INTRODUZIONE AL ROUTING DELLO SMARPHONE

NEOFITI

- breve introduzione su android e linux
- cos’è e perché effettuare il rooting
- perché non bisogna prendere sottogamba il rooting
- le partizioni di android (recovery, system, data, bootloader)

MEDIUM

- rooting
- accedere alla modalità fastboot e alla recovery
- cosa sono adb e fastboot?
- perché e come sbloccare il BL
- perché e come sostituire la recovery

ADVANCED

- tipi di rooting e le relative app
- cosa fare dopo il rooting (idee creative)

Cos’è Android?

Android è un sistema operativo basato sul kernel Linux nato per essere utilizzato su dispositivi mobili, generalmente con touchscreen, il cui sviluppo è principalmente sostenuto da un consorzio di aziende cui fa capo Google.

Sebbene sia nativamente distribuito sotto licenza Apache, quindi open-source nonché liberamente modificabile e redistribuibile, i produttori dei dispositivi che lo utilizzano aggiungono parti di software non-libero per integrarlo con i relativi servizi proprietari. Un esempio tipico sono i Google Play Services e le GApps.

Cos’è il rooting e perché effettuarlo?

A differenza dei sistemi GNU/Linux che siamo abituati a utilizzare su PC desktop e portatili, i dispositivi che utilizzano Android, all’uscita dalla fabbrica, sono configurati per non permettere l’accesso e la modifica a file e partizioni di sistema. Da qui la necessità di una procedura che torni a renderci veri proprietari del nostro terminale: il rooting.

Il rooting, per definizione, è il processo che permette di ottenere i privilegi dell’utente root, quindi il pieno controllo su file, processi, partizioni di sistema e quant’altro presente all’interno del dispositivo su cui è installato Android e su cui normalmente non si potrebbe intervenire.

Una volta eseguito il rooting possiamo effettuare una moltitudine di operazioni: dal liberarci di quelle noiose app preinstallate nel sistema, altrimenti note come bloatware, all’abilitare funzioni normalmente disabilitate dalla fabbrica, fino a flashare una versione “cucinata” di Android (custom ROM) per aumentare le performance del dispositivo, accrescerne le funzionalità o semplicemente per avere un maggior grado di personalizzazione.

Perché non bisogna prendere sottogamba il rooting?

Poiché, come detto in precedenza, il root espone il sistema alla mano del suo utente, è bene ricordare anche come questo possa essere estremamente pericoloso qualora non si abbia una completa conoscenza dei comandi che si impartiscono o delle regole che si configurano.

Da grandi poteri derivano grandi responsabilità!

Per fare un esempio, una policy troppo permissiva nelle concessioni dei privilegi amministrativi potrebbe lasciare che applicazioni di cui non si ha totalmente chiaro il funzionamento causino danni irreparabili al sistema o si approprino di informazioni personali: come vale la pena ricordare nuovamente, un processo che ha privilegi amministrativi può fare QUALSIASI cosa all’interno del sistema, se siamo noi è un conto ma se fosse un altro all’interno del nostro dispositivo?

Menzione d’onore merita anche la possibilità di brick del dispositivo in caso non ci si attenga scrupolosamente alle indicazioni della guida, ovvero di perdere ogni possibilità di poterlo riutilizzare, trasformandolo in un costoso (o no) fermacarte.

Infine è altrettanto importante sapere che molto spesso il rooting o la procedura per ottenerlo invalida irreparabilmente la garanzia del produttore, quindi è consigliato controllare bene prima di procedere, qualora si volesse evitare.

Come viene gestita l’archiviazione?

Per meglio capire a cosa ci si riferisca quando si parla di zone normalmente limitate all’utente è utile avere un idea di come sia strutturata la memoria flash di tali dispositivi, essi siano tablet o smartphone. Tra le tante, è possibile osservare le principali partizioni:

- **/boot** contiene kernel e ramdisk ed è rigorosamente read-only all’uscita di fabbrica, ovvero non consente modifiche da parte di un utente non privilegiato (non root);
- **/system**, anch’essa read-only, è la sede effettiva del sistema operativo, così come interfaccia utente e applicazioni di sistema;
- **/recovery** (read-only) è una partizione avviabile contenente una console utilizzata per effettuare le operazioni di amministrazione del sistema più comuni, di cui parleremo a breve;
- **/data** è il luogo dove sono riversate le informazioni generate dall’utente tramite il sistema e le applicazioni installate (tipicamente un reset di fabbrica non fa altro che formattare questa partizione);
- **/cache**, come si può immaginare, contiene i file di cui le applicazioni fanno frequente uso (viene anch’essa formattata, insieme a /data, durante un factory reset);
- **/sdcard** è la partizione dedicata ai file utente, multimediali (foto, musica, video) e documenti

In cosa consiste il rooting?

Ci sono diverse tecniche che permettono di accedere ai privilegi amministrativi del sistema, passando da una semplice app one-click, proseguendo per il flash del binario “su” e finendo con lo sblocco del bootloader, sostituzione della recovery e modifica della partizione /boot, in caso di systemless rooting.

