

## Divya Mobil Uygulamasına Güvenlik Testleri Dersleri

Bu makalede Divya (Damn Insecure and Vulnerable Application) adlı kasıtlı zafiyet içeren bir mobil uygulamaya güvenlik testleri uygulanacaktır.

### Gereksinimler

Santoku Mobile Security and Forensic Linux VM	// Attacker Desktop VM
Samsung Galaxy S4 VM	// Victim Mobile VM
DIVA Vulnerable Mobile Application	// Victim Mobile App

Santoku adlı mobil güvenlik ve adli analiz araçlarının bulunduğu sanal makinadan hedef bir mobil cihazdaki zafiyete sahip mobil uygulamaya güvenlik testleri uygulanacaktır.

Öncelikle saldırıları düzenleyeceğimiz Santoku sanal makinasını başlatalım.



Santoku VM'de kullanacağımız araçlar şu şekildedir:

### Santoku VM'de Kullanılacak Araçlar

\*Dex2jar ve JD-GUI

\*ApkTool

\*Drozer

\*Android Kullanışlı Araçlar (adb, sqlite3, aapt vs.)

// Reverse Engineering Tools

// Reverse Engineering Tools

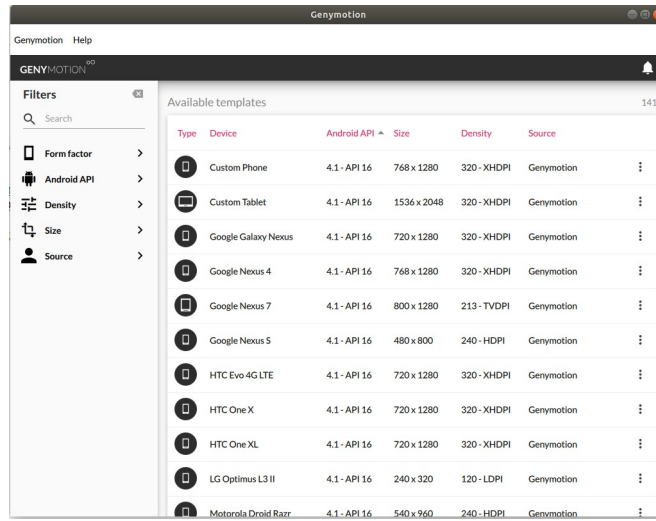
// Dynamic Analysis Tool

// Android Utilities

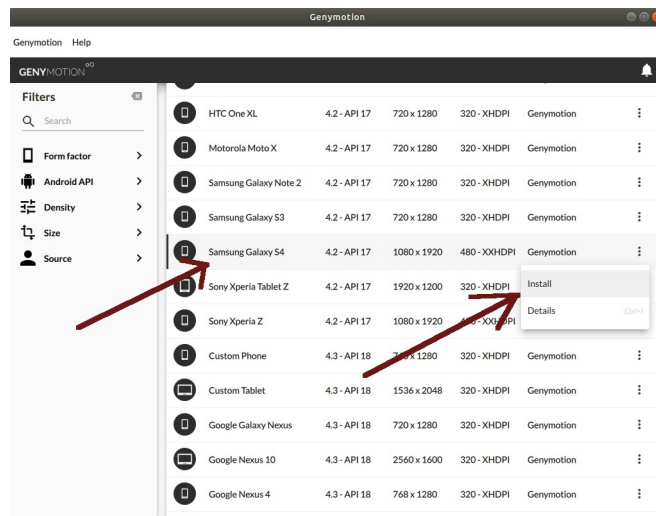
Ardından genymotion sanallaştırma aracı ile hedef bir mobil sanal sistem oluşturalım. Bunun için hedef mobil sanal sistem olarak Samsung Galaxy S4 belirleyelim.

Ubuntu 18.4 LTS Terminal:

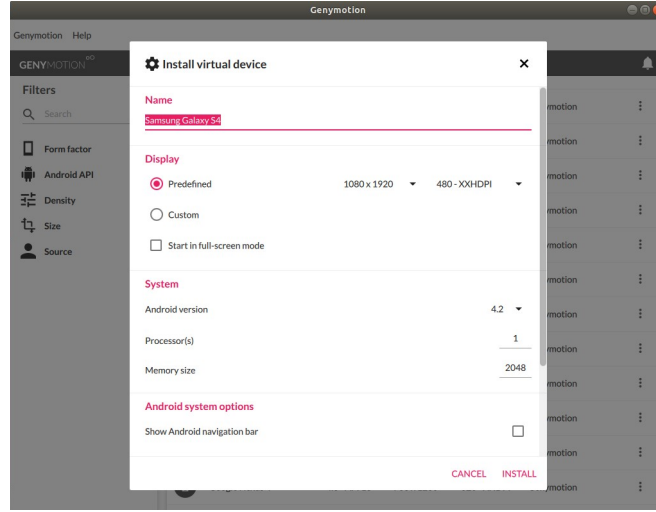
> genymotion



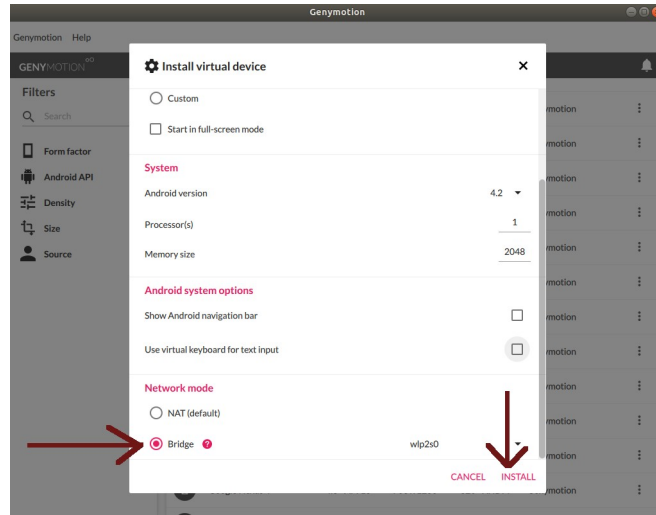
( Genymotion Sanallaştırma Aracı Başlar )



