

FURKAN ENES POLATOĐLU

# MOBİL SİSTEMLERİN GÜVENLİĐİ



# Mobil Sistemlerin Güvenliđi

19/02/2021

Furkan Enes Polatođlu  
furkanenes1160@icloud.com

## İçindekiler

Android (İřletim Sistemi) .....	4
Android Kullanım Alanları .....	4
İstatistikler .....	4
Android Güvenlik Modeli .....	4
Eleřtiriler .....	5
Android Mimarisi .....	5
1. Linux Çekirdek (Linux Kernel) .....	5
2. Kütüphaneler (Libraries) .....	5
3. Android Çalışma Zamanı (Android Runtime) .....	5
3.1. Çekirdek Kütüphaneleri (Core Libraries) .....	6
3.2. Dalvik Sanal Makinesi (DVM – Dalvik Virtual Machine) .....	6
DEX Dosyaları .....	6
4. Uygulama Çatısı (Application Framework) .....	6
5. Uygulamalar (Applications) .....	6
Android Uygulamaları .....	7
Smali Code .....	7
DEX ve Smali kod dosyaları arasındaki fark .....	7
APK .....	7
JAR .....	7
Android Paket İçeriđi .....	8
Kaynak Kod Dönüřümü .....	12
Decompile (Kaynak Koda Dönüřtürme) .....	12
Dex2jar .....	12
JD-GUI .....	12
Disassembling (Makine Dilini Montaj Diline Çevirmek) .....	13

<b>Kod Karmaşıklştırma (Obfuscation)</b> .....	14
<b>OWASP Mobile Top 10</b> .....	14
<b>M1 – Improper Platform Usage (Hatalı Platform Kullanımı)</b> .....	15
<b>M2 – Insecure Data Storage (Güvensiz Veri Saklama)</b> .....	15
<b>M3 – Insecure Communication (Güvenli Olmayan İletişim)</b> .....	15
<b>M4 – Insecure Authentication (Güvensiz Doğrulama)</b> .....	15
<b>M5 – Insufficient Cryptography (Yetersiz Şifreleme)</b> .....	15
<b>M6 – Insecure Authorization (Güvensiz Yetki)</b> .....	15
<b>M7 – Client Code Quality (İstemci Kod Kalite Sorunları)</b> .....	15
<b>M8 – Code Tampering (Kod Kurcalama)</b> .....	15
<b>M9 – Reverse Engineering (Tersine Mühendislik)</b> .....	16
<b>M10 – Extraneous Functionality (Gereksiz İşlevsellik)</b> .....	16
<b>Mobil Sızma Testi Araçları</b> .....	17
<b>ADB (Android Debug Bridge)</b> .....	17
<b>ADB Komutları</b> .....	17
<b>Burp Suite</b> .....	20
<b>SQLite Browser ve SQLite3</b> .....	21
<b>AndroBugs Framework</b> .....	22
<b>MobSF (Mobile Security Framerwork)</b> .....	23
<b>Drozer</b> .....	24
<b>Drozer Kullanarak Android Uygulama Güvenliği Nasıl Test Edilir?</b> .....	25
<b>OWASP ZAP</b> .....	29
<b>Mobil Sızma Testi Uygulaması</b> .....	30
<b>DIVA (Damn Insecure and Vulnerable App)</b> .....	30
<b>1. Diva Insecure Logging</b> .....	31
<b>2. Hardcoding Issues - Part 1</b> .....	33
<b>3. Insecure Data Storage – Part 1</b> .....	35
<b>4. Insecure Data Storage – Part 2</b> .....	36
<b>5. Insecure Data Storage – Part 3</b> .....	37
<b>6. Insecure Data Storage – Part 4</b> .....	39
<b>8. Input Validation Issues - Part 2</b> .....	40
<b>11. Access Control Issues – Part 3</b> .....	41
<b>13. Input Validation Issues – Part 3</b> .....	42

## Android (İşletim Sistemi)

- Android; Google ve Open Handset Alliance tarafından mobil aygıtlar için geliştirilmekte olan Linux tabanlı özgür ve ücretsiz bir işletim sistemidir.
- İşletim sisteminin açık kaynak kodlu olduğu söylene de kodlarının az ama çok önemli bir kısmı Google tarafından gizli tutulmaktadır.
- Android işletim sisteminin desteklenen uygulama uzantısı “.apk”dır.
- Android, derlenmiş Java kodunu çalıştırmak için Android Runtime (ART) kullanır.

## Android Kullanım Alanları

- Cep telefonları, tabletler, akıllı saatler...
- Arabalar, akıllı ev sistemleri
- Mobil bankacılık
- Internet of Things (IoT) vs.

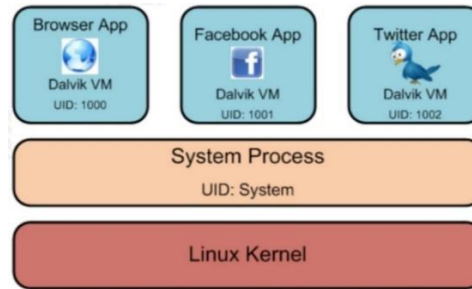
## İstatistikler

- Akademik ve iş için kullanılan akıllı telefon ve tablet oranı: **%76**
- 2015-2016 yıllarında akıllı telefonları hedef alan zararlı yazılım artış oranı: **%250**
- Platformlara göre zararlı yazılımlar: Android (**%73**), IOS (**%32**)

## Android Güvenlik Modeli

- Linux güvenlik modeli esas alınmıştır (UID/GUID)
- Uygulama bazlı izinler kullanılmaktadır.
- Uygulama izinleri, “AndroidManifest.xml” dosyasında tanımlanmaktadır.
- Uygulama kurulumu için uygulamanın sertifika ile imzalanmış olması gerekmektedir.
- Rootlanmamış bir cihaz için root erişimi mümkün değildir. “su” uygulaması sistemde bulunmaz.

- UID/GUID: Linux çekirdeği, kullanıcıları sadece basit sayılar olarak algılar. Her kullanıcı için tam sayılardan oluşan benzersiz bir tanımlama yapılmıştır, çünkü bir bilgisayar için sayılarla uğraşmak harflerden oluşan isimler ile uğraşmaktan daha kolaydır. Bunlara kullanıcı kimliği (uid) ve grup kimliği (guid) denir. Bunlardan ilerde bahsedeceğiz.



## Eleştiriler

Android, Google tarafından tamamen özgür yazılım olarak yayınlanmadığı için eleştirilmektedir. SDK'nın bazı bölümleri hala patentli olup açık kaynak kodlu değildir. Bunun Google tarafından bilinçli olarak yapıldığına inanılmaktadır.

## Android Mimarisi



### **1. Linux Çekirdek (Linux Kernel)**

Android mimarisindeki en alt katmandır. Bu katmanda donanımsal bilgiler ve uygulamaların çalışabilmesi için gerekli sürücüler yer alır (klavye sürücüler, ses sürücüler, Wi-fi sürücüler, kamera ve görüntü sürücüler, işlem ve hafıza denetimi, güç denetimi).

### **2. Kütüphaneler (Libraries)**

Çekirdeğin üstünde yer alan katman genelde C++ ve C dilleri ile yazılmış kütüphaneleri içerir (libc, SSL). Bu katmada sistem kütüphaneleri, mp3, mpeg4, jpg gibi çoklu ortam bileşenleri için medya kütüphaneleri ve 2D/3D grafikler için OpenGL/SGL içeren kütüphaneler bulunur.

Android İşletim Sistemi kendi verilerini tutabildiği bir SQLite isimli bir veritabanına sahiptir. Kütüphaneler katmanında veri tabanı için SQLite kütüphaneleri gibi temel kütüphaneler de yer alır.

### **3. Android Çalışma Zamanı (Android Runtime)**

Android'i mobil Linux uygulamasından ayıran en önemli katmandır. Android alt seviye işler için (hafıza yönetimi, donanım sürücüler gibi) Linux kernelini kullanmaktadır ve temel Java kütüphanelerini içerir. Bu katmanda Çekirdek Kütüphaneleri ve Dalvik Sanal Makinesi yer almaktadır.

### 3.1. Çekirdek Kütüphaneleri (Core Libraries)

Çekirdek kütüphaneleri, Java için veri yapılarını, hizmetleri, dosya erişimi, ağ erişimi ve grafik bileşenlerini de içermektedir.

### 3.2. Dalvik Sanal Makinesi (DVM – Dalvik Virtual Machine)

Dalvik Sanal Makinesi (DVM) Android işletim sisteminin en önemli bileşenidir. Android, tek bir cihaz üzerinde çoklu işlemleri etkili bir şekilde çalışmasını sağlayan DVM'i kullanır. DVM, bellek yönetimi, işlemler ve güvenlik gibi düşük seviye görevleri yoluna koymak için cihazdaki linux çekirdeğini kullanır ve minimum bellek kullanımı için optimize edilmiş bir format olan Dalvik dosyalarını çalıştırır. DVM özetle;

Belleği verimli kullanır.

Her süreç için ayrı bir dalvik sanal makinesi çalıştırılır.

Java sınıf dosyalarının yapısını optimize edilmiş .dex formatına dönüştürür.

**DEX Dosyaları:** Android sisteminin altındaki işgücünün en dikkat çekici özelliklerinden biri, Java bayt kodunu kullanmamasıdır. Bunun yerine, DEX adı verilen Dalvik yürütülebilir dosyalarını kullanılır.

Android programları, .dex (Dalvik Yürütülebilir) dosyalara derlenir ve bunlar .apk' da cihazdaki tek bir dosyaya sıkıştırılır.

## 4. Uygulama Çatısı (Application Framework)

Android uygulamalarının yazılımını oluştururken yazılımcıya uygulama çatısı sağlayan uygulama servisleri;

**Aktivite Yöneticisi (Activity Manager):** Aktivitelerinizin yaşam çemberini kontrol eder. Aktivite yığınının yönetimini içerir.

**Görünümler (Views):** Aktiviteler için kullanıcı ara yüzü yapılmasında kullanılır.

**Uyarı Yöneticisi (Notification Manager):** Kullanıcılara yapılan bildirimler ve uyarılar için uyumlu ve tutarlı işlev sağlar.

**İçerik Sağlayıcılar (Content Providers):** Uygulamanın veri paylaşımını sağlar. Telefon rehberi, resim, müzik vb. verilerin uygulamalarca erişimini sağlayan arabirimlerdir. SQL benzeri erişim ara yüzüne sahiptirler.

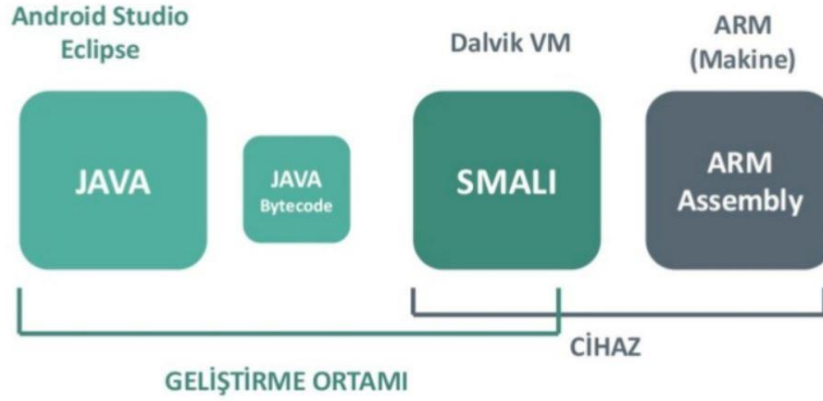
**Kaynak Yöneticisi (Resource Manager):** Dışarıda tutulmak üzere diziler ve grafikler gibi kodsuz kaynakları destekler.

## 5. Uygulamalar (Applications)

Android uygulama çatısındaki servisler ve sınıflar kullanılarak oluşturulan yerel ve 3.parti Android uygulamalarını kapsar. Yerel uygulamalar arasında e-mail istemcisi, sms programı, takvim, google maps, telefon rehberi gibi temel uygulamalar yer almaktadır.

## Android Uygulamaları

- Java + Android SDK ile geliştirilir.
- Android Dalvik VM ile çalıştırılır.
- Java ---> .class ---> .dex



**Smali Code:** apk'ların kaynak koduna yakın bir şekilde decompile edilmiş halidir. Ancak smali kodları yürütülebilir makine kodları ile java kaynak kodları arasında bir yerde bulunmaktadır bu nedenle okunması java diline göre biraz zor olsa da makine dilini daha düzgün bir şekilde yorumlayarak çıktı verebilmektedir.

### DEX ve Smali kod dosyaları arasındaki fark;

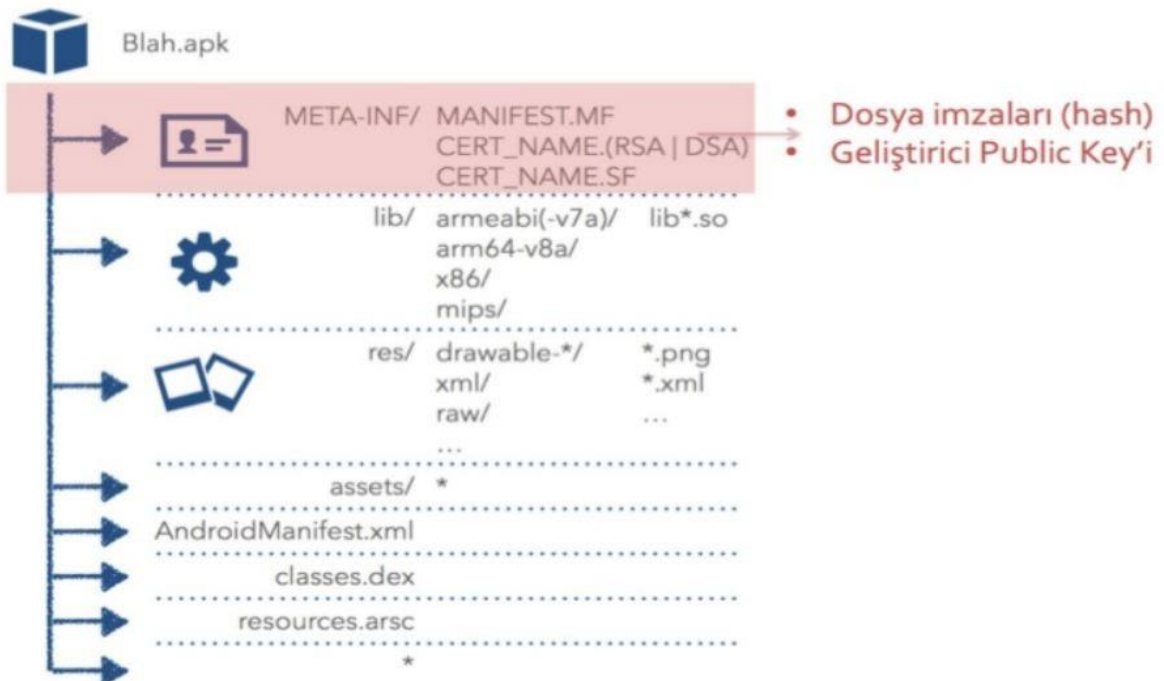
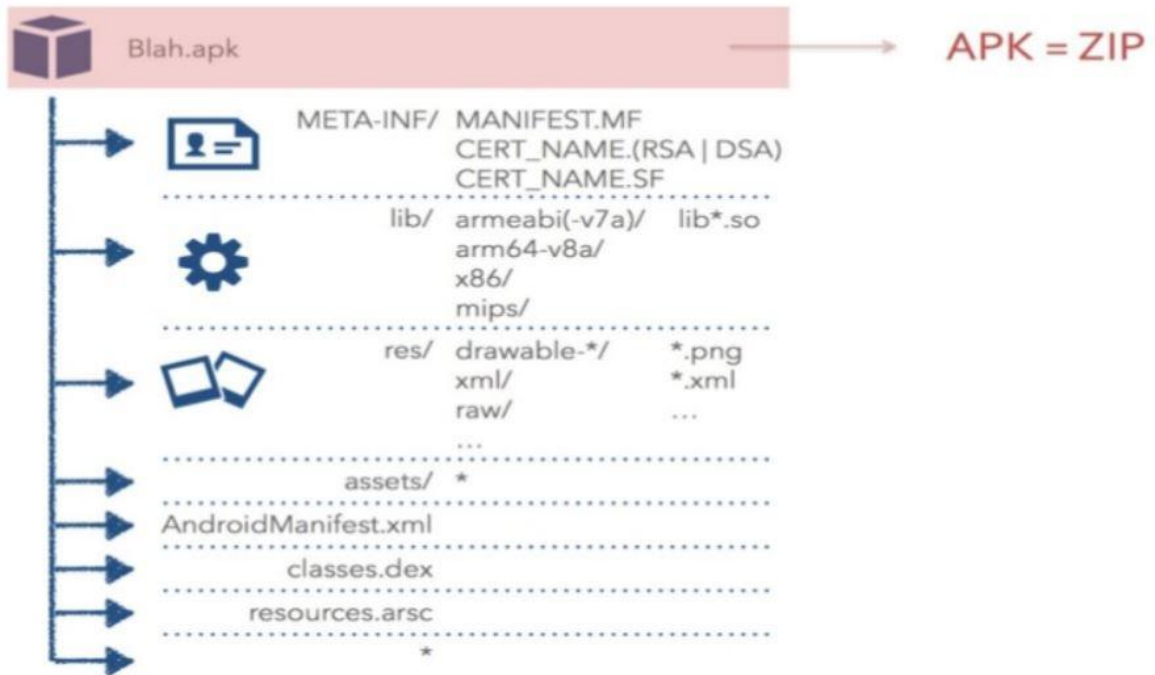
Bir uygulama kodu oluşturduğunuzda, apk dosyası ikili Dalvik bayt kodunu içeren bir .dex dosyası içerir. Bu, platformun gerçekten anladığı biçimdir. Bununla birlikte, ikili kodu okumak veya değiştirmek kolay değildir, bu nedenle insan tarafından okunabilir bir temsile dönüştürmek için araçlar vardır. İnsanların okuyabileceği en yaygın biçim Smali olarak bilinir.