In base al produttore (quindi alle policy di sicurezza che adotta) e alla versione di Android, si decide come e da dove partire.

Nel caso sia sufficiente una delle cosiddette app “one-click” basta scaricarne il pacchetto .apk, installarlo e lasciare che quest’ultimo si prenda tutta la fatica del lavoro.
A fini esemplificativi ma non esaustivi si possono ricordare RootMaster, Framaroot, BaiduRoot e Kingo Root.

In tutti gli altri casi sarà necessario l'uso del pc e ci si calerà nella giungla di exploit dove solo la comunità di XDA può correrci in aiuto.
Basta infatti una semplice ricerca con "root $modello" per trovare la procedura da seguire per il proprio dispositivo.

Cosa sono adb e fastboot?

Per essere in grado di operare dal pc bisogna innanzi tutto armarsi degli SDK Platform Tools, nello specifico di adb e fastboot.

Essi sono due strumenti fondamentali per l'utente che si cimenta nello sviluppo di Android e anche nel nostro caso permettono alcune funzioni richieste per il rooting.
Adb (acronimo di Android Debug Bridge) permette, una volta connesso il telefono via USB, di installare app, riavviare il dispositivo nella partizione desiderata e di aprire una shell unix direttamente in Android.

Fastboot invece è un protocollo di comunicazione con il bootloader, sempre attraverso interfaccia USB. Ci consente, attraverso l'omonima modalità di avvio del dispositivo, di eseguire il flash di partizioni, dell'intera memoria, il boot di immagini e lo sblocco del bootloader.
Per accedere a tale modalità è sufficiente tenere premuta una combinazione di tasti a dispositivo spento (la più comune è composta dal tasto di accensione e dal tasto per abbassare il volume contemporaneamente) finché non compare l'immagine del robottino verde e la scritta Download Mode o Fastboot Mode.
È considerata l'ultima spiaggia nel caso in cui tutte le altre partizioni siano compromesse ad un livello tale da non permetterne più l'avvio, compresa la recovery.

L'accesso alla recovery invece segue la stessa procedura della modalità fastboot con la sola differenza della combinazione di tasti premuti (anche in questo caso la più comune è il tasto di accensione e il tasto per alzare il volume contemporaneamente), ne parleremo approfonditamente a breve.

Perché e come effettuare lo sblocco del bootloader

Da diversi anni a questa parte diventa requisito sempre più frequente per il rooting, se non quasi obbligatorio, lo sblocco del bootloader (contenuto nella partizione di /boot di cui parlavamo poc'anzi).

All'interno di /boot viene scritta una chiave pubblica che serve per validare l'hash contenuto nella ROM Android e ad ogni accensione del dispositivo si verifica questo controllo: se l'hash è corretto la ROM viene avviata, ma se un qualsiasi componente, file o attributo viene modificato si modifica anche l'hash, di conseguenza il bootloader ne impedisce l'avvio.

Naturalmente noi VOGLIAMO modificare il contenuto di */system* (per esempio aggiungendo l'eseguibile "*su*" in */system/xbin*) o di /*boot,* nel caso di systemless rooting, pertanto si rende necessario bypassare questo controllo.

Così come per il rooting, lo sblocco del bootloader non ha procedura comune per tutti. Per gli smartphone di Google (Pixel, Nexus) e i Oneplus, ad esempio, è sufficiente abilitarne lo sblocco dalle "opzioni sviluppatore" (un menù nascosto nelle impostazioni), riavviare il telefono in modalità fastboot, collegare il dispositivo al pc e lanciare "*fastboot OEM unlock*" dalla console "fastboot" di cui abbiamo parlato poco fa.
I Samsung più recenti rincarano la dose rispetto alla procedura precedente aggiungendo un grado di sicurezza chiamato VaultKeeper. Niente di preoccupante in ogni caso, esiste la procedura per liberarsi in fretta anche di questa scocciatura.
Per esperienza personale mi sento di aggiungere anche Xiaomi alla lista di how-to-unlock, dove si rende necessario richiedere un codice di sblocco direttamente a Xiaomi, il quale viene consegnato esclusivamente dopo che siano trascorse 250 ore dalla richiesta. Questo viene giustificato specialmente per evitare che i rivenditori mettano le mani al sistema prima di rispedirli ai clienti.

Perché e come sostituire la recovery

Una volta sbarazzati dalle noie del bootloader si passa alla *recovery*, un pezzo molto importante del nostro sistema.
Le recovery stock (ovvero fornite dalla fabbrica) permettono di eseguire alcune opzioni di amministrazione, quali il reset di fabbrica, il flash di aggiornamenti e il riavvio del dispositivo.
Proprio per venire incontro alle necessità più disparate sono nate di pari passo con le ROM Android, recovery sempre più avanzate, fino ad arrivare ad oggi, dove una recovery comune permette gestire in modo completo le partizioni, formattandole singolarmente, eseguendone VERI backup nonché ripristini, eseguire il flash di pacchetti e immagini, avviare un'istanza di trasferimento dati via USB tramite MTP, il sideloading e molto altro.