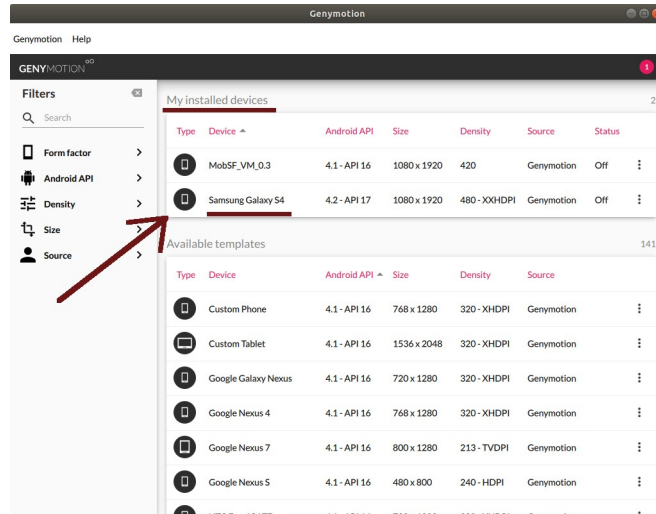
( Samsung Galaxy S4 Mobil Sanal Sistemini Oluştur Denir )



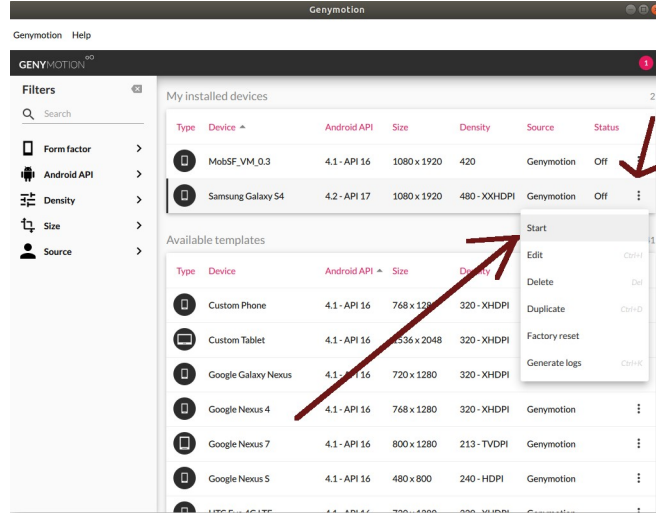
( Samsung Galaxy S4 Mobil Sanal Sistemi İsmi Düzenlenir )



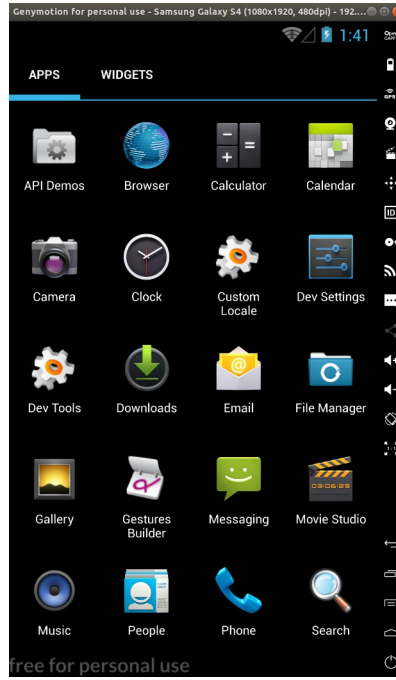
( Samsung Galaxy S4 Mobil Sanal Sistemi Network Ayarı Düzenlenir )



( Genmotion Cihazlarım Sekmesine Samsung Galaxy S4 Mobil Sanal Sistemi Yerleşir )



( Samsung Galaxy S4 Mobil Sanal Sistemi Başlatılır )



( Samsung Galaxy S4 Mobil Sistemi Başlar )

Şimdi saldırgan makina ve hedef makina hazır olduğuna göre hedef makinaya zafiyete sahip DIVA adlı mobil uygulamayı yükleyelim.

Ubuntu 18.04 LTS Terminal:

(

not:

Ubuntu 18.04 LTS ana makinaya adb tool'u kurulumu için bkz. Yaz Tatili 2014 / Android Mobil Belgeler / Adb Kurulumu.txt

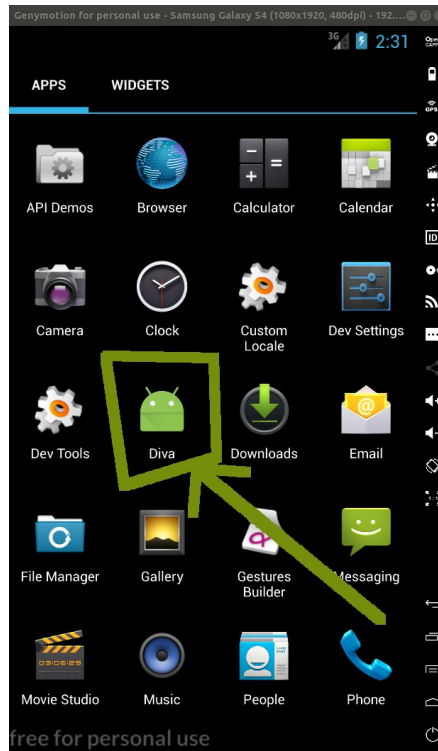
)