**APK:** Android Application Package, Android uygulama kurulum dosyasıdır. Dosya ZIP dosya formatına sahip .apk uzantılı dosyalardır. APK dosyasının uzantısını .zip olarak değiştirdikten sonra WinZip, WinRAR gibi arşiv programları ile dosya içeriği görüntülenebilir.



**JAR:** Genellikle birçok Java sınıfı dosyasını, verileri ve kaynakları dağıtım için tek bir dosyada toplamak için kullanılan bir paket dosyası biçimidir. JAR dosyaları, Java'ya özgü içeren arşiv dosyalarıdır. ZIP biçiminde oluşturulmuştur ve genellikle .jar dosya uzantısına sahiptir. ZIP ve RAR dosyalarını biliyorsanız JAR dosyası da aslında aynı şeydir. Aradaki fark, JAR dosyalarının Java Runtime Environment tarafından kullanılmak üzere tasarlanmış uygulamalardır.

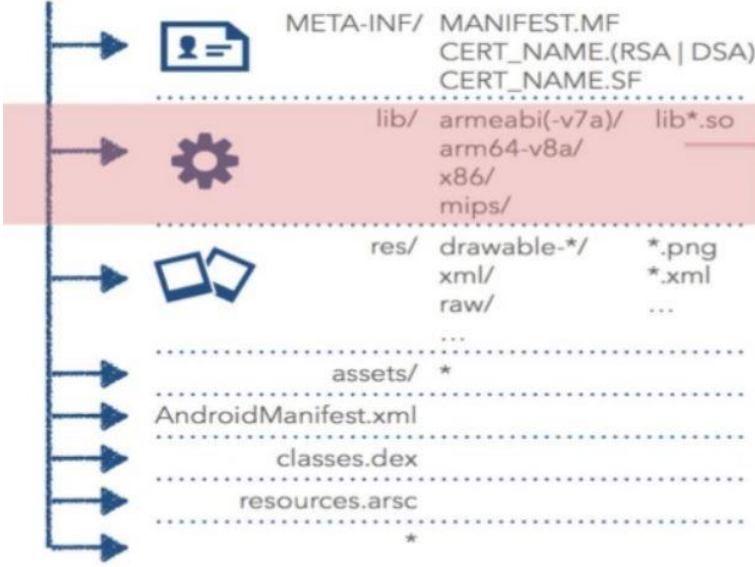
## Android Paket İçeriği







Blah.apk

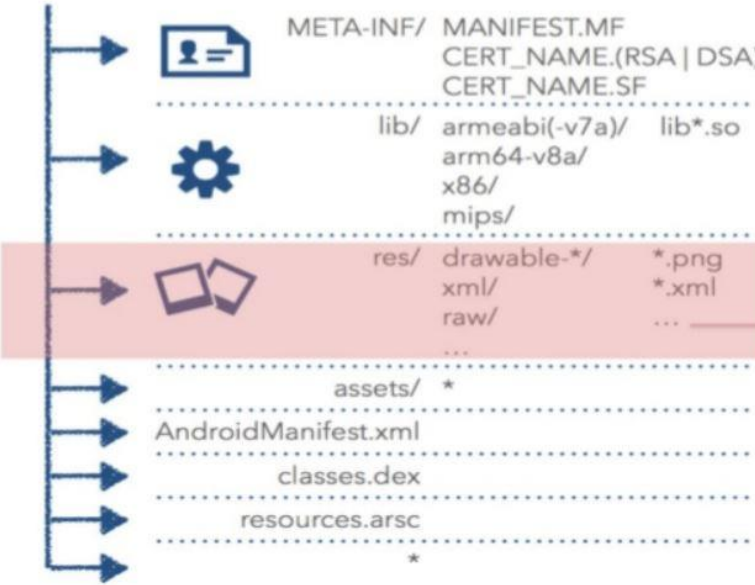


- İşlemci mimarisine göre compile edilmiş native kütüphaneler (Native ELF dosyaları)

- JAR Dosyaları (kütüphaneler)



Blah.apk

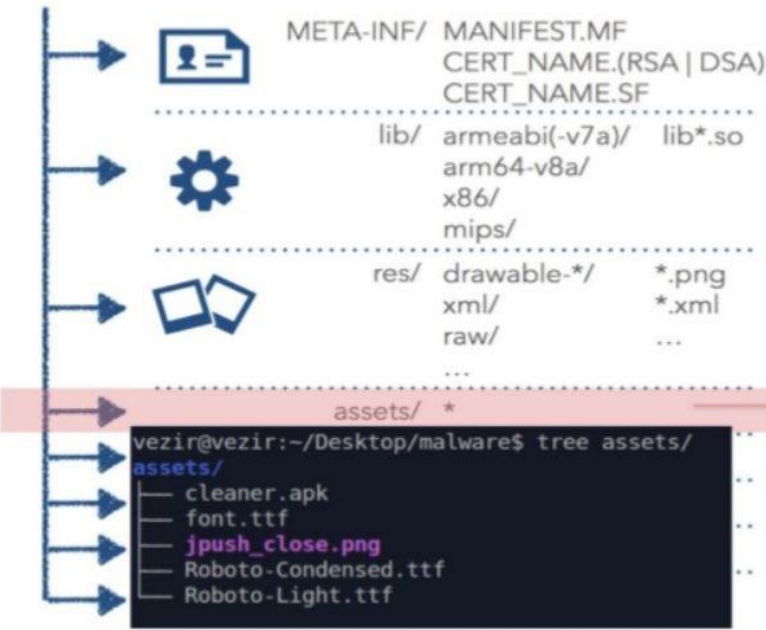


- anim:** Compile edilmiş animasyon dosyaları
- drawable:** Resim dosyaları
- layout:** UI/view tanımlamaları
- values:** Diziler, renkler, style'lar, string'ler dimensions
- xml:** Compile edilmiş XML dosyaları
- raw:** Compile edilmemiş raw dosyalar

Compile işlemi AAPT (Android Asset Packaging Tool) tarafından yapılır



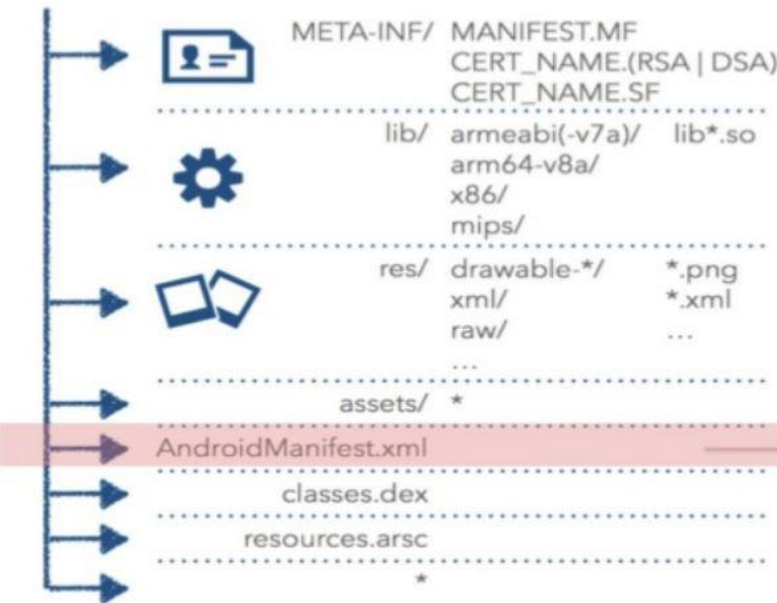
Blah.apk



- Çoğu zaman raw dosyalar bulunur.
- Resimler, fontlar, ses dosyaları
- Bazı malware'ler bu dizinde cihaza kurmak üzere APK dosyaları saklarlar



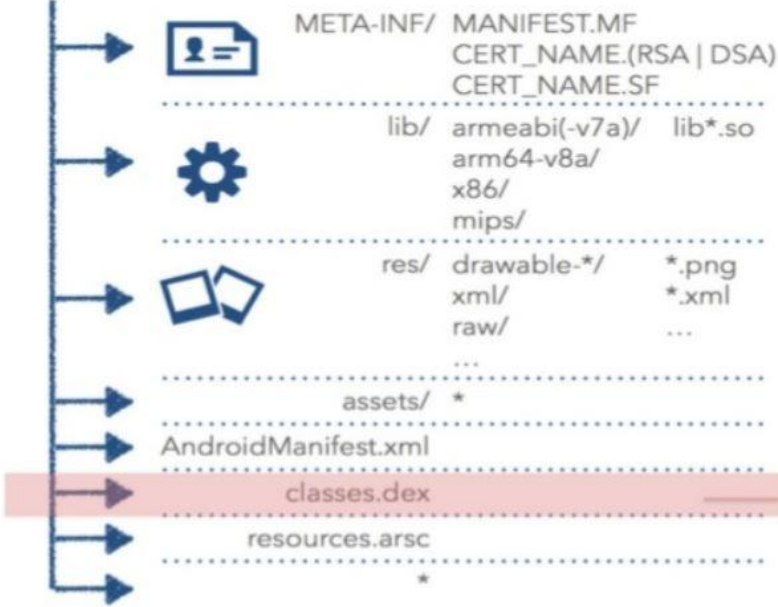
Blah.apk



- Uygulama meta-data'ları
  - Paket ismi
  - Versiyon bilgisi
  - ...
- Uygulamanın talep ettiği izinler
- Uygulamada bulunan komponentler
  - Activity
  - Service
  - Broadcast Receiver
  - Content Provider
- Compile edilmiş olarak paket içerisinde yer alır.



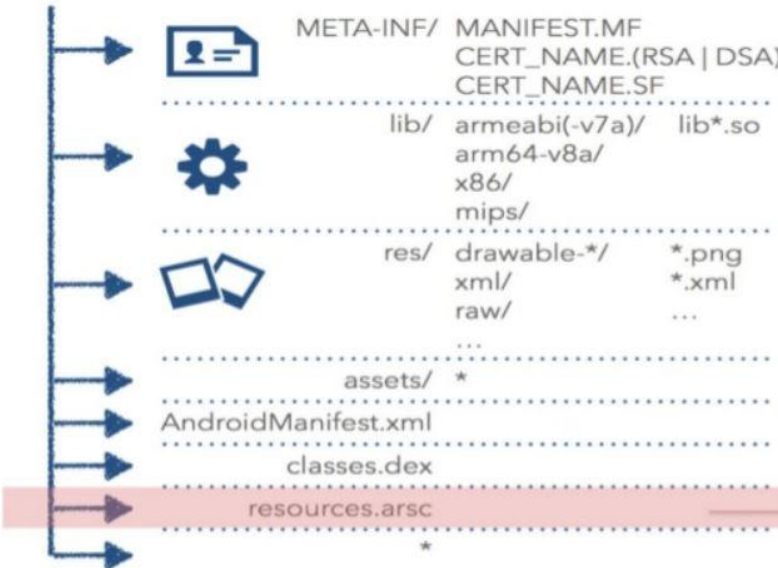
Blah.apk



- DEX: Dalvik Executable
- Android'in EXE'si
- Dalvik VM üzerinde çalışır
- DEX: Dalvik VM için compile edilmiş class dosyaları



Blah.apk



- Compile edilmiş resource'lar
  - R.java
  - string.xml
  - ids.xml
  - layouts.xml

## Kaynak Kod Dönüşümü

### Decompile (Kaynak Koda Dönüştürme)



#### **Dex2jar**

Adından da anlaşılacağı üzere .dex dosyalarını jar dosyalarına çevirmektedir. Class dosyasına dönüştürülmüş olan Android uygulaması, **JD-GUI** aracı ile kaynak koduna (decompile) geri dönüştürülebilir.

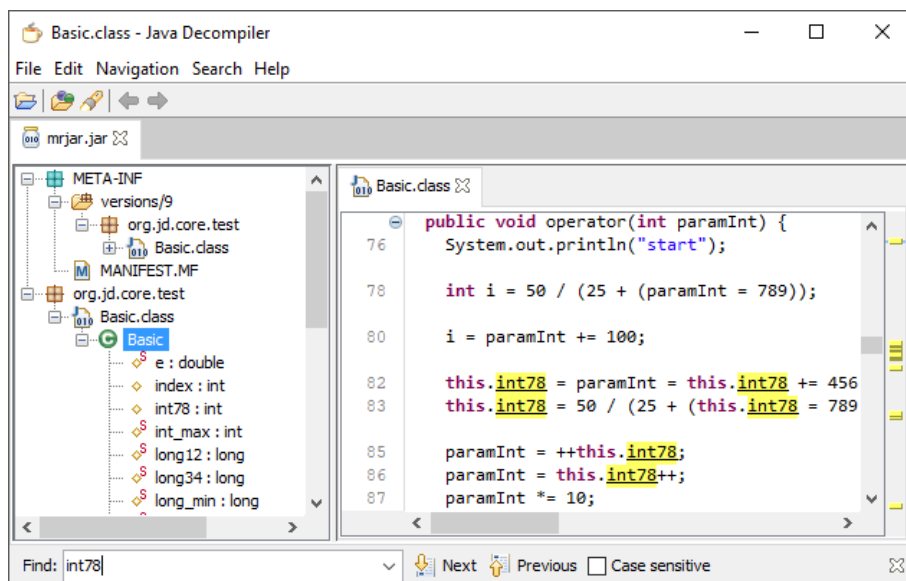
Resimde görüldüğü üzere .apk dosyamızı .jar haline getirebiliriz.

```
C:\Mobil Pentest\dex2jar-0.0.9.15>dex2jar.bat insecurebank.apk
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar insecurebank.apk -> insecurebank_dex2jar.jar
Done.

C:\Mobil Pentest\dex2jar-0.0.9.15>
```

#### **JD-GUI**

JAR haline getirdiğimiz dosyamızı görüntülemek için kullanabiliriz.