La procedura è davvero elementare: una volta recuperata l'immagine della recovery compatibile con il nostro dispositivo è sufficiente avviare la modalità fastboot, collegare il dispositivo al pc ed eseguire "fastboot flash recovery $file".
Al termine basterà riavviare il telefono direttamente in recovery e godersi le nuove funzioni.

Tra le recovery più famose, quella che spicca per funzioni e quantità di dispositivi compatibili è senz'altro la TWRP.

Tipi di rooting

A partire dalle prime versioni di Android, come anticipato in precedenza, per eseguire il rooting era sufficiente inserire il binario "*su*" all'interno di /*system*/*xbin* e renderlo eseguibile.
Naturalmente era necessario anche servirsi di app di "gestione" delle concessioni, in modo da essere avvisati graficamente ogniqualvolta veniva effettuata una richiesta di esecuzione con privilegi amministrativi. Si ricordano tra le principali SuperSU e Superuser.

A partire da Android 6, in seguito ai numerosi sforzi degli sviluppatori di Android di scoraggiare la pratica del rooting, iniziò a diventare necessaria la modifica della partizione di /boot al fine di rendere il demone di "su" avviabile. Fu quindi lo stesso sviluppatore di SuperSU, Chainfire, a presentare un nuovo approccio al rooting denominato "systemless".
Il systemless rooting prevede l'inclusione del binario "su" direttamente nella partizione /boot, andando quindi ad eliminare ogni modifica nella partizione /system, dando così anche il nome alla tecnica.

I vantaggi del systemless rooting sono:

1. la semplicità di esecuzione, visto che è solamente necessario eseguire il flash dell'immagine patchata di /boot;
2. È possibile nascondere alle applicazioni installate la condizione rooted del dispositivo
3. È possibile continuare a ricevere gli update OTA e non è più necessario eseguire il flash per intero di /system, /boot e /recovery ad ogni update (che era richiesto con il vecchio metodo di rooting

Al momento della stesura di tale documento il metodo di systemless rooting più utilizzato è per mezzo di Magisk e della sua companion app Magisk Manager grazie al suo sviluppatore topjohnwu.
È sufficiente seguire la guida sul suo repository per ottenere con rapidità il controllo totale del nostro dispositivo.

Cosa fare dopo il rooting

Una volta in possesso del potere possiamo eseguire un sacco di cose interessanti:

1. Rimuovere di tutte le app di terze parti, non necessarie al sistema, altrimenti note come bloatware. Ci sono diverse app che semplificano il task e che offrono anche funzioni aggiuntive come il congelamento, la disattivazione, il backup ed il ripristino delle app. La più famosa è senz'altro Root Uninstaller;
2. Modificare i parametri del kernel o installarne uno custom in base alle proprie esigenze
3. Modificare l'interfaccia utente
4. Recuperare file cancellati
5. Cambiare del DNS, che in alcuni dispositivi è possibile effettuare anche senza rooting;

6. Recuperare le password delle reti Wi-Fi salvate nel dispositivo
7. Installare framework di terze parti o avviare intere distribuzioni
8. Utilizzare patcher per applicazioni come Lucky Patcher
9. Installare custom ROM o custom recovery tramite Flashify senza la necessità di una custom recovery
10. L'uso dei moduli Magisk nei systemless root;

Proprio quest'ultimo punto apre l'ultima discussione di questo workshop:
Il rooting causa il blocco di alcune app che esigono che il dispositivo sul quale sono installate sia non rooted.
Questo è un problema perché ci costringe a scegliere tra l'uso di una app o l'amministrazione completa del nostro device.
In nostro aiuto, un altra volta, viene Magisk con i suoi moduli, nello specifico il modulo Magisk Hide. Esso permette di nascondere la condizione di rooted alle applicazioni e di non costringerci alla noiosa decisione tra l'una e l'altro.

Oltre a detto modulo, ne esiste una moltitudine per le più disparate esigenze. Si ricorda, tra i principalmente utilizzati: Viper4Android (per la gestione del comparto audio), Youtube Vanced (elimina ADs, permette la riproduzione in background e PiP), CloudflareDNS, App Systemizer (per rendere app in userspace di sistema), Call Recorder… la lista è infinita e chi vuole può svilupparsi il suo modulo siccome Magisk è sotto licenza GPL.

Di recente John Wu, lo sviluppatore di questo indispensabile tool, ha reso noto di essere divenuto membro del team di sicurezza di Android in Google.
In conseguenza di ciò, dopo i timori di alcuni, si è tirato un sospiro di sollievo quando ha fatto sapere che non terminerà lo sviluppo di Magisk ma dovrà abbandonare il supporto a Magisk Hide.
Ciò comunque non significa che sarà abbandonato, siccome stiamo sempre parlando di un progetto libero ed open-source e nessuno vieta al resto della comunità di mantenerne vivo lo sviluppo.

Viva il software libero!!