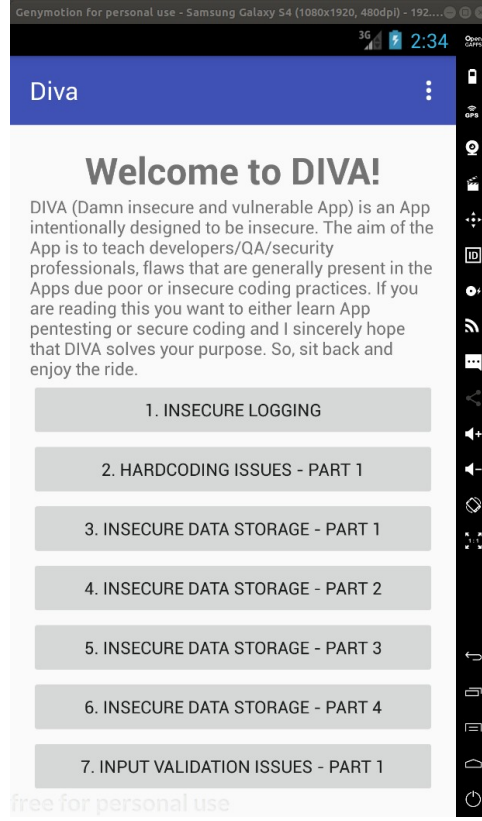
- > adb devices
- > adb install "/home/hefese/Desktop/diva-beta.apk"

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~  
File Edit View Search Terminal Help  
hefese@TUBITAK-HASANFSIMSEK3:~$ adb install "/home/hefese/Desktop/diva-beta.apk"  
Performing Push Install  
/home/hefese/Desktop/diva-beta.apk: 1 ..d. 38.4 MB/s (1502294 bytes in 0.037s)  
  pkg: /data/local/tmp/diva-beta.apk  
Success  
hefese@TUBITAK-HASANFSIMSEK3:~$
```

Görüldüğü üzere adb tool'u ile Diva adlı vulnerable apk dosyası yüklenir ve uygulama sanal mobil sisteme yerleşir.





Şimdi testlere geçelim.

## **Başlangıç: Android Uygulamalarda Tersine Mühendislik ile Okunabilir Java Kaynak Kodları Elde Etme (Dex2Jar, JD-GUI, ApkTool)**

Başlangıç olarak santoku saldırgan sistemimizden hedef mobil sistemdeki Diva adlı zafiyete sahip mobil uygulamanın tersine mühendislik ile okunabilir kaynak kodlarını (.java kaynak kodlarını ve sonra .smali assembly kaynak kodlarını) elde edelim. Çünkü bu kaynak kodlar uygulanacak güvenlik testlerinde üzerlerinden geçilerek tespit / teyit / ... gibi işlemlerde yardımcı olacaktır.

### **[-] Uyarı:**

Santoku sanal makinası genymotion mobil sanal makinasını göremediğinden / erişemediğinden Santoku makinasında yüklü adb aracıyla genymotion mobil sanal makinasındaki .apk çekilememektedir. Bu nedenle önce ana makinadan genymotion mobil sanal makinasındaki apk dosyası çekilecektir ve sonra Santoku sanal makinasına ana makinadaki apk dosyası scp ile kopyalanacaktır.

Ana makinadan hedef mobil sistemdeki zafiyete sahip Diva isimli uygulamayı çekebilmek için dosya yolunu tespit edelim.

Ubuntu 18.04 LTS Terminal:

```
> adb shell
root@android:/# cd /data/app/
root@android:/data/app# ls | grep "diva"
```

Çıktı:

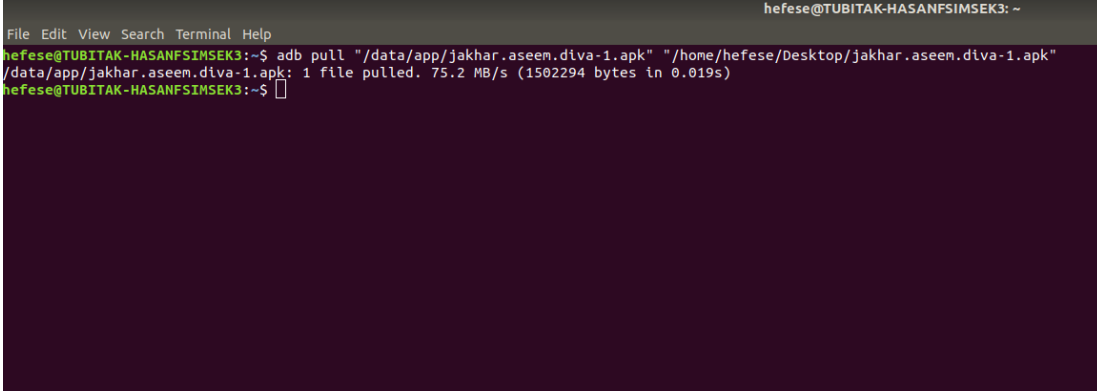
```
jakhar.aseem.diva-1.apk  
root@android:/data/app# exit
```

Hedef sanal mobil sistemdeki zafiyete sahip Diva isimli uygulamanın dosya yolu tespiti ( /data/app/jakhar.aseem.diva-1.apk ) sonrası ana makinadan adb pull komutu ile Diva apk uygulamasını çekelim.

Ubuntu 18.04 LTS Terminal:

```
> adb pull "/data/app/jakhar.aseem.diva-1.apk" "/home/hefese/Desktop/jakhar.aseem.diva-1.apk"
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~  
File Edit View Search Terminal Help  
hefese@TUBITAK-HASANFSIMSEK3:~$ adb pull "/data/app/jakhar.aseem.diva-1.apk" "/home/hefese/Desktop/jakhar.aseem.diva-1.apk"  
/data/app/jakhar.aseem.diva-1.apk: 1 file pulled. 75.2 MB/s (1502294 bytes in 0.019s)  
hefese@TUBITAK-HASANFSIMSEK3:~$
```

Sonra ana makinadaki Diva apk dosyasını santoku sanal makinasına scp ile kopyalayalım.

Santoku Terminal:

( Not: scp için santoku sanal sisteminde ssh server kurulu olmalıdır )

```
> sudo su # Şifre : santoku  
> apt-get install openssh-server # ssh server kurulur
```

Ubuntu 18.04 LTS Terminal:

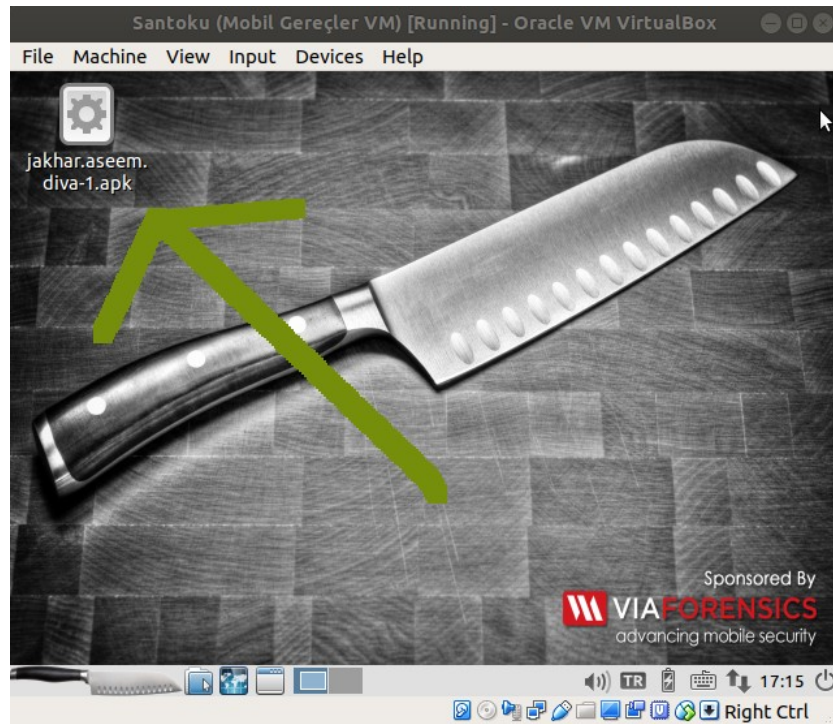
```
> scp "/home/hefese/Desktop/jakhar.aseem.diva-1.apk"  
santoku@192.168.0.15:~/home/santoku/Desktop/jakhar.aseem.diva-1.apk"
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~  
File Edit View Search Terminal Help  
hefese@TUBITAK-HASANFSIMSEK3:~$ scp "/home/hefese/Desktop/jakhar.aseem.diva-1.apk" santoku@192.168.0.15:"/home/santoku/Desktop/jakhar.aseem.diva-1.apk"  
santoku@192.168.0.15's password:  