- Decompile edilmiş JAR kodu tekrar compile edilerek çalıştırılabilir hale getirilemez.
- Decompile edilen kod yaklaşık koddur. %100 geri dönüşüm gerçekleşmez.
- Dex2jar çıktısından elde edilen JAR kodu çalıştırılmaz.
- Dalvik Bytecode, JAR koduna dönüştürülerek kolay okunabilir ve anlaşılabilir hale gelir.

### Disassembling (Makine Dilini Montaj Diline Çevirmek)



DEX kodunu Smali kodu dönüştürüp okunabilir hale getirdikten sonra analiz edebilmek için kullanılan bazı araçlar mevcuttur. Bunlar;

- Baksmali
- Dedexer
- apktool

Biz burada APKTool aracını kullanacağız.

Aşağıda gördüğümüz gibi DEX dosyası, okunabilir Dalvik Bytecode'a (Smali koduna) dönüştürülüyor.

```
Komut İstemi
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\Ahmet GUREL>cd C:\

C:\>cd "Mobil Pentest\APKTool"

C:\Mobil Pentest\APKTool>apktool.bat b insecurebank.apk
Exception in thread "main" brut.androlib.AndrolibException: brut.directory.PathNotExist: apktool.yml
    at brut.androlib.Androlib.readMetaFile(Androlib.java:143)
    at brut.androlib.Androlib.build(Androlib.java:160)
    at brut.androlib.Androlib.build(Androlib.java:155)
    at brut.apktool.Main.cmdBuild(Main.java:182)
    at brut.apktool.Main.main(Main.java:67)
Caused by: brut.directory.PathNotExist: apktool.yml
    at brut.directory.AbstractDirectory.getFileInput(AbstractDirectory.java:183)
    at brut.androlib.Androlib.readMetaFile(Androlib.java:139)
    ... 4 more

C:\Mobil Pentest\APKTool>
```

Bunun gibi tersine mühendislik uygulamalarına karşı alabilecek önlemler;

## Kod Karmaşıklaştırma (Obfuscation)

- Kullanılmayan sınıflar ve metodlar temizlenir.
- Bytecode optimize edilir.
- Geriye kalan sınıflar, metodlar, alanlar ve değişkenler anlamsız kısa isimlerle adlandırılır.

- Obfuscation işlemi için kullanılan bazı araçlar mevcuttur. Bunlar;

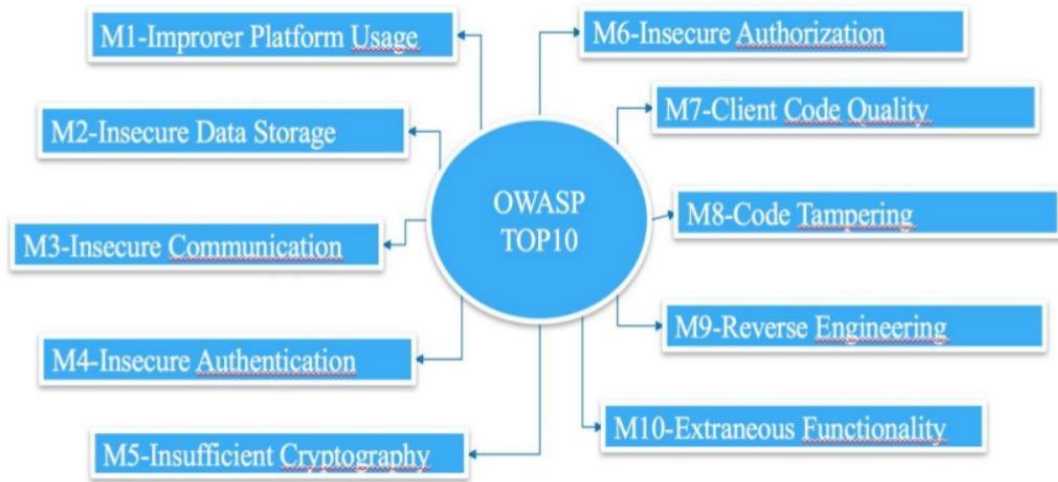
- Proguard
- DexGuard
- Allatori

Obfuscation örneği;

```
public class MyVehicleClass{  
    private Motor    myMotor;  
    private Tekerlek myTekerler;  
    private int      vitesSayisi;  
  
    public int suratHesapla(int sure){  
        ...  
        return sonSurat;  
    }  
}
```

```
public class A{  
    private B    a;  
    private C    b;  
    private int  c;  
  
    public int a(int a){  
        ...  
        return c;  
    }  
}
```

## OWASP Mobile Top 10



- OWASP, 2-3 yılda bir mobil uygulamaları üzerinde tespit edilen önemli zafiyetleri yayınlır.
- OWASP Testing Metodolojisi bu zafiyetlerin nasıl giderileceğine dair bir liste içerir.
- Bu zafiyetler önem derecesine göre aşağıdaki şekilde sıralanabilir;

### **M1 – Improper Platform Usage (Hatalı Platform Kullanımı)**

- Platform güvenlik kontrollerinin hatalı veya kötüye kullanılması.
- Mobil uygulamalarda bu riski yaşamamanın çeşitli yolları vardır;
- Yayımlanmış klavuz ihlali,
- Sözleşme veya yaygın bir uygulama hali,
- Kasıtsız kötüye kullanma

### **M2 – Insecure Data Storage (Güvensiz Veri Saklama)**

- Bu kısım güvensiz veri depolama ve istenmeyen veri sızıntılarını kapsar.
- SQL Databases
- Log files
- Binary data stores
- Cookie stores
- SD card
- Cloud synced.

### **M3 – Insecure Communication (Güvenli Olmayan İletişim)**

- Güvenlik açısından yanlış SSL versiyonlarının kullanımı
- Hassas verilerin clear-text olarak gönderilmesi
- İletişimin sağlandığı kanallar arasında zayıf iletişimlerin kurulması.

### **M4 – Insecure Authentication (Güvensiz Doğrulama)**

- Mobil uygulama herhangi bir parolayı veya paylaşılan sırları cihazda yerel olarak saklarsa, güvensiz kimlik doğrulama sorunuyla karşılaşır.
- Mobil uygulama bir şifre girmeyi kolaylaştırmak için zayıf bir şifre politikası kullanıyorsa, güvensiz kimlik doğrulama uygular.

### **M5 – Insufficient Cryptography (Yetersiz Şifreleme)**

- Hassas kod bilgileri şifrelenir. Ancak yine de şifreleme yetersiz kalabilir.
- Yaygın hatalar;
- Zayıf şifreler
- Yanlış şifreleme

### **M6 – Insecure Authorization (Güvensiz Yetki)**

- Yetkilerde hataları yakalama
- Örneğin; cihaz kayıt, kullanıcı tanımlama kimlik doğrulama sorunları farklıdır.
- Eğer uygulamada kullanıcı kimlik doğrulama yoksa bu kimlik doğrulama hatası değil başarısız yetkilendirme hatasıdır.

### **M7 – Client Code Quality (İstemci Kod Kalite Sorunları)**

- Mobil istemciye kod düzeyinde uygulama hatasıdır.
- Bufferoverflow, format string güvenlik açıkları ve çözümün mobil cihaz üzerinde çalışan bazı kodu tekrar yazmak olan çeşitli kod düzeyindeki hatalar gibi riskleri bu yakalar.

### **M8 – Code Tampering (Kod Kurcalama)**

- Uygulama mobil cihaza yüklendikten sonra kod ve veri kaynakları orada bulunur.
- Bir saldırgan doğrudan uygulamanın kodunu veya kullandığı sistem API'lerini değiştirebilir.
- Böylece saldırgan kişisel ya da parasal kazanç için yazılımın kullanım amacını yıkarak kötü amaçlı kullanması yöntemidir.

**M9 – Reverse Engineering (Tersine Mühendislik)**

- Uygulamanın kaynak kodu, kütüphaneleri, algoritması ve diğer kaynakların tespitidir.
- Saldırgan doğabilecek açıklıkları, şifreleri vb. bilgileri yararına kullanabilir.

**M10 – Extraneous Functionality (Gereksiz İşlevsellik)**

- Genellikle geliştiricilerin arka kapı bırakması
- Örneğin, geliştirici bir uygulamada şifre unutmuş olabilir.
- Diğer bir örnek de test sırasında 2 faktörlü kimlik doğrulamayı devre dışı bırakmasıdır.

\* Bu noktaya kadar *“Android yapısı/mimarisi, APK dosyasının içeriği, bir mobil uygulamanın kaynak kod dönüşümleri, kaynak kod karmaşılaştırması ve OWASP Mobile Top 10”* gibi daha çok teorik konularından bahsettik.

Şimdi ise bir Android Emülatör kurulumu gerçekleştirerek, emülatör üzerinde mobil sızma testlerinde kullanılan bazı araçların kullanımını gösterdikten sonra son olarak zafiyetli bir uygulama üzerinde uygulamalar yapıp bitireceğiz.



## Mobil Sızma Testi Araçları

### ADB (Android Debug Bridge)

```
platform-tools -- -bash -- 95x18
~/canary_fresh_sdk/platform-tools -- -bash
lgleason@MacBook-Pro-40:~/canary_fresh_sdk/platform-tools$ ./adb
Android Debug Bridge version 1.0.40
Version 4986621
Installed as /Users/lgleason/canary_fresh_sdk/platform-tools/./adb

global options:
-a      listen on all network interfaces, not just localhost
-d      use USB device (error if multiple devices connected)
-e      use TCP/IP device (error if multiple TCP/IP devices available)
-s SERIAL use device with given serial (overrides $ANDROID_SERIAL)
-t ID   use device with given transport id
-H      name of adb server host [default=localhost]
-P      port of adb server [default=5037]
-L SOCKET listen on given socket for adb server [default=tcp:localhost:5037]

general commands:
devices [-l]    list connected devices (-l for long output)
help           show this help message
```

ADB, Android bir cihazla iletişim kurmanızı sağlayan, çok yönlü bir komut satırı aracıdır. ADB komutları, uygulama yükleme, hata ayıklama, cihazdan ekran görüntüsü alma, video çekme gibi birçok işlemi sizin için kolaylaştırır.

### ADB Komutları

#### ADB Debugging

- adb devices : Bağlı olan cihazların listesini görüntüler.
- adb forward --list : Tüm socket bağlantılarını listeler.
- adb forward tcp:xxxx tcp:yyyy : xxxx bağlantı noktasının yyyy'e aktarılmasını sağlar.
- adb kill-server : Sunucu işlemini sonlandırır.

#### Wireless

- adb connect : Hedef aygıt ile bağlantı kurmaya yarar.
- adb tcpip 5555 : Hedef aygıtı 5555 numaralı bağlantı noktasında bir TCP/IP bağlantısı dinleyecek şekilde ayarlar.
- adb connect 192.168.xxx.xxx : Wi-Fi üzerinden bir cihaza bağlar.
- adb devices : ADB'ye bağlı tüm cihazları listeler.
- adb usb : Usb modunda yeniden başlatır

#### Package Manager

- adb install : Hedef aygıtı yükleme yapmak için kullanılır.
- adb install test.apk : Tek bir paketi aktararak yükleyebilirsiniz.
- adb install-multiple test.apk test2.apk : Birden fazla apk'yı aktararak yükleyebilirsiniz.
- adb install-multi-package test.apk demo.apk : Birden fazla apk'yı cihaza aktararak atomik olarak kurabilirsiniz.

- adb install -r test.apk : Uygulamada ki mevcut verileri koruyarak yeniden yükleyebilirsiniz.
- adb install -t test.apk : Test paketlerine izin verebilirsiniz. (yalnızca hata ayıklanabilir paketler için)
- adb install -d test.apk : Sürüm kodunun eski sürümlere geçmesine izin verir ( Yalnızca ayıklanabilir paketler için )

adb install -g test.apk : Tüm çalışma zamanı izinlerini ver ( Uygulama bildiriminde listelenen tüm izinleri ver.

adb install --instant test.apk : Uygulamayı geçici bir yükleme olarak yüklenmesine olanak sağlar.

adb install --fastdeploy test.apk : Hızlı kurulmasına neden olur.

adb install --no-streaming test.apk : Paket yöneticisini her zaman ayrı adımlar olarak çağırıp apk'yı yüklemeye olanak sağlar.

### **ADB uninstall, shell ve pm komutları**

adb uninstall test.apk : test.apk'yı cihazdan silmeye yarar

adb unistall -k test.apk : Önbellek de bulunan verileri saklar daha sonra silme işlemi gerçekleştirir.

adb shell pm list packages : Tüm paketleri listeler

adb shell pm list packages -f : İlişkili paketler listelenir

adb shell pm list packages -a : Bilinen tüm paketler listelenir (APEX'ler hariç)

adb shell pm list packages --apex-only : Sadece APEX paketlerini listeler

adb shell pm list packages -d : Sadece devre dışı bırakılmış paketleri listeler

adb shell pm list packages -e : Sadece etkin paketleri listeler

adb shell pm list packages -s : Sadece sistem paketlerini listeler

adb shell pm list packages -3 : Sadece üçüncü taraf paketlerini listeler

adb shell pm list packages -i : Sadece kurulum dosyalarını gösterir.

adb shell pm list packages -U : Paket Uid'sini göster

adb shell pm list packages -u : Kaldırılmış paketleri dahil et

adb shell pm list packages --show-versioncode : Version sürümü için

adb shell pm list packages --uid UID : Sadece bu uid'ye sahip paketleri gösterir

adb shell pm list packages -user USER\_ID : Belirtilen id'ye sahip alt paketleri gösterir

adb shell pm path com.android.chrome : Yüklü paket adının apk yolunun adını yazdırın

adb shell pm clear com.test.abc : Paketle ilişkili tüm verileri siler.

### **File Manager**

adb pull/mnt/sdcard/Download/test.apk pc.apk : Android cihazınızdan dosya kopyalamanıza yarar

adb push pc.apk mnt/sdcard/Download/test.apk : Local cihazınızdan Android cihazınıza dosya kopyalamanızı sağlar

adb shell ls /system/bin : Dosya yollarını ve dosyaları listeler

adb shell ls -a : Tüm gizlilik içeren dosyaları listeler

adb shell ls -d : Sadece dosya yollarını listeler

adb shell ls -R mntsdcard/Download : Yinelemeli alt klasörlerin listeleri

adb shell cd /mnt/sdcard/Download : Dosya yolunu değiştirirsiniz

adb shell rm /mnt/sdcard/Download/test.apk : rm dosyaları, dizinleri ve sembolik bağlantıları kaldırmak için kullanılan bir komut satırı yardımcı programıdır

adb shell rm -f /mnt/sdcard/Download/test.apk : Zorla, onay almadan kaldırır.

adb shell rm -i /mnt/sdcard/Download/test.apk : Kaldırma işlemi onay alarak yapar.

adb shell rm -rR /mnt/sdcard/Download : Dizin içerisinde yinelemeli olarak kaldırma işlemi yapar.

adb shell rm -v /mnt/sdcardDownload/test.apk : Bilgiler vererek kaldırma işlemi yapar.

mkdir /sdcard/tmp : Dizin oluşturmak için kullanılır.

mkdir -m 777 sdcard/tmp set permission mode : İzinleri belirterek dizin oluşturur.

mkdir -p sdcardtmp/sub1/sub2 create parent directories as needed : Gerekli izin yapısını oluşturarak dizini oluşturur.

adb shell touch mntsdcard/Download/test.txt : Bir dosyanın zaman damgalarını oluşturmak için kullanılır.

adb shell pwd : Şifreyi deęiřtirmek için kullanılır.

cp sdcard/test.txt sdcard/demo.txt : dosyayı kopyalamak için kullanılır.

adb shell mv mnt/sdcard/Download/test.txt mnt/sdcard/DCIM/test.txt : Dosyaları ve dizinleri bir yerden başka bir yere taşımak için kullanılır.

adb shell mv -f mnt/sdcard/Download/test.txt mnt/sdcard/DCIM/test.txt : İstenen dizine istenen dosyayı taşıır ve dosyayı eski dizinden siler.

