Guida al riconoscimento dell'emulatore nell'analisi di un malware in Android

Author : Gianluigi Spagnuolo

Date : 3 settembre 2018

Per comprendere il comportamento di un malware si eseguono un'analisi statica e una dinamica; gli strumenti che si occupano di questo, di solito, vengono utilizzati in un ambiente emulato e virtualizzato.

Avviene però, a volte, che un malware cerchi di eludere queste analisi individuando l'ambiente di esecuzione e non permettendo di fatto a questi tool di identificarne il vero comportamento.

I malware più avanzati, dunque, implementano diversi controlli per individuare un'esecuzione in un ambiente emulato; in questo modo, il software può tentare di variare il proprio comportamento, nascondendo le azioni malevole e continuando così indisturbato il proprio lavoro.

In pratica, quando uno di questi controlli individua l'esecuzione in un possibile emulatore, il malware modifica il suo comportamento, risultando innocuo oppure semplicemente terminando l'esecuzione.

Nel mondo Android, dove l'analisi in un emulatore è quasi d'obbligo, questo comportamento è più accentuato.

L'emulatore di Android è basato sull'emulatore open source QEMU (Quick Emulator). QEMU emula un'architettura traducendo le istruzioni della CPU emulata in istruzioni della CPU reale, ovvero del processore che ospita l'emulatore. Essendo l'emulatore di Android un fork di QEMU, ha tutte le sue caratteristiche ma condivide anche gran parte dei metodi che i malware utilizzano per identificarlo.

Sul perché è importante ovvero perché è preferibile fare l'analisi con un emulatore

Per analizzare un malware, si cerca di comprenderne il comportamento in modo da poter

valutare i rischi concreti derivanti dalla sua esecuzione; per osservare l'impatto, sull'host e sulla rete, si effettuano diverse analisi sul codice incriminato. È preferibile svolgere l'analisi dinamica in un ambiente controllato e monitorato, dunque in un emulatore.

Per eseguire un'analisi dinamica, visto il numero e la complessità dei malware, si usano degli appositi strumenti che, in gran parte, sono basati sulla virtualizzazione o sull'emulazione di uno smartphone.

Molti malware includono delle tecniche di individuazione dell'emulatore, in modo da rendere difficoltosa, quando non impossibile, l'analisi attraverso tool e tecniche di *reverse engineering* specifiche.

Da una parte è conveniente svolgere le analisi in un ambiente virtualizzato, dall'altra parte, per un malware, non essendoci usi significativi dell'emulazione e della virtualizzazione su dispositivi mobili, è ragionevole sfruttare le differenze tra ambiente emulato e ambiente reale per modificare il proprio comportamento.

Sul riconoscimento ovvero cosa fanno i malware per individuare l'esecuzione in un emulatore

Sfruttando le differenze tra ambiente virtuale e reale, un malware può identificare la propria esecuzione in un emulatore e modificare di conseguenza il proprio comportamento. Differenze che riguardano ad esempio elementi hardware non perfettamente implementati, componenti hardware e software parzialmente o completamente non virtualizzati (presenza e gestione dei vari sensori), durata dell'esecuzione, informazioni sul dispositivo (codice IMEI, codice IMSI, versione della build del SO), etc.

Possiamo dividere le informazioni usate per individuare un emulatore essenzialmente in tre categorie:

- **informazioni statiche**: informazioni basate su valori inizializzati dall'emulatore in fase di avvio. Il controllo, e quindi l'individuazione, da parte di un malware può essere fatto sulla presenza e sul contenuto di tali informazioni, come ad esempio l'identificativo univoco del dispositivo, la routing table, la versione della build, ecc.
- **informazioni dinamiche**: informazioni basate sulla presenza e sul comportamento di alcuni componenti hardware, come ad esempio i sensori, presenti su un dispositivo fisico e per loro natura difficili da emulare. I dispositivi mobili hanno dei sensori che interagiscono con l'ambiente circostante e ne misurano alcuni valori; questo comportamento è difficile da emulare in maniera realistica. L'esistenza stessa di alcuni sensori, nonché il loro livello di simulazione, è la chiave dei metodi di individuazione di uno smartphone o di un emulatore in un sistema classico.
- **informazioni relative all'ambiente virtuale**: informazioni basate sulla macchina virtuale e sulla sua, eventuale, incompleta implementazione di un'architettura.