jakhar.aseem.diva-1.apk 100% 1467KB 69.0MB/s 00:00  
hefese@TUBITAK-HASANFSIMSEK3:~$
```

Böylece Santoku sistemi hedef mobil sistemdeki apk dosyasını alır.



Şimdi Santoku sanal sistemindeki tersine mühendislik araçları ile Diva isimli mobil uygulama apk dosyasından okunabilir kaynak kodlar elde edelim.

### i) Dex2Jar ve JD-GUI

Mobil uygulama dosyasındaki (.apk'daki) binary kodlardan okunabilir .java kaynak kodları elde etmek için dex2jar dönüştürme aracı kullanılabilir. JD-GUI aracı ile de okunabilir .java kaynak kodları görüntülenebilir.

Bilgi:

Android sistemlerde mobil uygulama dosyaları (.java dosyaları) derlendiklerinde .dex (dalvik executable) çalıştırılabilir formatında olurlar. Bu .dex (dalvik executable) çalıştırılabilir formatındaki dosyalar tek bir dosya halinde sıkıştırıldıklarından .apk halinde olurlar.

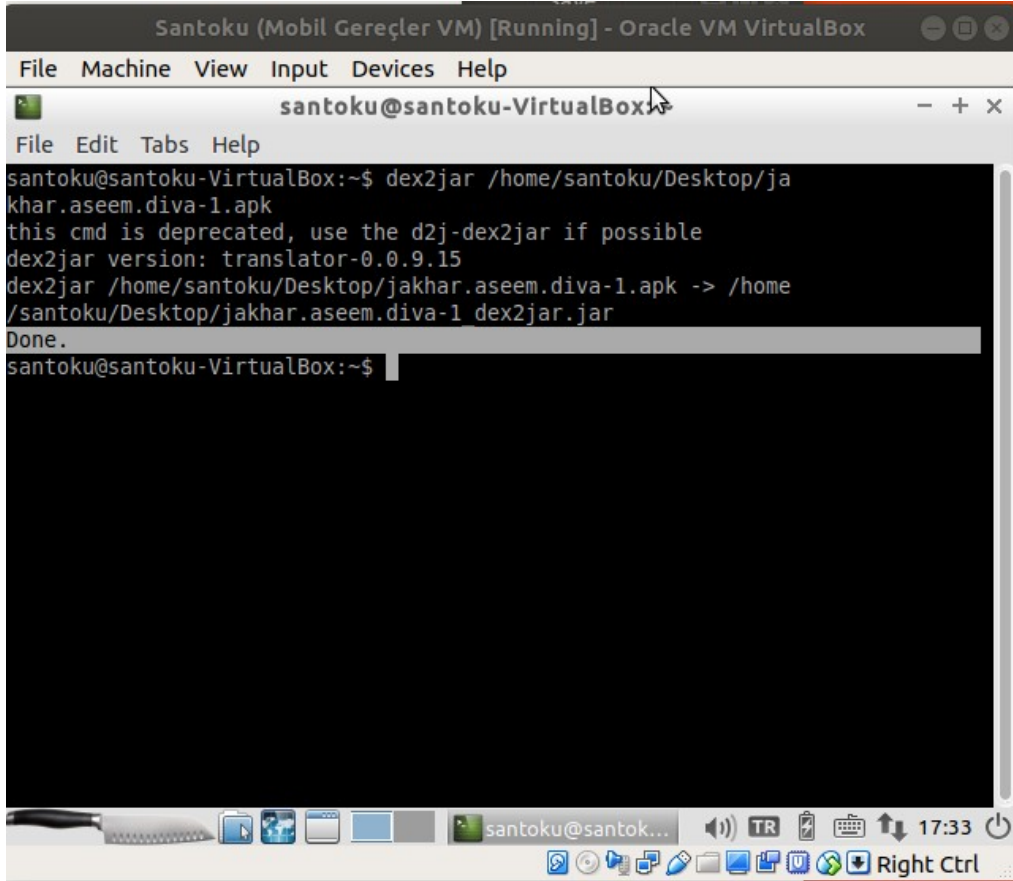


Dex2Jar ile mobil uygulama dosyasını (dex dosyalarını) okunabilir .java kaynak kodları haline dönüştürelim.

Santoku Terminal:

> dex2jar /home/santoku/Desktop/jakhar.aseem.diva-1.apk

Çıktı:



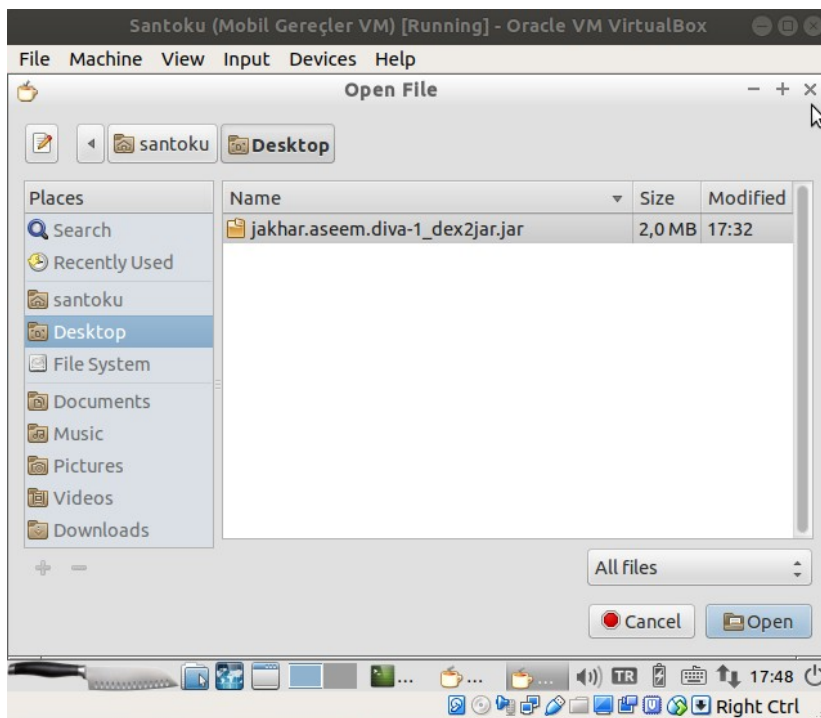
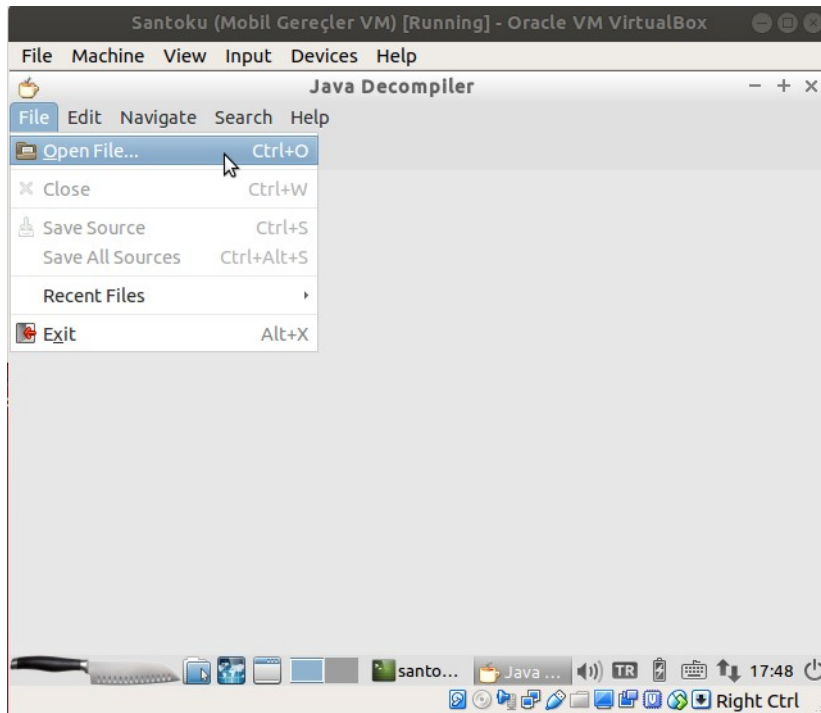
```
Santoku (Mobil Gereçler VM) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