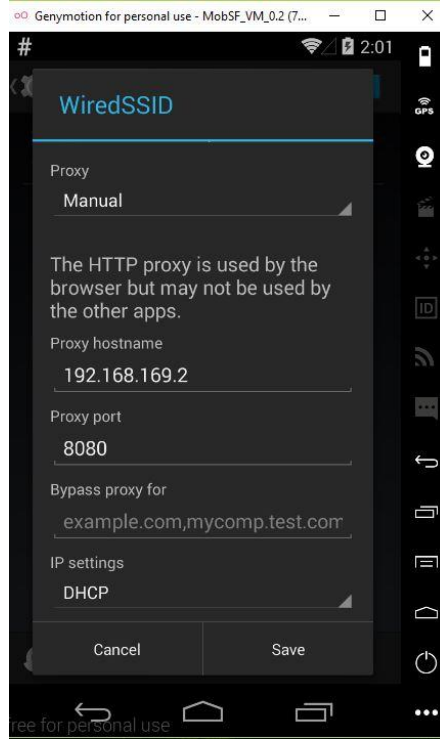
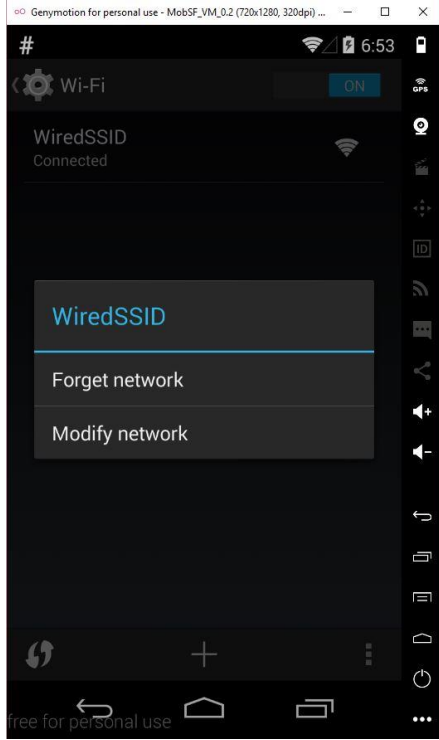
adb shell mv -i /mnt/sdcard/Download/test.txt /mnt/sdcard/DCIM/test.txt : Eęer taşınacak dizinde aynı isimli bir dosya var ise uyarı verir.

adb shell mv -n /mnt/sdcard/Download/test.txt /mnt/sdcard/DCIM/test.txt : Taşınacak dizinde aynı isimde bir dosya var ise üzerine yazmaz.

## Burp Suite

- Burp gelişmiş bir proxy yazılımıdır.
- Bunun dışında birçok teste yardımcı olmakta ve imkan tanımaktadır.
- Web Testlerinin olmazsa olmazı Burp Suite, mobil testlerimizde de o kadar önemlidir.

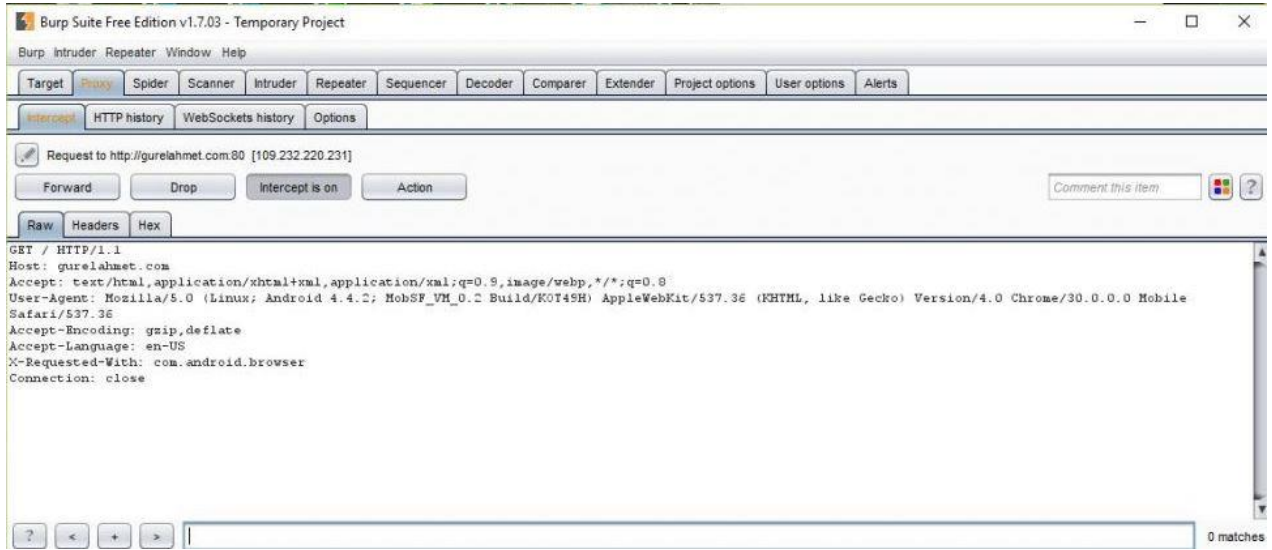
Burp Suite telefonumuza bağlayalım;



Ayarlara (Settings) girerek daha sonra Wi-Fi ye tıklayarak WiredSSID'nin üzerine basılı tutarak Modify network diyerek Proxy belirliyoruz. Burada IP adresi test yaptığınız makinenin IPSidir.



Burp Suite açarak, Proxy'ye tıklayıp, oradan Options sekmesine gelip, Add e basıyoruz ve resimdeki gibi kendi IP adresinizi ve port numaranızı giriyorsunuz.

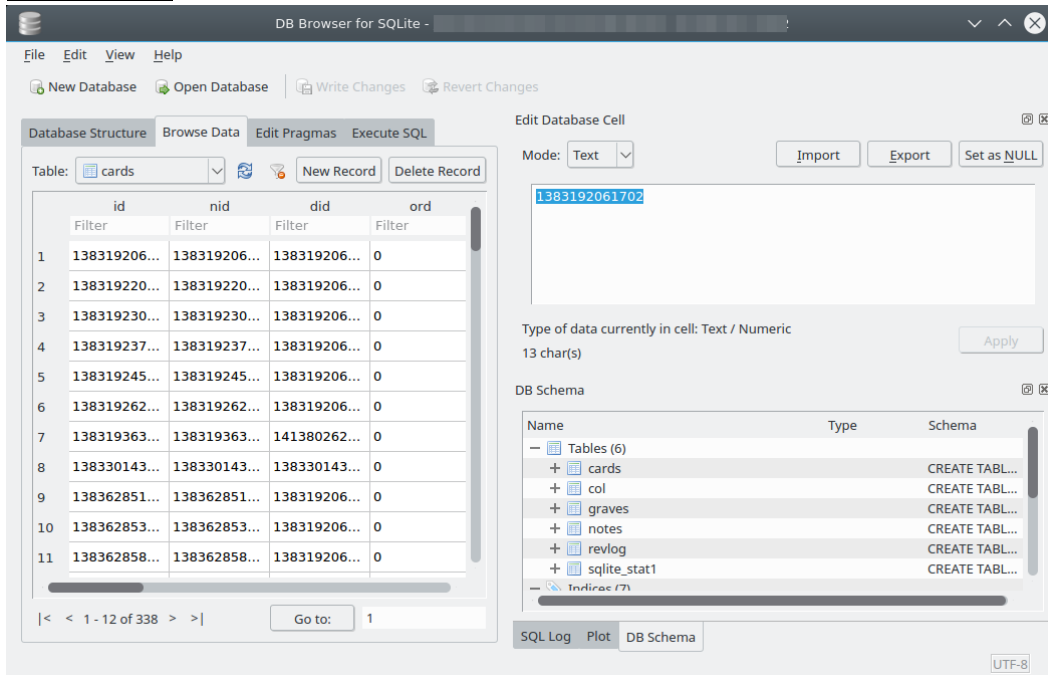


Artık Burp Suite hazır mobil testimde kullanabilirsiniz. Resimde gördüğümüz gibi emulatordeki isteği yakalamakta.

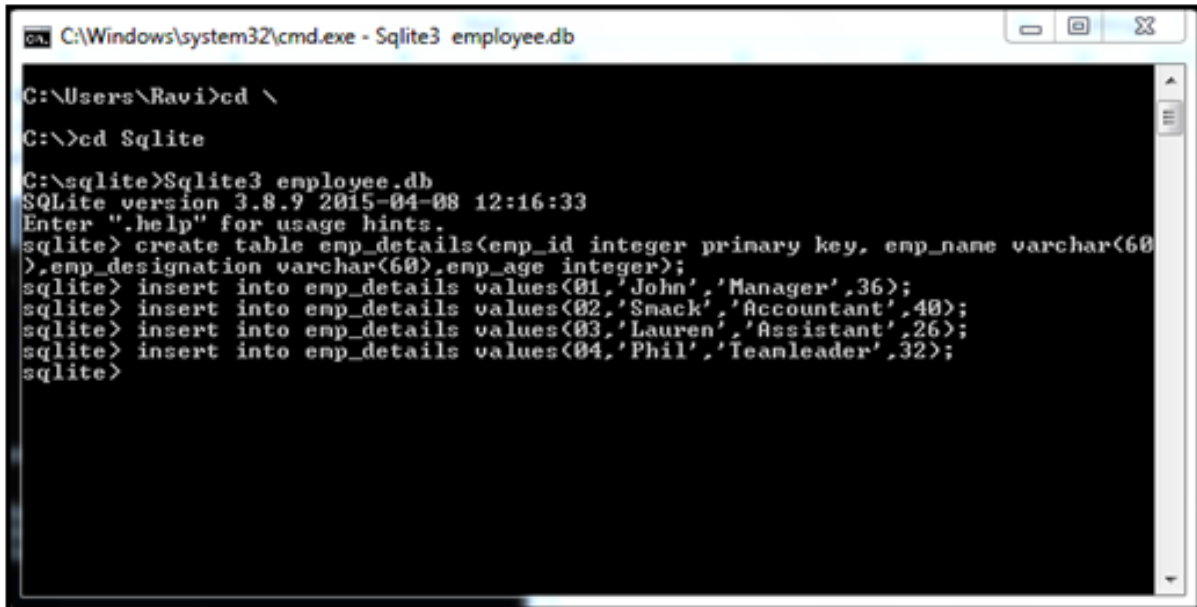
### SQLite Browser ve SQLite3

- Veritabanı incelemelerinde kullanılır. Bir uygulamayı cihazımıza aktardıktan sonra eğer dosyalar cihazda tutuluyorsa veritabanı dosyalarını ADB ile kendi bilgisayarımıza indirebiliriz.
- Bu database dosyalarının içeriğini SQLite Browser ile görüntüleyebiliriz.
- Bunun dışında da SQLite3 ile veritabanını seçerek sorgular yazıp bununla da görüntüleyebiliriz.

### SQLite Browser



## SQLite3 Command Line

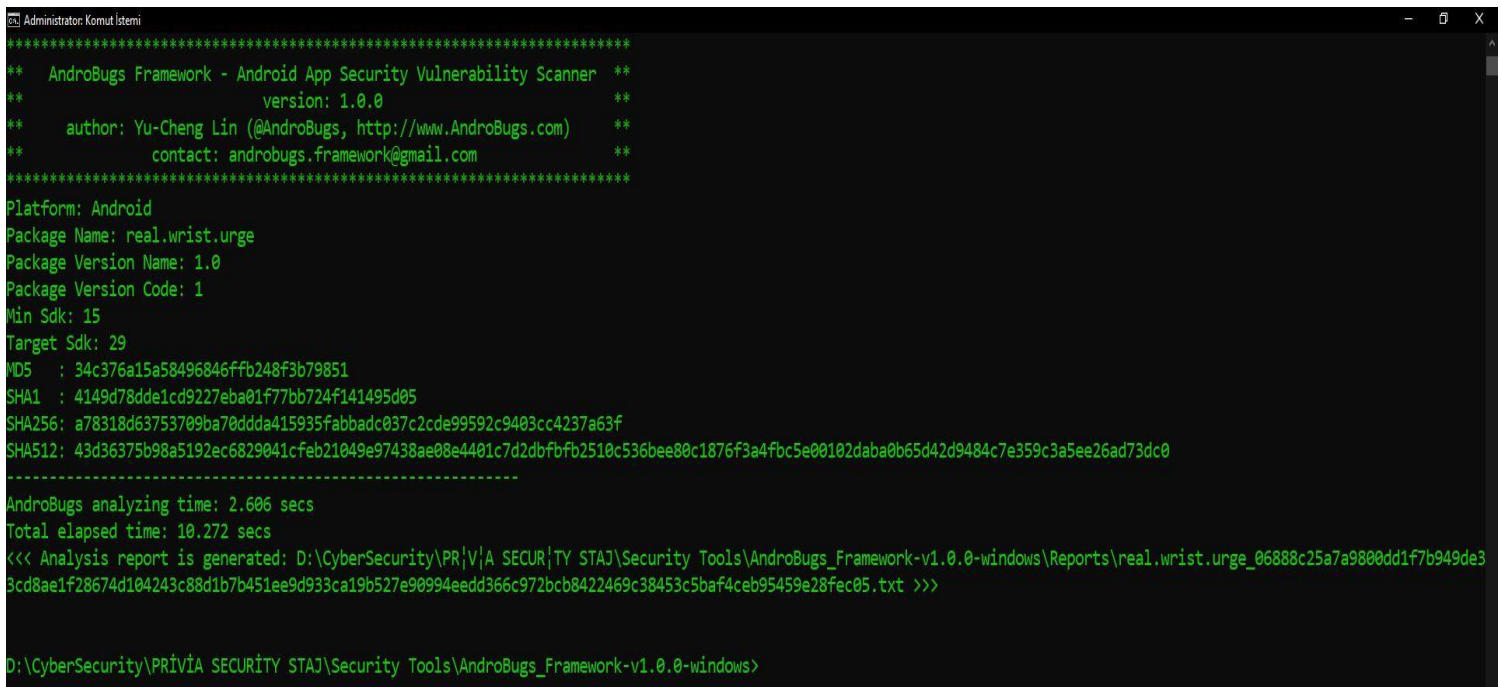


```
C:\Windows\system32\cmd.exe - SQLite3 employee.db

C:\Users\Ravi>cd \
C:\>cd Sqlite
C:\sqlite>Sqlite3 employee.db
SQLite version 3.8.9 2015-04-08 12:16:33
Enter ".help" for usage hints.
sqlite> create table emp_details(emp_id integer primary key, emp_name varchar(60), emp_designation varchar(60), emp_age integer);
sqlite> insert into emp_details values(01,'John','Manager',36);
sqlite> insert into emp_details values(02,'Snack','Accountant',40);
sqlite> insert into emp_details values(03,'Lauren','Assistant',26);
sqlite> insert into emp_details values(04,'Phil','Teamleader',32);
sqlite>
```

## AndroBugs Framework

- AndroBugs Framework, Android uygulamalarda güvenlik testi gerçekleştiren frameworklerden bir tanesidir.
- Kullanımı oldukça basittir. Konsol üzerinden "androbugs -f apk\_dosyası" şeklinde kullanarak frameworkümüzü çalıştırdık.
- Bunun sonucunda kendi klasörünün altında Reports klasörünün altında detaylı rapor oluşmaktadır.

