoard è l'applicazione che gestisce la schermata Home di iOS.***
[2] ***DFU, Device Firmware Update***
[3] ***Sistemi Scada:*** https://it.wikipedia.org/wiki/SCADA

A cura di: Daniele Rigitano