santoku@santoku-VirtualBox
File Edit Tabs Help
santoku@santoku-VirtualBox:~$ dex2jar /home/santoku/Desktop/ja
khar.aseem.diva-1.apk
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar /home/santoku/Desktop/jakhar.aseem.diva-1.apk -> /home
/santoku/Desktop/jakhar.aseem.diva-1 dex2jar.jar
Done.
santoku@santoku-VirtualBox:~$
```

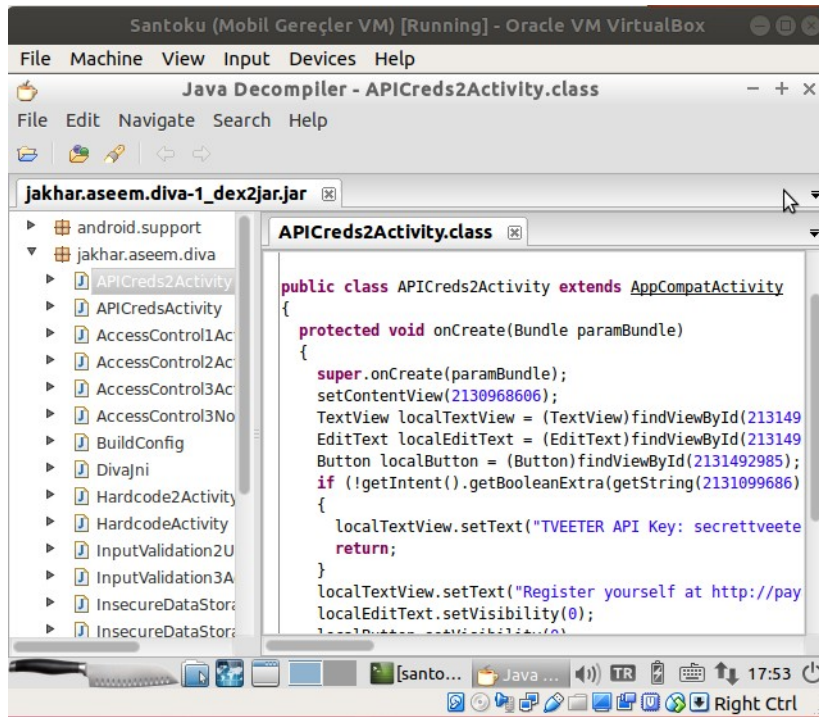
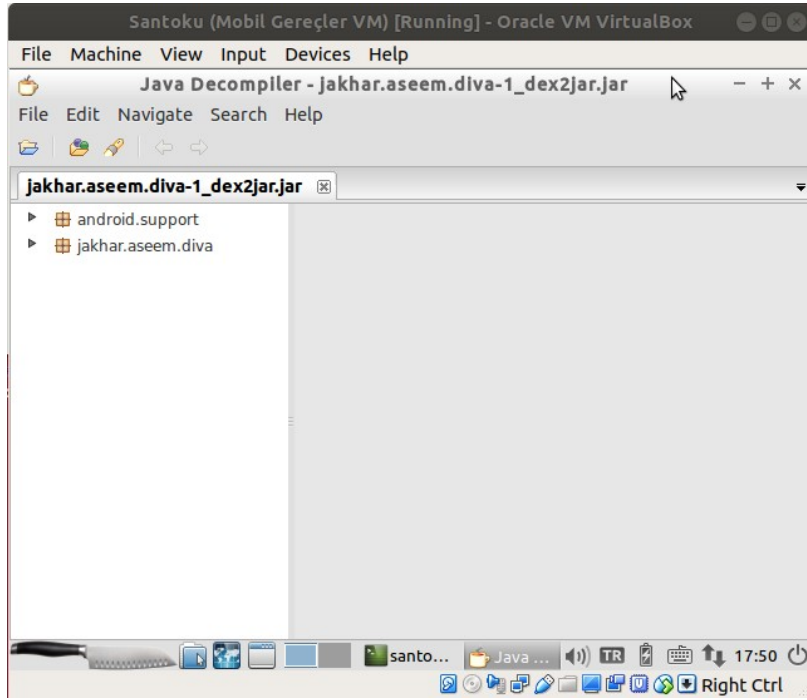
Dönüşen .java okunabilir kaynak kodlarını görüntüleyelim.

Santoku Terminal:

> jd-gui

Çıktı:





## ii) ApkTool

Mobil uygulama dosyasındaki (.apk'daki) binary kodlardan okunabilir .smali assembly kodları elde etmek için ve okunabilir AndroidManifest.xml dosyasını elde etmek için apktool aracı kullanılabilir.

## [!] Bilgi: Smali Assembly Kodları Nedir

Smali ve Baksmali birer assembly / disassembly araçlarıdır. Android sistemlerdeki Dalvik Virtual Machine tarafından üretilen çalıştırılabilir android uygulama dosyalarını (.dex dosyalarını) assembly formatına çevirmeye veya geri dex formatına çevirmeye yarar. Bu araçlar dex formattaki tüm işlevleri (annotation'ları, debug bilgileri, satır bilgileri,...) destekler niteliktedir.

Örneğin; android java dosyasındaki;