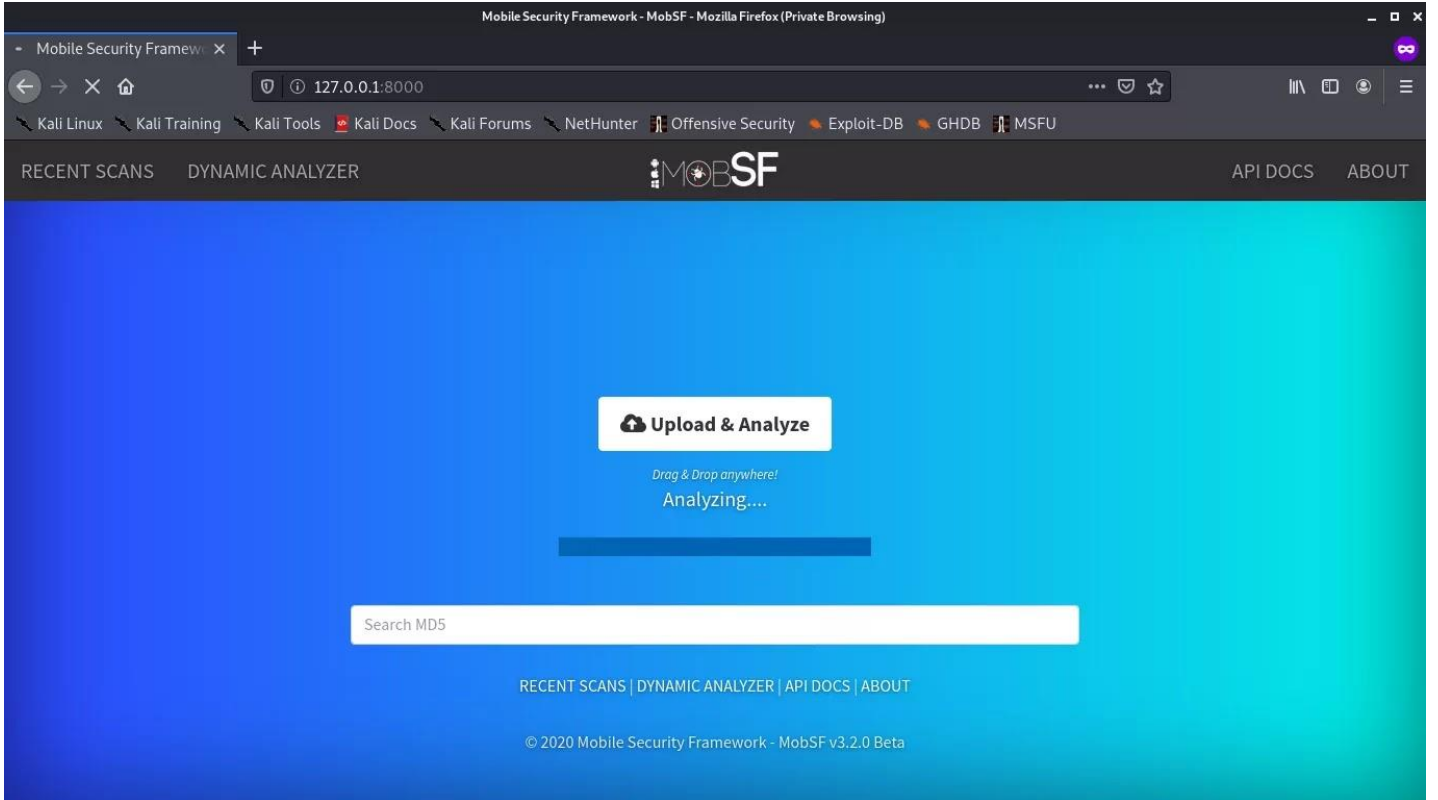


```
Administrator: Komut İstemi
*****
** AndroBugs Framework - Android App Security Vulnerability Scanner **
** version: 1.0.0 **
** author: Yu-Cheng Lin (@AndroBugs, http://www.AndroBugs.com) **
** contact: androbugs.framework@gmail.com **
*****
Platform: Android
Package Name: real.wrist.urge
Package Version Name: 1.0
Package Version Code: 1
Min Sdk: 15
Target Sdk: 29
MD5 : 34c376a15a58496846ffb248f3b79851
SHA1 : 4149d78dde1cd9227eba01f77bb724f141495d05
SHA256: a78318d63753709ba70dda415935fabbadc037c2cde99592c9403cc4237a63f
SHA512: 43d36375b98a5192ec6829041cf2b21049e97438ae08e4401c7d2dbfbfb2510c536bee80c1876f3a4fbc5e00102daba0b65d42d9484c7e359c3a5ee26ad73dc0
-----
AndroBugs analyzing time: 2.606 secs
Total elapsed time: 10.272 secs
<<< Analysis report is generated: D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\AndroBugs_Framework-v1.0.0-windows\Reports\real.wrist.urge_06888c25a7a980dd1f7b949de33cd8ae1f28674d104243c88d1b7b451ee9d933ca19b527e90994eedd366c972bcb8422469c38453c5baf4ceb95459e28fec05.txt >>>

D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\AndroBugs_Framework-v1.0.0-windows>
```

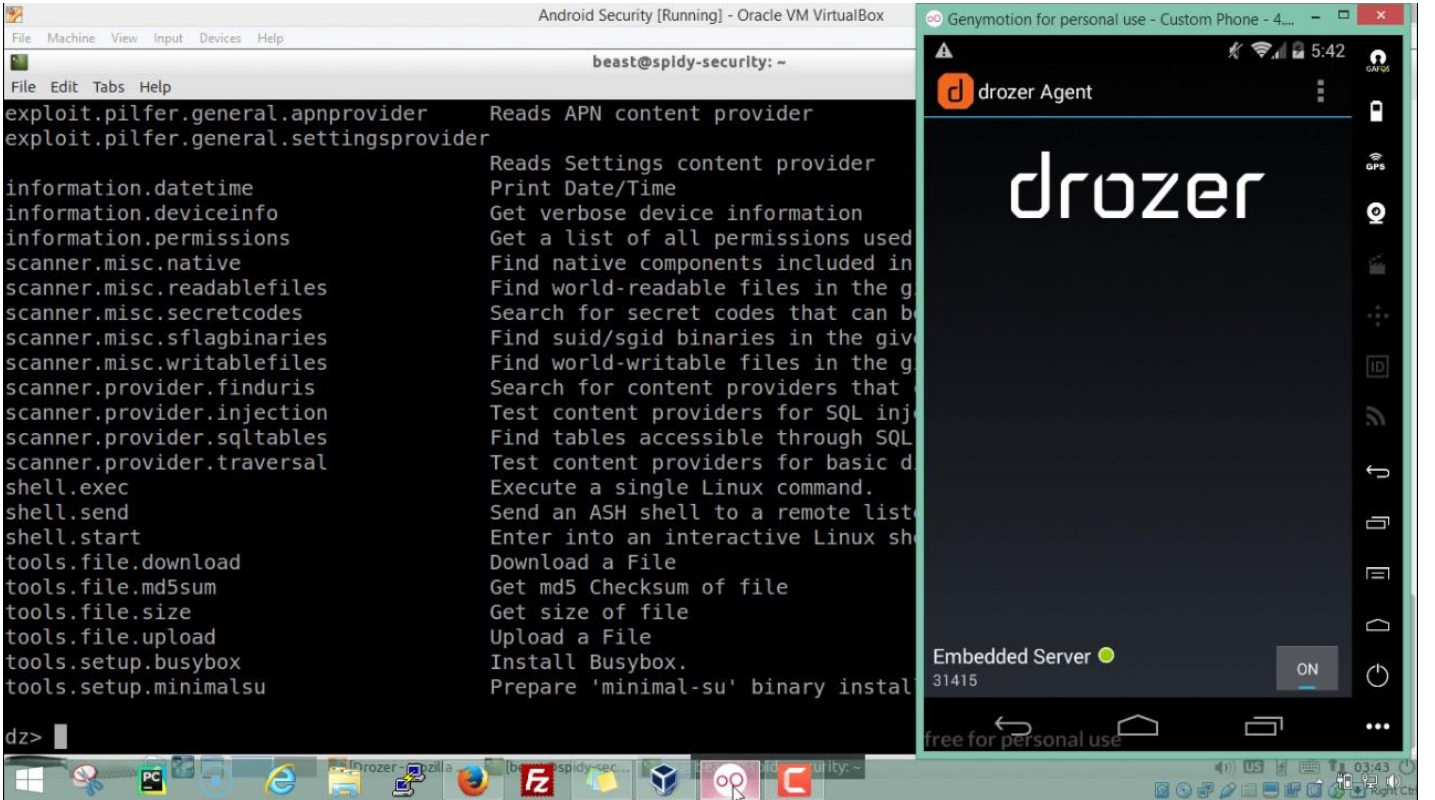
## MobSF (Mobile Security Framework)

- Mobil uygulama analizi yapan bir frameworktür.
- Şu an en kullanışlı ve sağlam araç denebilir. Oldukça popüler ve güzel bir araçtır.
- MobSF'i indirdikten sonra Windows, Linux ve OSX'e kurabiliriz. Kurduktan sonra tarayıcıda localhost olarak çalışmaktadır.
- Herhangi bir APK dosyasını seçtikten sonra analiz etmeye başlar ve bize sonuçlarını gösterir. Oldukça basit bir kullanımı vardır.



## Drozer

- Drozer, mobil testlerde kullanılan dinamik analiz yapan bir framework'tür.
  - Uygulama çalışırken test etme imkanı verir.
  - AndroBugs ve MobSF araçlarında ise statik analiz yaptık fakat uygulama çalışmıyordu. Drozer'da uygulama çalışırken testlerimizi gerçekleştiriyoruz.
  - Drozer'ı kendi bilgisayarımıza kurduktan sonra aynı zamanda emülatördeki mobil cihaza da yükleyerek birbirleri ile haberleşmesini sağlıyoruz.
- 
- Kendi bilgisayarımıza Drozer'i kurduktan sonra "agent.apk" dosyasını emülatöre kurmayı unutmamalıyız.
  - Bunu ister sürükleyip bırak isterseniz de "adb install agent.apk" komutu ile yapabiliriz.
  - Yükledikten sonra agent.apk dosyasını emülatörde açarak "off" durumundan "on" durumuna getirmeliyiz.
  - Daha sonra ADB kullanarak "adb forward tcp:31415 tcp:31415" komutunu verdikten sonra son olarak drozer aracımızın dizini içerisinde "drozer.bat console connect" diyerek drozer'in komut satırına düşebiliriz.





## Drozer Kullanarak Android Uygulama Güvenliği Nasıl Test Edilir?

**1. Paket Bilgilerinin Getirilmesi:** Bağlı cihazlarda bulunan paketleri alabiliriz, ayrıca herhangi bir kurulu paket hakkında bilgi alabiliriz.

Cihazda bulunan tüm paketlerin listesini almak için.

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

Yukarıdaki listeden bir paket adı aramak için

```
dz> run app.package.list -f herhangi_bir_keywords
```

Herhangi bir paket hakkında temel bilgi almak için

```
dz> run app.package.info -a com.facebook.lite
```

**2. Saldırı Yüzeyini Belirleyin:** Bu, güvenlik keşfi parçasıdır. Dışa aktarılan Faaliyetlerin, Yayın Alıcılarının, İçerik Sağlayıcıların ve Hizmetlerin sayısını kontrol ederek başlanır. Komutlar aşağıdaki gibidir:

```
dz> run app.package.attacksurface <package_name>
```

```
3 activities exported
```

```
0 broadcast receivers exported
```

```
2 content providers exported
```

```
2 services exported is debuggable
```

**3. Aktiviteleri Başlatma:** Şimdi, dışa aktarılan aktiviteleri başlatmaya ve kimlik doğrulamasını atlamaya çalışacağız. Bu yüzden dışa aktarılan edilen tüm faaliyetleri başlatmakla başlıyoruz.

```
//Bir dz paketinden etkinlik listesi almak için
```

```
dz> run app.activity.info -a <package_name>
```

```
//Seçili herhangi bir etkinliği başlatmak için
```

```
dz> run app.activity.start --component <package_name> <activity_name>
```

**4. İçerik Sağlayıcılardan Okuma:** Daha sonra uygulama tarafından dışa aktarılan İçerik Sağlayıcılar hakkında daha fazla bilgi toplamaya çalışacağız.

```
//İçerik sağlayıcıları hakkında bilgi almak için:  
dz> run app.provider.info -a <package_name>
```

**Example Result:**

```
Package: com.mwr.example.sieveAuthority:  
com.mwr.example.sieve.DBContentProvider  
Read Permission: null  
Write Permission: null  
Content Provider: com.mwr.example.sieve.DBContentProvider  
Multiprocess Allowed: True  
Grant Uri Permissions: False  
Path Permissions:  
Path: /Keys  
Type: PATTERN_LITERAL  
Read Permission: com.mwr.example.sieve.READ_KEYS  
Write Permission: com.mwr.example.sieve.WRITE_KEYS
```

Yukarıdaki içerik sağlayıcı, Veritabanı Destekli İçerik Sağlayıcı olarak kabul edilebilecek DBContentProvider olarak adlandırılmıştır. İçerik URI'lerini tahmin etmek çok zordur , ancak drozer, yolları tahmin etmek ve erişilebilir içerik URI'lerinin bir listesini belirlemek için çeşitli yolları bir araya getiren bir tarayıcı modülü sağlar. İçerik URI'lerini şu şekilde alabiliriz:

```
//Seçili paket için içerik URI'lerini almak için  
dz> run scanner.provider.finduris -a <your_package>
```

**Example Result:**

```
Scanning com.mwr.example.sieve...  
Unable to Query content://com.mwr.  
example.sieve.DBContentProvider/  
...  
Unable to Query  
content://com.mwr.example.sieve.DBContentProvider/Keys  
Accessible content URIs:  
content://com.mwr.example.sieve.DBContentProvider/Keys/  
content://com.mwr.example.sieve.DBContentProvider/Passwords  
content://com.mwr.example.sieve.DBContentProvider/Passwords/
```

Artık bu içerik URI'lerinden bilgi almak ve hatta veri tabanındaki verileri değiştirmek için diğer drozer modüllerini kullanabiliriz.

```
//Yukarıdaki içerik URI'lerini kullanarak verileri almak veya değiştirmek için:
```

```
dz> run app.provider.query
content://com.mwr.example.sieve.DBContentProvider/Password/ --vertical

_id: 1
service: Email
username: incognitoguy50
password: PSFjqXIMVa5NJFudgDuuLVgJYFD+8w== (Base64-encoded)
email: incognitoguy50@gmail.com
```