Ad esempio, alcune tecniche di individuazione appartenenti alle categorie precedenti sono le

seguenti:

- Si possono ottenere molte informazioni sul dispositivo attraverso le **API di Android**. In particolare, quelle che ci interessano sono le operazioni di telefonia e l'interrogazione diretta dell'hardware. Si può ricavare via API il *Device ID*; fino alla versione 26 delle API, *getDeviceId* di *TelephonyManager* restituiva il codice IMEI per i telefoni GSM e il codice MEID o ESN per i CDMA, ora invece occorre chiamare *getImei()* e *getMeid()* per ottenere rispettivamente il codice IMEI e il MEID. Il codice IMEI e il codice MEID sono numeri unici che identificano un telefono nella rete GSM o CDMA, tali codici possono essere usati da un malware per identificare un emulatore, che non essendo eseguito in queste reti non possiede un vero codice IMEI/MEID.
  Un altro elemento tipico dei dispositivi mobili è l'IMSI (International Mobile Subscriber Identity) che identifica una SIM card, recuperabile in Android con il metodo *getSubscriberId()* di *TelephonyManager*.
  Infine si possono ispezionare le proprietà del sistema che ci dicono, ad esempio, il modello, il prodotto, l'hardware sottostante e informazioni simili, molto utili per individuare dove un'applicazione viene eseguita.
- Si può individuare se la **rete** è emulata: ad esempio analizzando gli indirizzi, la tabella di routing, i server DNS, il gateway.
- Può essere usata qualsiasi tecnica utilizzata per individuare **QEMU**, che come detto sta alla base dell'emulatore di Android.
- Possono essere sfruttati i differenti **componenti** hardware e software presenti nel dispositivo reale e nell'emulatore.

Inoltre, possono essere analizzate anche le differenze dovute alla natura dei due sistemi (emulatore e dispositivo fisico), che, per quanto simili, nascono per esigenze diverse: ad esempio, si può vedere se ci sono, e quanti sono, i contatti, si può ispezionare il log delle chiamate, il numero di applicazioni installate oppure il numero e la lunghezza degli SMS.

Sul come si può aggirare ovvero come si può eludere il controllo in fase di analisi

Abbiamo detto che per poter analizzare un malware e scoprire eventuali tecniche anti-emulazione, abbiamo bisogno di eseguire un'analisi dinamica. Per carpirne il comportamento abbiamo attualmente poche strade:

1. Possiamo scrivere e applicare una **patch** che riguarda i controlli eseguiti dal malware, ad esempio, sovrascrivendo le funzioni non desiderate.
2. Oppure, possiamo usare uno strumento come **Frida** (https://www.frida.re) per intercettare i metodi del malware che si occupano del controllo e immunizzarne il comportamento.

Tra le possibili soluzioni, più onerose delle precedenti, e sicuramente più drastiche, c'è la possibilità di:

- modificare l'emulatore in modo da renderlo più affine ad uno smartphone
- rendere più verosimili gli eventi generati dai sensori simulati
- migliorare il supporto ARM in QEMU

Conclusioni

Gran parte degli attuali strumenti, che si occupano dell'analisi dei malware, non prevedono la personalizzazione dei propri parametri. In questo modo, rendono immediato il riconoscimento dell'ambiente di esecuzione da parte di un malware. Come abbiamo visto, però, ci sono delle soluzioni per rendere un'analisi, anche se non perfetta, più precisa e meno dispendiosa.

A cura di: **Gianluigi Spagnuolo**