```
int x = 42
```

ifadesi derlendiğinde ve çalıştırılabilir dex formatına dönüştüğünde şu binary koda (daha öz olarka hex koda) dönüşür:

```
13 00 2A 00
```

Eğer bu binary kodu baksmali aracı ile disassembly yaparsak,

```
const/16 v0, 42
```

okunabilir assembly kodu elde edilir. Yani dex binary kodundan baksmali aracı ile bir smali okunabilir assembly kodu elde edilir. Assembly koda smali assembly denmesinin nedeni smali aracı teknolojisiyle dex binary kodlardan assembly kodların elde edilebiliyor olmasındandır veya elde edilen assembly kodlardan dex binary koda dönüşümün yapılabilir olmasındandır.

Şimdi apktool aracı ile okunabilir smali assembly kodlarını ve AndroidManifest.xml dosyasını elde edelim.

## [!] Uyarı:

Santoku sanal sistemimde yüklü apktool aracı eski teknoloji içeren Diva adlı mobil uygulama dosyasından smali assembly dosyalarını ve manifest dosyasını çıkaramamıştır. Hata olarak major/minor versiyon uyumsuzluğu hatası vermiştir. Bu nedenle yararlanılan makalede apktool'unun eski sürümü 2.0.3 üzerinden gidilerek Diva uygulama dosyası üzerinde test yapıldığından internetten apktool'unun 2.0.3 eski sürümü indirilmiştir,

<https://bitbucket.org/iBotPeaches/apktool/downloads/>

ve java -jar ile apktool 2.0.3 sürümü diva mobil uygulama dosyası üzerinde başarılı bir şekilde çalıştırılmıştır.

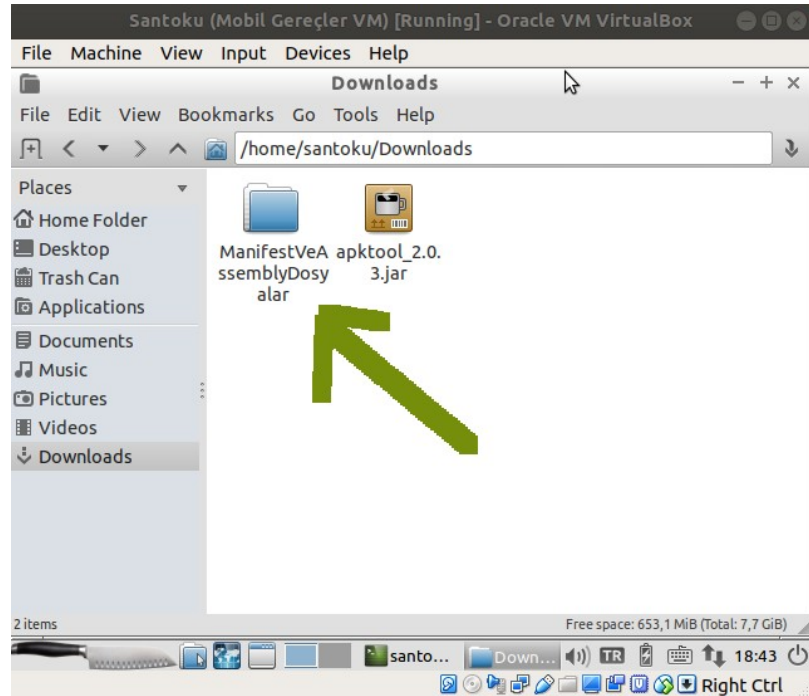
Not: apktool aracının yeni sürümlerinde parametre syntax'ı değişmiştir.

Santoku Linux Terminal:

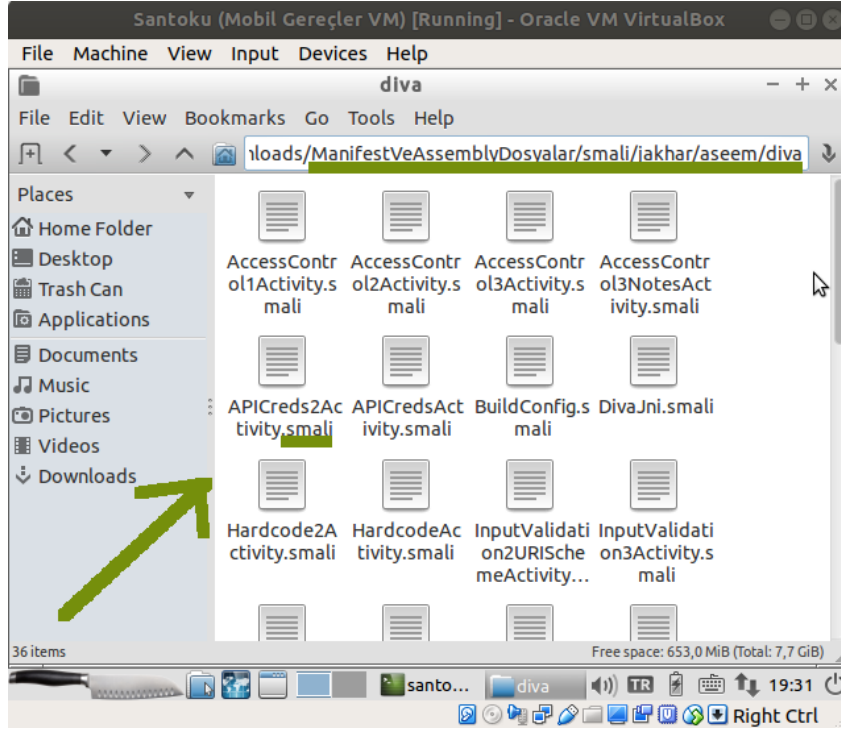
```
> cd Downloads
> chmod a+x apktool_2.0.3.jar
> java -jar apktool_2.0.3.jar d "/home/santoku/Desktop/jakhar.aseem.diva-1.apk"
-o ManifestVeAssemblyDosyalar/ // (*) d : decode
```

Çıktı:

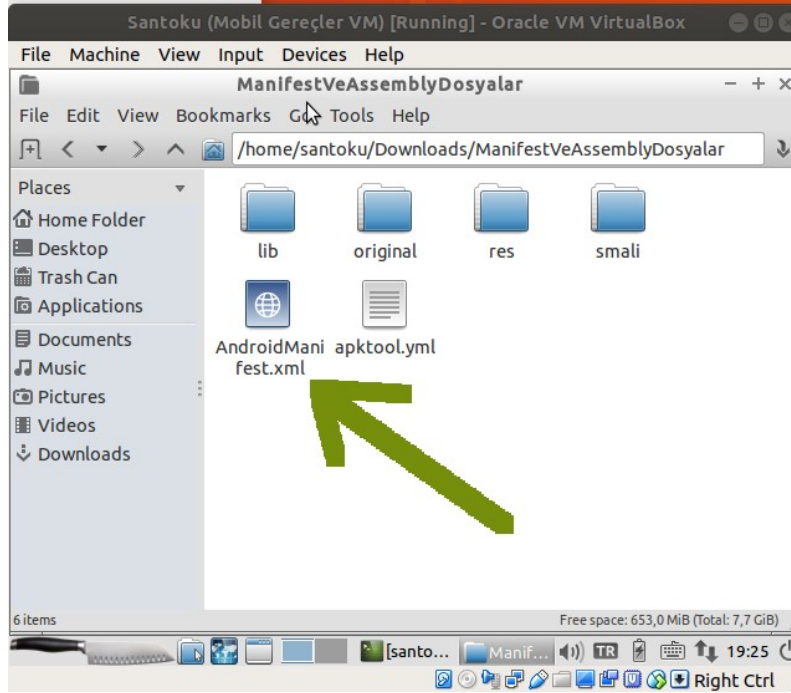
```
Santoku (Mobil Gereçler VM) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
santoku@santoku-VirtualBox: ~/Downloads
File Edit Tabs Help
santoku@santoku-VirtualBox:~/Downloads$ java -jar apktool_2.0.3.jar d /home/s
antoku/Desktop/jakhar.aseem.diva-1.apk -o ManifestVeAssemblyDosyalar
I: Using Apktool 2.0.3 on jakhar.aseem.diva-1.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/santoku/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values ** XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
santoku@santoku-VirtualBox:~/Downloads$
```



apktool aracının mobil uygulama dosyasından (.apk'dan) dönüştürdüğü smali assembly dosyaları çıktı klasöründeki smali/ dizini altında yer alır.



apktool aracının mobil uygulama dosyasından (.apk'dan) dönüştürdüğü ve elde ettiği AndroidManifest.xml dosyası çıktı klasöründeki kök dizinde yer alır.