Android platformu, verileri depolamak için SQLite veritabanlarını kullanmaya teşvik eder. SQLite veritabanları SQL Enjeksiyonuna karşı savunmasız olabilir. Projeksiyon ve seçim alanlarını manipüle ederek SQL enjeksiyonunu test edebiliriz.

```
//SQL enjeksiyon kullanarak saldırmak için:
dz> run app.provider.query
content://com.mwr.example.sieve.DBContentProvider/Passwords/ --projection ""

unrecognized token: "' FROM Passwords" (code 1): , while
compiling: SELECT '

FROM Passwords
dz> run app.provider.query
content://com.mwr.example.sieve.DBContentProvider/Passwords/ --
selection ""

unrecognized token: "'" (code 1): , while compiling: SELECT *
FROM Passwords WHERE ('
```

Android, yürütmeye çalıştığımız tüm sorguyu gösteren ayrıntılı bir hata mesajı döndürür ve bu, veritabanındaki tüm tabloları listelemek için kullanılabilir.

```
//SQL enjeksiyon kullanarak saldırmak için:
dz> run app.provider.query

content://com.mwr.example.sieve.DBContentProvider/Passwords/ --projection "*" FROM
SQLITE_MASTER WHERE type='table';--"

| type | name | tbl_name | rootpage | sql |
| table | android_metadata | android_metadata | 3 | CREATE TABLE... |
| table | Passwords | Passwords | 4 | CREATE TABLE... |
| table | Key | Key | 5 | CREATE TABLE... |
```

Bir içerik sağlayıcı, temeldeki dosya sistemine erişim sağlayabilir. Bu, uygulamaların dosyaları, Android korumalı alanının aksi halde engelleyeceği yerlerde paylaşmasına olanak tanır.

```
//Dosya sistemindeki dosyaları okumak için
dz> run app.provider.read <URI>

//Dosyadan içerik indirmek için
dz> run app.provider.download <URI>

//Enjeksiyon zafiyetlerini kontrol etmek için
dz> run scanner.provider.injection -a <package_name>

//Dizin geçişi güvenlik açıklarını kontrol etmek için
dz> run scanner.provider.traversal -a <package_name>
```

**5. Hizmetlerle Etkileşim:** Dışa aktarılan hizmetlerle etkileşim için, Drozer'dan aşağıdakileri kullanarak daha fazla ayrıntı sağlamanı isteyebiliriz:

```
//Dışa aktarılan hizmetler hakkında ayrıntılı bilgi almak için
dz> run app.service.info -a <package_name>
```

**6. Gelişmiş Seçenekler:** Daha fazla bilgi almak için harika komutlar da uygulayabiliriz:  
shell.start - Cihazda etkileşimli bir Linux kabuğu başlatın.

“shell.start” - *Cihazda etkileşimli bir Linux kabuğu başlatın.*

“tools.file.upload / tools.file.download” - *Dosyaların Android cihaza / cihazdan kopyalanmasına izin verin.*

“tools.setup.busybox / tools.setup.minimalsu” - *Cihaza yararlı ikili dosyalar yükleyin.*

## OWASP ZAP

- Zed Attack Proxy (ZAP) basit ve kullanımı kolay bir şekilde tasarlanmıştır. Daha önce, güvenlik açıklarını bulmak için yalnızca web uygulamaları için kullanılıyordu, ancak şu anda tüm test uzmanları tarafından mobil uygulama güvenlik testleri için yaygın olarak kullanılmaktadır.

The screenshot shows the OWASP ZAP interface. The top panel displays the HTTP request details for a successful GET request to https://m.facebook.com. The response is a 200 OK status with a content-type of text/html. The bottom panel shows a warning for CSP: script-src unsafe-inline, indicating a security issue with the script-src directive in the response headers.

**HTTP/1.1 200 OK**  
Set-Cookie: datr=70agYl78Shw0NwXvP1HrIUx; expires=Wed, 08-Feb-2023 07:40:28 GMT; Max-Age=63072000; path=/; domain=.facebook.com; secure; httpOnly; SameSite=None  
Set-Cookie: fr=1fzghHsuXoa3k78Z..8gl0rs..iL.AAA.0.0.Bgl0rs.AHXZ2FKDFA; expires=Sun, 09-May-2021 07:40:27 GMT; Max-Age=7759999; path=/; domain=.facebook.com; secure; httpOnly; SameSite=None  
Set-Cookie: sp=70agYl78Shw0NwXvP1HrIUx; expires=Wed, 08-Feb-2023 07:40:28 GMT; Max-Age=63072000; path=/; domain=.facebook.com; secure; httpOnly; SameSite=None  
content-security-policy: default-src \* data: blob: 'self'; script-src \* facebook.com \*.fbcdn.net \*.facebook.net \*.google-analytics.com \*.virtualland.net \*.google.com 127.0.0.1 \* \*.spotlocal.com \*.unsafe-inline \*.unsafe-eval' blob: data: 'self'; style-src data: blob: 'unsafe-inline' \*.facebook.com facebook.com \*.fbcdn.net \*.facebook.net \*.spotlocal.com \*.wss://\*.facebook.com \* https://fb.scandianlocal.com \*.attachment.fbxs.com ws://localhost\* blob: \*.cdninstagram.com \*.self chrome-extension://boadgeojhgndaghljhdicfmlpafd chrome-extension://diochdbjfkbcapmhcmlaeajidm;  
Cache-Control: private, no-cache, no-store, must-revalidate  
X-Frame-Options: DENY  
Content-Type: text/html; charset=utf-8

Cok geniş cevap gövdesi (167.766 bytes) - görüntülemek için görünümü değiştir (Yukarıdaki 'Gövde: Geniş Cevap' yazan açılır menüyü kullanarak).  
Bu mesajın yüklenmesi biraz zaman alabilir.  
Seçenekler/Görünüm üzerinden 'Geniş Cevap' görünümü için kullanılan en düşük mesaj boyutunu değiştirilebiliriz.

**CSP: script-src unsafe-inline**  
URL: https://m.facebook.com/  
Risk: Medium  
Güvenlilik: Medium  
Parametre: content-security-policy  
Saldırı:  
default-src \* data: blob: 'self'; script-src \* facebook.com \*.fbcdn.net \*.facebook.net \*.google-analytics.com \*.virtualland.net \*.google.com 127.0.0.1 \* \*.spotlocal.com \*.unsafe-inline \*.unsafe-eval' blob: data: 'self'; style-src data: blob: 'unsafe-inline' \*.facebook.com facebook.com \*.fbcdn.net \*.facebook.net \*.spotlocal.com \*.wss://\*.facebook.com \* https://fb.scandianlocal.com \*.attachment.fbxs.com ws://localhost\* blob: \*.cdninstagram.com \*.self chrome-extension://boadgeojhgndaghljhdicfmlpafd chrome-extension://diochdbjfkbcapmhcmlaeajidm;  
CWE ID: 16  
WASC ID: 15  
Kaynak: Pasif (10055 - CSP)  
Açıklama:  
script-src includes unsafe-inline.  
Diğer bilgiler:

The screenshot shows the Facebook mobile app login screen. The user is prompted to enter their phone number or email address and password. There is a 'Giriş Yap' (Log In) button and a 'Yeni Hesap Oluştur' (Create New Account) button. The screen also displays language options and a 'Şifreni mi Unuttun?' (Forgot your password?) link.

Android için Facebook uygulamasını indir ve daha hızlı gezin.

facebook

Cep telefonu numarası veya e-posta

Şifre

Giriş Yap

veya

Yeni Hesap Oluştur

Şifreni mi Unuttun?

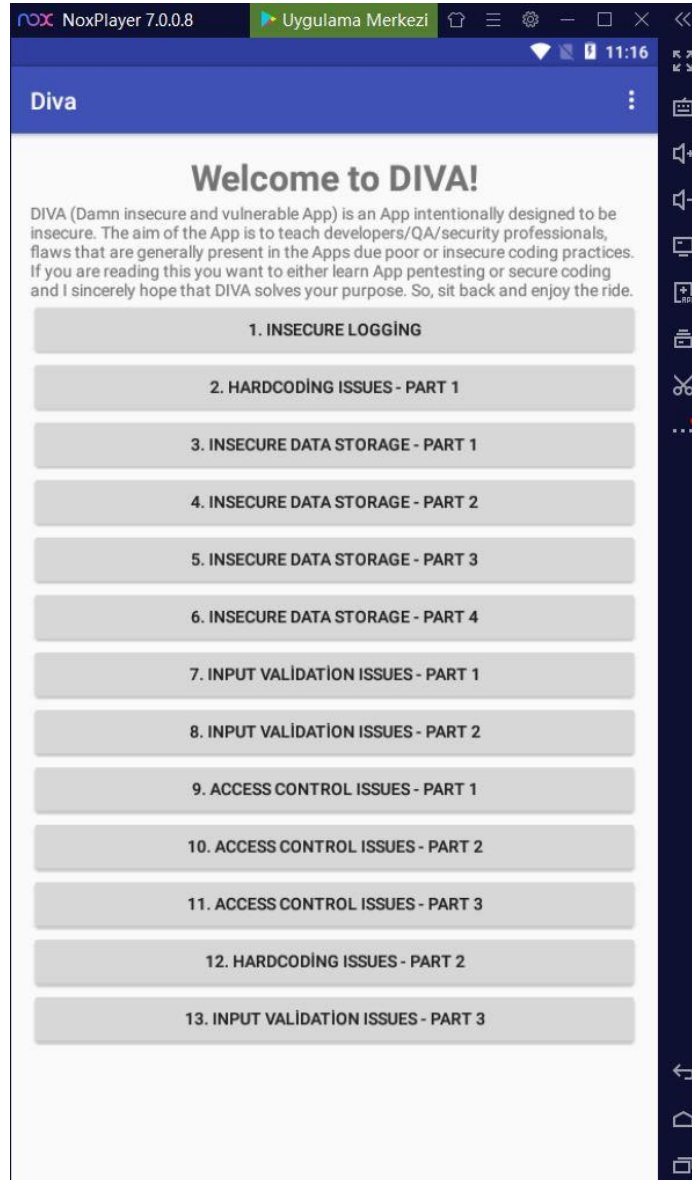
Türkçe العربية Zaza Portuguese (Brasil) Kurdi (Kurmanci) English (UK) Español

Hakkımızda · Yardım · Daha Fazla

Facebook Inc.

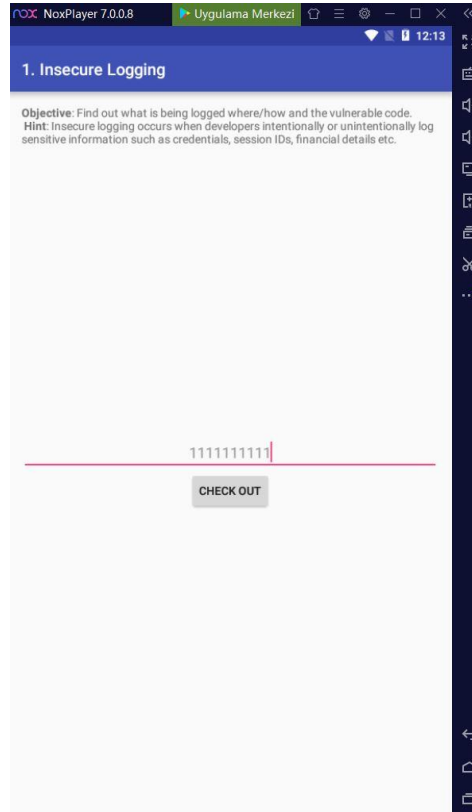
## Mobil Sızma Testi Uygulaması

### DIVA (Damn Insecure and Vulnerable App)

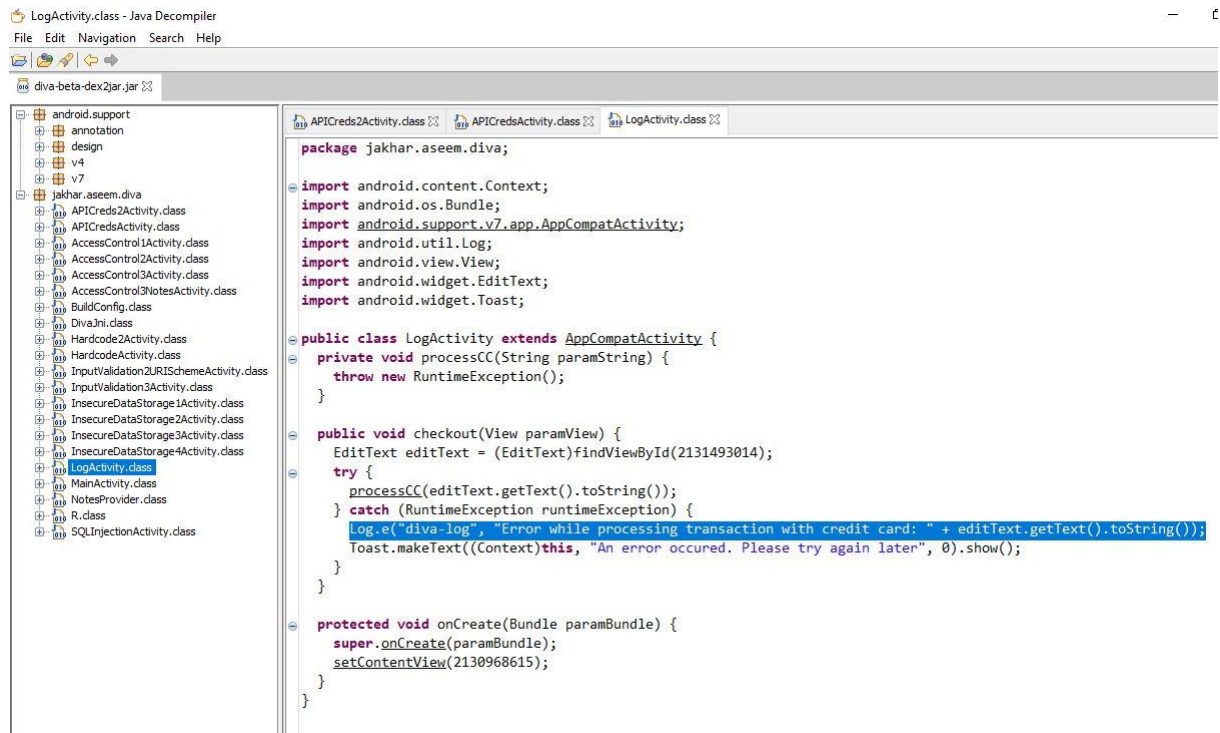


## 1. Diva Insecure Logging

Diva uygulamasının ilk örneği güvensiz log kayıtlarından kaynaklanan bir güvenlik açığıdır. Kullanıcıdan kredi kartı bilgisinin girilmesi istenmektedir.



Fakat kaynak kodda bu işlem yapılırken loglama açık bırakılmış yani yazılımın loglarına düşmekte.



```
LogActivity.class - Java Decompiler
File Edit Navigation Search Help
diva-beta-dex2jar.jar
android.support
  annotation
  design
  v4
  v7
  jakhar.aseem.diva
    APICredits2Activity.class
    APICreditsActivity.class
    AccessControl1Activity.class
    AccessControl2Activity.class
    AccessControl3Activity.class
    AccessControl3NotesActivity.class
    BuildConfig.class
    DivaJni.class
    Hardcode2Activity.class
    HardcodeActivity.class
    InputValidation2URISchemeActivity.class
    InputValidation3Activity.class
    InsecureDataStorage1Activity.class
    InsecureDataStorage2Activity.class
    InsecureDataStorage3Activity.class
    InsecureDataStorage4Activity.class
    LogActivity.class
    MainActivity.class
    NotesProvider.class
    R.class
    SQLInjectionActivity.class
APICredits2Activity.class
APICreditsActivity.class
LogActivity.class
package jakhar.aseem.diva;
import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
public class LogActivity extends AppCompatActivity {
    private void processCC(String paramString) {
        throw new RuntimeException();
    }
    public void checkout(View paramView) {
        EditText editText = (EditText)findViewById(2131493014);
        try {
            processCC(editText.getText().toString());
        } catch (RuntimeException runtimeException) {
            Log.e("diva-log", "Error while processing transaction with credit card: " + editText.getText().toString());
            Toast.makeText((Context)this, "An error occurred. Please try again later", 0).show();
        }
    }
    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2130968615);
    }
}
```

Bu logları, cihazımıza “adb” ile bağlanarak “logcat” ile inceleyebiliriz.

adb shell -> ps | grep ‘diva’ komutu ile cihazda çalışan süreçlerden “diva” uygulamasının pid numarasına ulaşabiliyoruz.