```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="jakhar.a
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<application android:allowBackup="true" android:debuggable="true" android:icon="@mipi
<activity android:label="@string/app_name" android:name="jakhar.aseem.diva.MainActi
  <intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
  </intent-filter>
</activity>
<activity android:label="@string/d1" android:name="jakhar.aseem.diva.LogActivity"/>
<activity android:label="@string/d2" android:name="jakhar.aseem.diva.HardcodeActivity
<activity android:label="@string/d3" android:name="jakhar.aseem.diva.InsecureDataStor
<activity android:label="@string/d4" android:name="jakhar.aseem.diva.InsecureDataStor
<activity android:label="@string/d5" android:name="jakhar.aseem.diva.InsecureDataStor
<activity android:label="@string/d6" android:name="jakhar.aseem.diva.InsecureDataStor
<activity android:label="@string/d7" android:name="jakhar.aseem.diva.SQLInjectionActiv
<activity android:label="@string/d8" android:name="jakhar.aseem.diva.InputValidation2
<activity android:label="@string/d9" android:name="jakhar.aseem.diva.AccessControl1A
<activity android:label="@string/apic_label" android:name="jakhar.aseem.diva.APICreds
  <intent-filter>
```

Şimdi zafiyetler içeren Diva mobil uygulamasındaki zafiyetleri test edelim ve sömürme adımlarını görelim.

#### Kaynaklar

- <http://www.payatu.com/damn-insecure-and-vulnerable-app/>
- <https://resources.infosecinstitute.com/cracking-damn-insecure-and-vulnerable-apps-diva-part-1/#gref>
- <https://resources.infosecinstitute.com/android-penetration-tools-walkthrough-series-drozer/#gref>
- <https://gurelahmet.com/mobil-android-s%C4%B1zma-testine-giri%C5%9F/>
- <https://stackoverflow.com/questions/7750448/what-are-dex-files-in-android>
- <https://stackoverflow.com/questions/30054156/apktools-apk-studio-could-not-decode-arsc-file>
- <https://bitbucket.org/iBotPeaches/apktool/downloads/>
- <https://stackoverflow.com/questions/4191762/how-to-view-androidmanifest-xml-from-apk-file>
- <https://stackoverflow.com/questions/30054156/apktools-apk-studio-could-not-decode-arsc-file>
- <https://tools.kali.org/reverse-engineering/smali>
- <https://stackoverflow.com/questions/30837450/what-is-smali-code-android>

Paketleme İçin Gözden Geçirilecekler / Mobil Hk / İnternette Edinilmiş Belgeler / Adb Tool Nedir ve Kullanımı.docs#Uygulama 1