Daha sonra “adb shell -> logcat | grep -i pid\_no” ile logları görüntüleyebilmekteyiz.

```
130|beyond1q:/ #
130|beyond1q:/ #
130|beyond1q:/ # ps | grep diva
u0_a49    3782  1882  930312 95668          c6699cc0 S jakhar.aseem.diva
beyond1q:/ # logcat | grep -i 3782
02-10 11:16:10.611 3782 3782 I art      : Late-enabling -Xcheck:jni
02-10 11:16:10.612 3782 3782 W art      : Unexpected CPU variant for X86 using defaults: x86
02-10 11:16:10.618 2176 2187 I ActivityManager: Start proc 3782:jakhar.aseem.diva/u0a49 for activity jakhar.aseem.diva/.MainActivity
02-10 11:16:10.682 3782 3782 D          : static HostConnection *HostConnection::createUnique(): call
02-10 11:16:10.682 3782 3782 D          : HostConnection::get() New Host Connection established 0xc1d9e0e0, tid 3782
02-10 11:16:10.716 3782 3803 I OpenGLRenderer: Initialized EGL, version 1.4
02-10 11:16:10.716 3782 3803 D OpenGLRenderer: Swap behavior 1
02-10 11:16:10.717 3782 3803 D          : HostConnection::get() New Host Connection established 0xc1d9e260, tid 3803
02-10 11:25:52.491 3782 3803 D OpenGLRenderer: endAllActiveAnimators on 0xa7ae2380 (RippleDrawable) with handle 0xa7ad8430
02-10 11:25:55.681 3782 3782 W IInputConnectionWrapper: finishComposingText on inactive InputConnection
02-10 11:37:40.270 3782 3803 D OpenGLRenderer: endAllActiveAnimators on 0xa7ae2380 (RippleDrawable) with handle 0xa79240e0
02-10 11:40:18.543 3782 3782 E diva-log: Error while processing transaction with credit card: 1111111111
```



## 2. Hardcoding Issues - Part 1

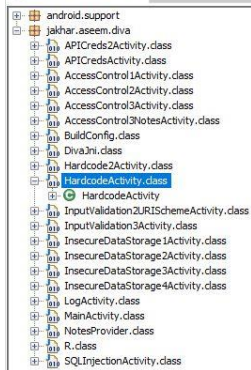
Bu aşamada ise yazılımcı tarafından kaynak kod içerisinde kullanılan sabitlerden kaynaklanan sorunlara değinilmiştir.

Bir input var ve doğru değeri girdimizde “başarılı”, girmedeğimizde ise “yeniden deneyin” tarzında bir hata mesajı veriyor.



Bu kodlamada en temel seviyede “girilen değeri yazılımcı tarafından belirlenen mesaja eşit ise true, değilse false mantığı kullanılmış”. Burada “girilen değeri yazılımcı tarafından belirlenen değere eşit mi?” kısmı kontrol edilirken belirlenen değeri clear-text olarak kaynak kodun içerisinde karşımıza çıkar.

Uygulamayı decompile ettiğimizde kaynak kodundaki değere ulaşabiliriz.

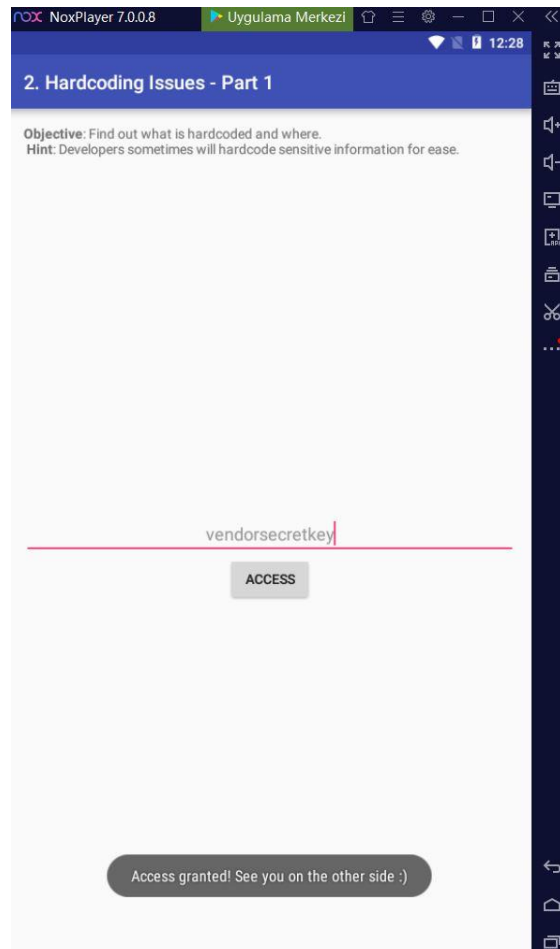


```
package jakhar.aseem.diva;

import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class HardcodeActivity extends AppCompatActivity {
    public void access(View paramView) {
        if (((EditText)findViewById(2131492987)).getText().toString().equals("vendorsecretkey")) {
            Toast.makeText((Context)this, "Access granted! See you on the other side :)", 0).show();
            return;
        }
        Toast.makeText((Context)this, "Access denied! See you in hell :D", 0).show();
    }

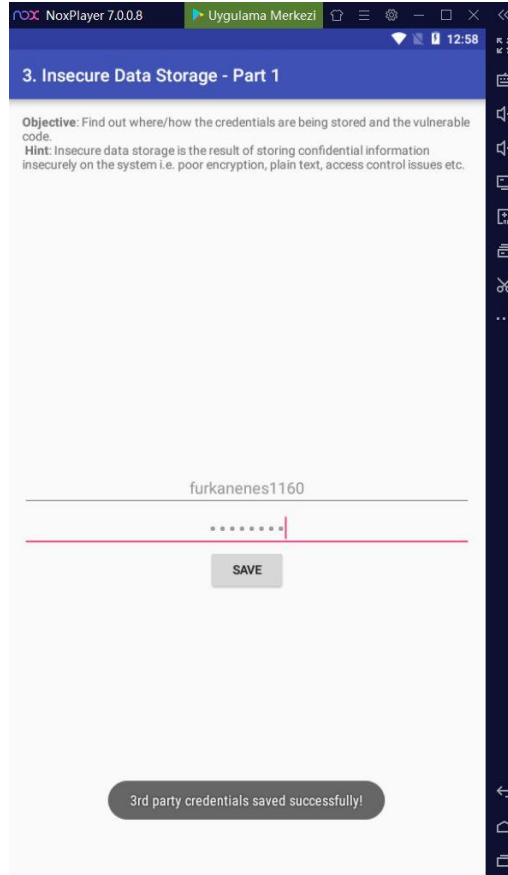
    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2130968607);
    }
}
```



### 3. Insecure Data Storage – Part 1

Bu bölümde güvensiz veri saklamadan kaynaklanan güvenlik açıklarına değinilmiştir.

Karşımızdaki input alanlarına username ve password girmemizi istemekte, girdiğimiz bu değerleri güvenli saklamamasından dolayı saldırganlar tarafından erişilebilir durumdadır.



Girdiğimiz bu verileri yazılımcı

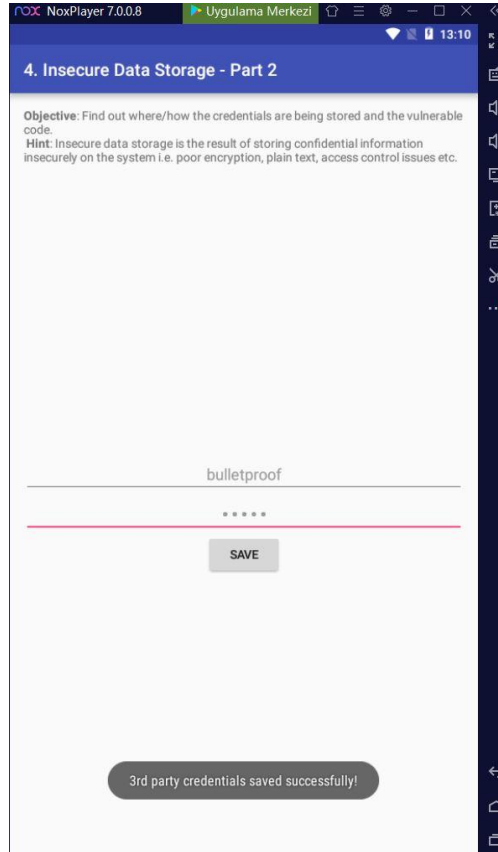
“/data/data/jakhar.aseem.diva/shared\_prefs/jakhar.aseem.diva\_preferences.xml” dosyasında tutmaktadır.

“adb” ile cihazda shell alarak cihaz üzerinde bu dosyaya giderek görüntülediğimizde, girdiğimiz username ve passworda ulaşmaktayız.

```
beyond1q:/data/data # cd jakhar.aseem.diva/
beyond1q:/data/data/jakhar.aseem.diva # ls
cache code_cache databases lib shared_prefs
beyond1q:/data/data/jakhar.aseem.diva # cd shared_prefs/
beyond1q:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva_preferences.xml
beyond1q:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva_preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="user">furkanenes1160</string>
  <string name="password">passw0rd</string>
</map>
beyond1q:/data/data/jakhar.aseem.diva/shared_prefs #
```

## 4. Insecure Data Storage – Part 2

Bu bölümde güvensiz veri saklama yöntemlerinden kaynaklanan bir zafiyet bulunmaktadır. Bu sefer girdiğimiz değerler veritabanına kaydedilir. Fakat kaydediği yer cihazın içinde “/data/data/jakhar.aseem.diva/databases” dizininin altındadır.



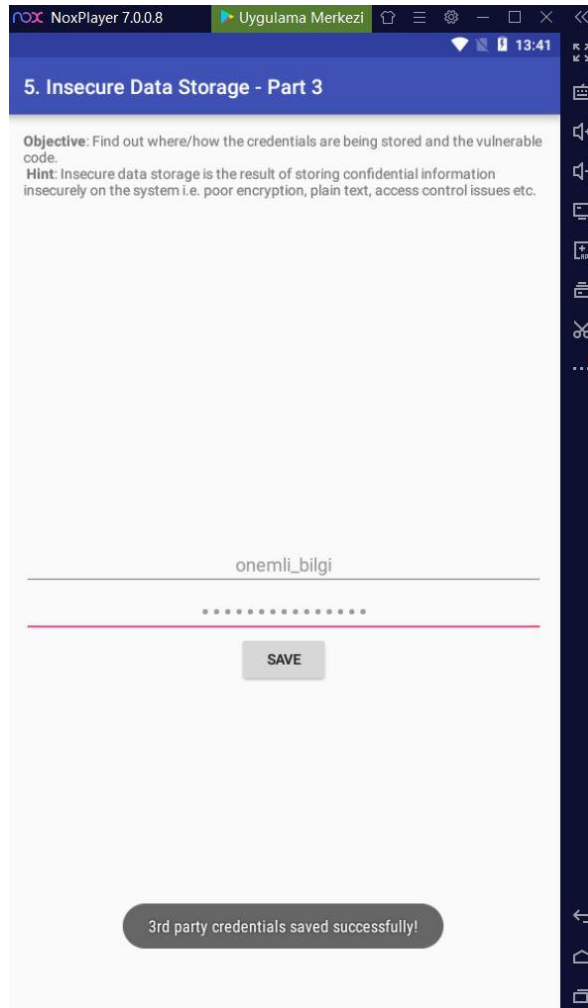
```
Komut İstemi
D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\NoxPlayerEmulator\Nox\bin>adb.exe shell
*7*[r*[999;999H*[6n*8beyond1q:/ #
beyond1q:/ #
beyond1q:/ #
beyond1q:/ # cd data/data/jakhar.aseem.diva/databases/
beyond1q:/data/data/jakhar.aseem.diva/databases # ls -l
total 60
-rw-rw---- 1 u0_a49 u0_a49 20480 2021-02-10 11:16 divanotes.db
-rw----- 1 u0_a49 u0_a49 8720 2021-02-10 11:16 divanotes.db-journal
-rw-rw---- 1 u0_a49 u0_a49 16384 2021-02-10 13:10 ids2
-rw----- 1 u0_a49 u0_a49 8720 2021-02-10 13:10 ids2-journal
beyond1q:/data/data/jakhar.aseem.diva/databases # exit
D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\NoxPlayerEmulator\Nox\bin>adb.exe pull /data/data/jakhar.aseem.diva/databases/ids2
C:\Users\furka\Desktop
[100%] /data/data/jakhar.aseem.diva/databases/ids2
D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\NoxPlayerEmulator\Nox\bin>
```

“ids2” adlı veritabanı dosyasını masaüstüne kaydettikten SQLite’ın kurulu olduğu dizine attıktan sonra SQLite Command Line veya SQLite Browser aracılığıyla veritabanına kaydedilen verileri görebiliriz.

```
D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\sqlite-tools-win32-x86-3340100\sqlite3.exe
SQLite version 3.34.1 2021-01-20 14:10:07
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open ids2
sqlite> .tables
android_metadata  myuser
sqlite> select * from myuser;
bulletproof|12345
sqlite>
```

### 5. Insecure Data Storage – Part 3

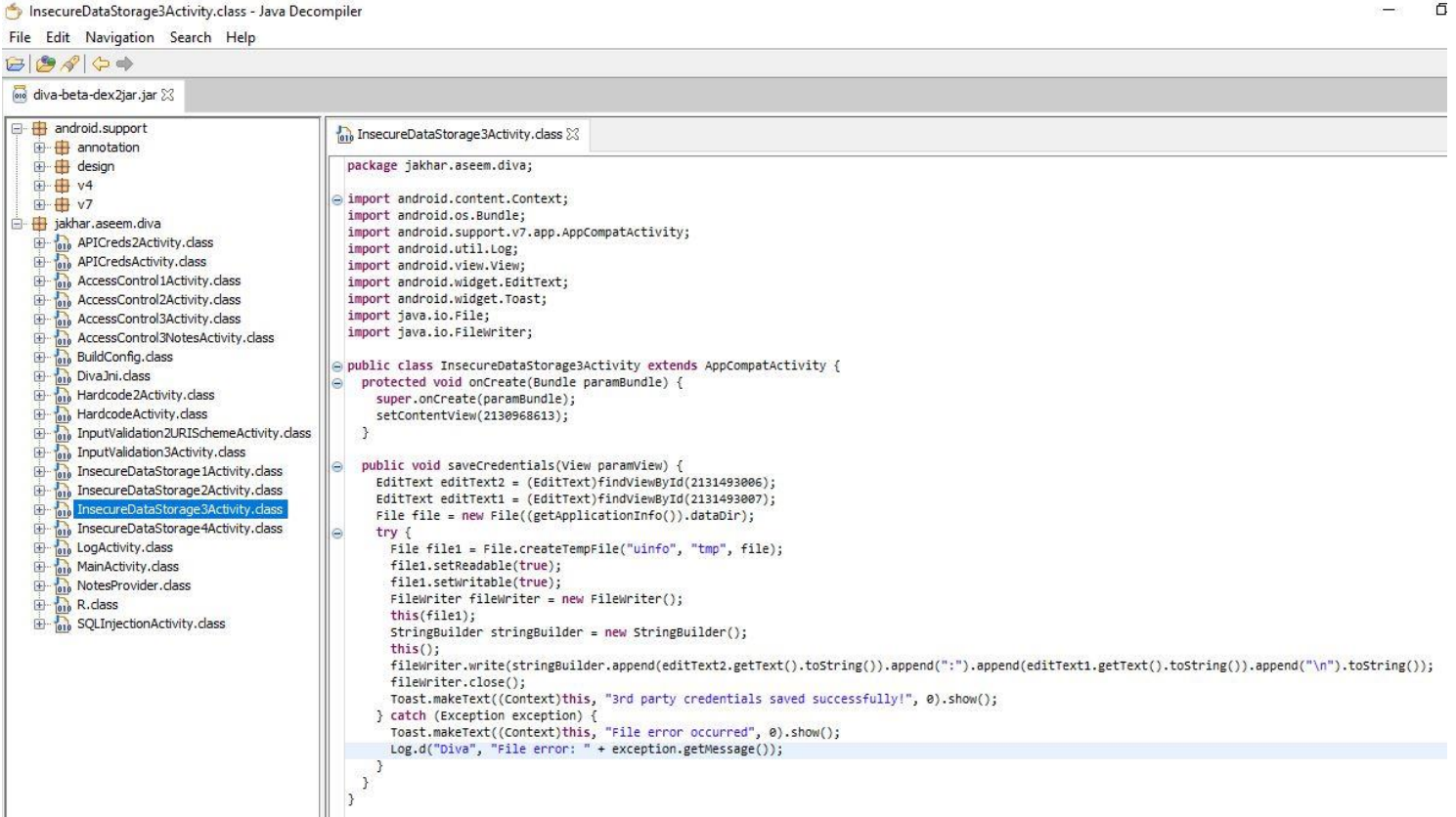
Bu kısımda yine güvensiz veri saklama sorunlarından birine değinilmiştir.



Uygulama, bulunduğu klasöre geçici bir dosya oluşturarak hassas verileri buraya kaydetmektedir. Bu tip tespitler için her zaman uygulamanın klasörü, databases ve shared\_preferences dizinleri incelenmelidir.

Bunun dışında .apk dosyası decompile edilerek kaynak kodu incelenmeli ve mutlaka AndroidManifest.xml dosyasındaki izinler ve diğer bilgiler analiz edilmelidir.

Aşağıdaki resimde görüldüğü gibi kaynak kodda “uinfo” adında geçici bir dosya oluşturulmakta ve alınan değerler bu dosyaya yazılmaktadır.



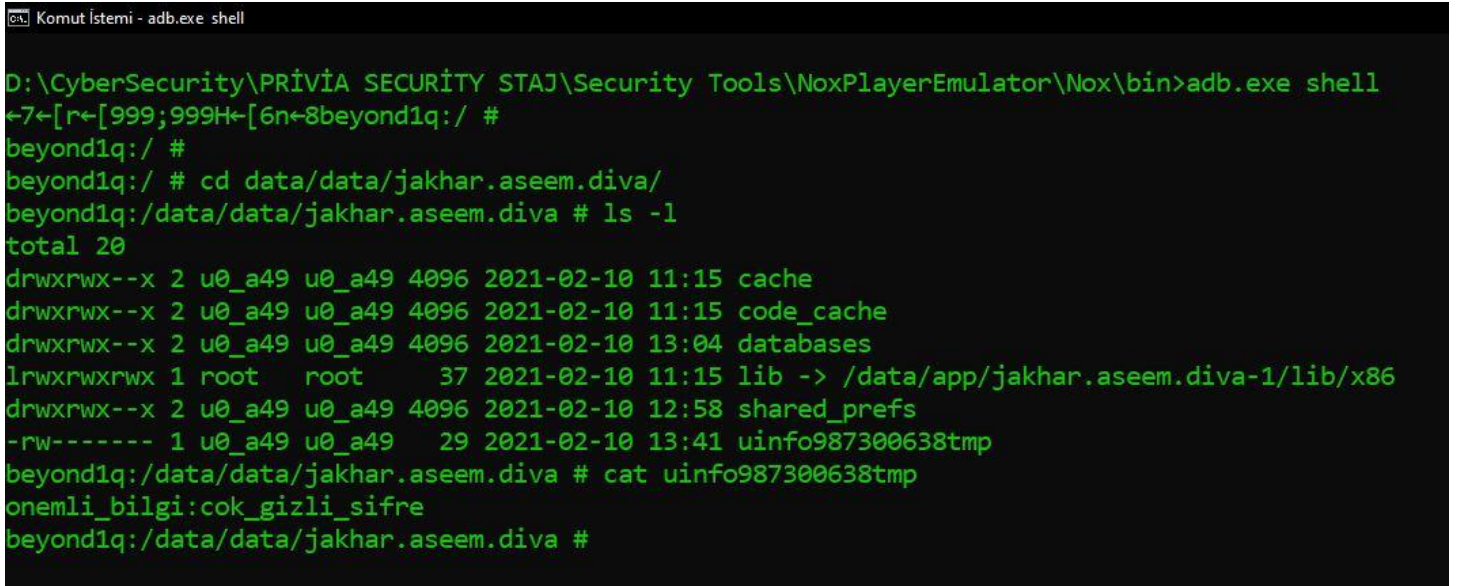
```
package jakhar.aseem.diva;

import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileWriter;

public class InsecureDataStorage3Activity extends AppCompatActivity {
    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2130968613);
    }

    public void saveCredentials(View paramView) {
        EditText editText2 = (EditText)findViewById(2131493006);
        EditText editText1 = (EditText)findViewById(2131493007);
        File file = new File(getApplicationInfo().dataDir);
        try {
            File file1 = File.createTempFile("uinfo", "tmp", file);
            file1.setReadable(true);
            file1.setWritable(true);
            FileWriter filewriter = new FileWriter();
            this(file1);
            StringBuilder stringBuilder = new StringBuilder();
            filewriter.write(stringBuilder.append(editText2.getText().toString()).append(":").append(editText1.getText().toString()).append("\n").toString());
            filewriter.close();
            Toast.makeText((Context)this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception exception) {
            Toast.makeText((Context)this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + exception.getMessage());
        }
    }
}
```

“adb” ile shell alarak, bu dosyayı inceleyecek olursak;

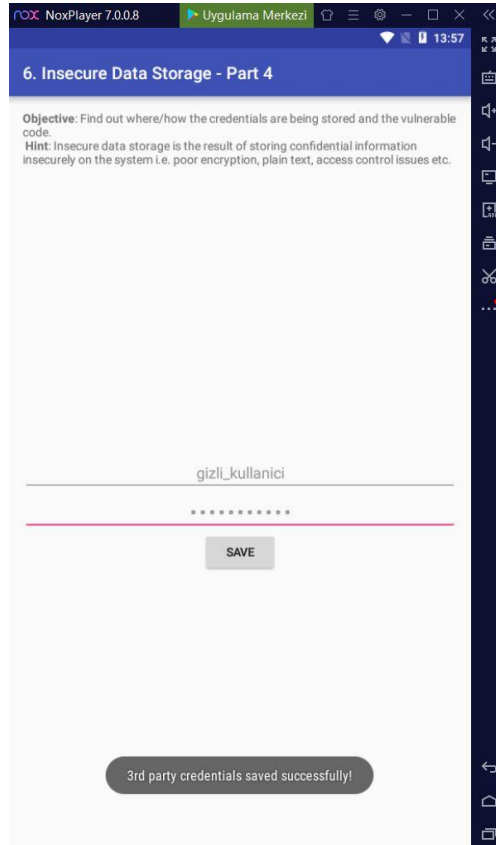


```
Komut İstemi - adb.exe shell

D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\NoxPlayerEmulator\Nox\bin>adb.exe shell
*7*[r←[999;999H←[6n←8beyond1q:/ #
beyond1q:/ #
beyond1q:/ # cd data/data/jakhar.aseem.diva/
beyond1q:/data/data/jakhar.aseem.diva # ls -l
total 20
drwxrwx--x 2 u0_a49 u0_a49 4096 2021-02-10 11:15 cache
drwxrwx--x 2 u0_a49 u0_a49 4096 2021-02-10 11:15 code_cache
drwxrwx--x 2 u0_a49 u0_a49 4096 2021-02-10 13:04 databases
lrwxrwxrwx 1 root root 37 2021-02-10 11:15 lib -> /data/app/jakhar.aseem.diva-1/lib/x86
drwxrwx--x 2 u0_a49 u0_a49 4096 2021-02-10 12:58 shared_prefs
-rw----- 1 u0_a49 u0_a49 29 2021-02-10 13:41 uinfo987300638tmp
beyond1q:/data/data/jakhar.aseem.diva # cat uinfo987300638tmp
onemli_bilgi:cok_gizli_sifre
beyond1q:/data/data/jakhar.aseem.diva #
```

## 6. Insecure Data Storage – Part 4

Bu kısımda yine girilen bilgiler cihaz içerisinde güvensiz şekilde tutulmaktadır. Verilerimiz girip save ediyoruz. Bu sefer SDCard içerisinde bir dosya oluşturup ona kaydetmektedir.



InsecureDataStorage4Activity.class - Java Decompiler

File Edit Navigation Search Help

diva-beta-dex2jar.jar

```
package jakhar.aseem.diva;

import android.content.Context;
import android.os.Bundle;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileWriter;

public class InsecureDataStorage4Activity extends AppCompatActivity {
    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2130968614);
    }

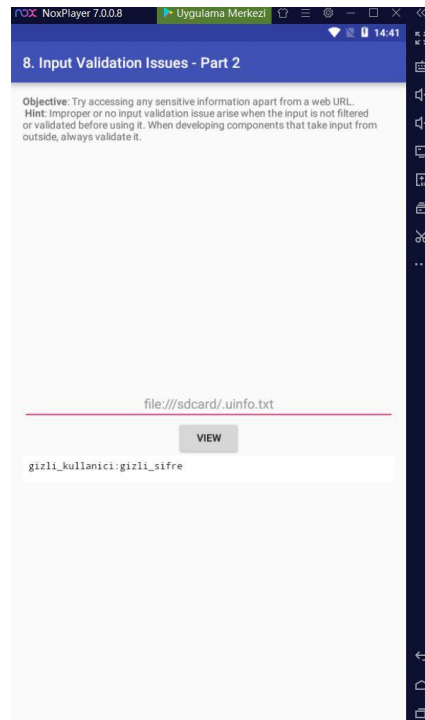
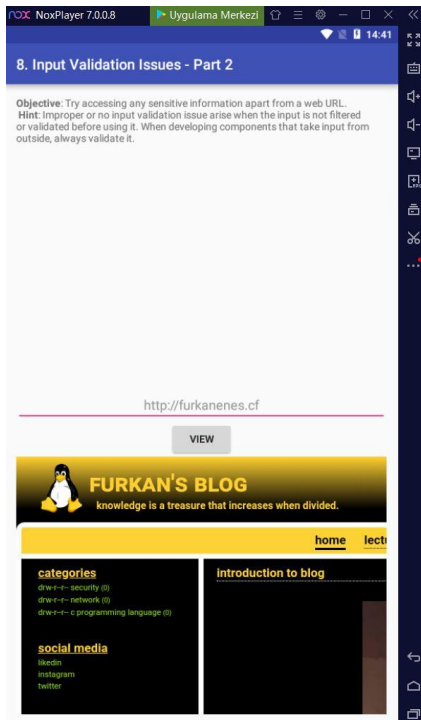
    public void saveCredentials(View paramView) {
        EditText editText2 = (EditText)findViewById(2131493010);
        EditText editText1 = (EditText)findViewById(2131493011);
        File file = Environment.getExternalStorageDirectory();
        try {
            File file1 = new File();
            StringBuilder stringBuilder2 = new StringBuilder();
            this();
            this(stringBuilder2.append(file.getAbsolutePath()).append("/.uinfo.txt").toString());
            file1.setReadable(true);
            file1.setWritable(true);
            FileWriter fileWriter = new FileWriter();
            this(file1);
            StringBuilder stringBuilder1 = new StringBuilder();
            this();
            fileWriter.write(stringBuilder1.append(editText2.getText().toString()).append(":").append(editText1.getText().toString()).append("\n").toString());
            fileWriter.close();
            Toast.makeText((Context)this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception exception) {
            Toast.makeText((Context)this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + exception.getMessage());
        }
    }
}
```

```
Komut İstemi - adb.exe shell

D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\NoxPlayerEmulator\Nox\bin>adb.exe shell
<7<[r<[999;999H<[6n<8beyond1q:/ #
beyond1q:/ #
beyond1q:/ #
beyond1q:/ # cd mnt/sdcard/
beyond1q:/mnt/sdcard # ls -al
total 52
drwxrwx--x 14 root sdcard_rw 4096 2021-02-10 01:17 .
drwx--x--x  4 root sdcard_rw 4096 2020-12-24 16:21 ..
-rw-rw----  1 root sdcard_rw   28 2021-02-10 14:06 .uinfo.txt
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 Alarms
drwxrwx--x  3 root sdcard_rw 4096 2020-12-24 16:21 Android
drwxrwx--x  1 root sdcard_rw    0 2021-02-10 11:15 Apps
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 DCIM
drwxrwx--x  1 root sdcard_rw    0 2021-02-08 10:19 Download
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 Movies
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 Music
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 Notifications
drwxrwx--x  1 root sdcard_rw    0 2021-02-10 01:18 Pictures
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 Podcasts
drwxrwx--x  2 root sdcard_rw 4096 2020-12-24 16:21 Ringtones
drwxrwx--x  2 root sdcard_rw 4096 2021-02-10 01:15 backups
-rw-rw----  1 root sdcard_rw  939 2021-02-07 11:20 cacert.cer
beyond1q:/mnt/sdcard # cat .uinfo.txt
gizli_kullanici:gizli_sifre
beyond1q:/mnt/sdcard #
```

## 8. Input Validation Issues - Part 2

Burada Input kontrol eksikliğinden kaynaklanan bir zafiyete değinilmiştir.





## 11. Access Control Issues – Part 3

Bu kısımda erişim kontrol sorunlarına değinilmiştir.

Kullanıcıdan bir PIN kodu girilmesi istenmektedir.

Girilen PIN kodu shared\_preferences dizinin altında .xml dosyasına yazılmaktadır. Buradan PIN koduna erişebiliriz.



```
Komut İstemi - adb.exe shell
D:\CyberSecurity\PRIVIA SECURITY STAJ\Security Tools\NoxPlayerEmulator\Nox\bin>adb.exe shell
beyond1q:/ #
beyond1q:/ # cd data/data/jakhar.aseem.diva/
beyond1q:/data/data/jakhar.aseem.diva # cd shared_prefs/
beyond1q:/data/data/jakhar.aseem.diva/shared_prefs # ls
WebViewChromiumPrefs.xml jakhar.aseem.diva_preferences.xml
at jakhar.aseem.diva_preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="notespin">0157</string>
  <string name="user">furkanenes1160</string>
  <string name="password">passw0rd</string>
</map>
beyond1q:/data/data/jakhar.aseem.diva/shared_prefs #
```

### 13. Input Validation Issues – Part 3

Burada inputa girilen deęerin uzunluęu kontrol edilmemiřtir.

Örneęin 10 karakter girildięinde herhangi bir sorun vermemektedir ancak 40 adet karakter girildięinde uygulama çökmektedir.

