Ders 3 - Insecure Data Storage > Part 1

Dersin Hedefi

Diva uygulamasındaki “3. Insecure Data Storage - Part 1” seçeneğine tıklayın. Bu seçenek ile açılan kayıt sayfasında üçüncü taraf servise kaydolmak için girilen kullanıcı adı ve parola bilgilerinin nerede ve ne şekilde depolandığını keşfedin. Ardından bu kayıt sayfasında servise kayıt için girilen hesap bilgilerini güvensiz depolayan kod satırlarını belirtin.

Dersin Açıklaması

Android uygulamalarda android uygulama verilerini depolama ikiye ayrılmaktadır: Yerel veri depolama ve uzak veri depolama. Yerel veri depolama birbirinden amaç ve kapsam açısından farklılık arzeden birçok tekniği barındırmaktadır. Uzak veri depolama ise uzakta çalışan bulut veritabanının teknolojisine göre değişiklik gösterir.

Bu “Insecure Data Storage > Part 1-4” (Güvensiz Veri Depolama > Part 1-4) mini serisinde yerel veri depolama yöntemleri arasından birbirinden farklı, fakat diva zafiyetli mobil uygulamasında üçüncü taraf bir servise kayıt olma sayfasındaki girilen bilgileri (kullanıcı adı ve şifre bilgilerini) depolama noktasında ortak işlev görecek dört farklı yöntem üzerinden gidilecektir. Bunlar; “Shared Preferences” (Paylaşımlı Tercihler), “SQLite Database” (SQLite İlişkisel Veritabanı), “Internal File Storage” (Dahili Dosya Depolama) ve “External File Storage” (Harici Dosya Depolama) şeklindedir. Bu yerel depolama yöntemlerine ilave olarak “Saved Instance State”, “Internal Cache Files”, “Realm Database”,... şeklinde listeye eklemeler yapılabilir. Fakat zafiyetli diva uygulamamızı ilgilendirmediğinden başka yöntemlere girilmeyecektir.

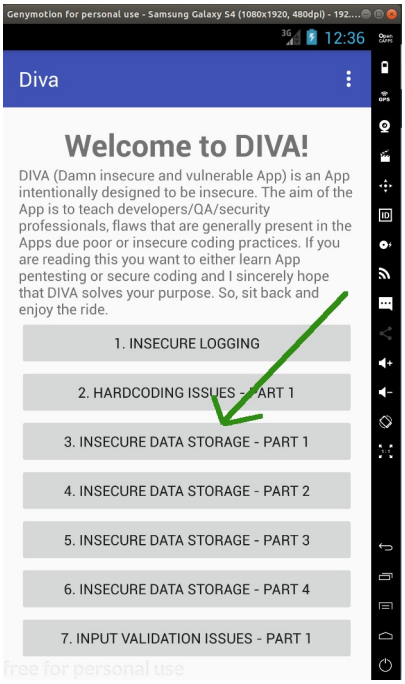
Yerel depolama yöntemi SharedPreferences (Paylaşımlı Tercihler) uygulama ayar / girdi / … tercihlerinin saklanmasında kullanılır. SQLite uygulamadaki kullanıcı taraflı girdilerin / verilerin veritabanı teknolojisi ile depolanmasında kullanılır. Internal File Storage uygulamadaki kullanıcı taraflı girdilerin / verilerin cihazın dahili depolama ünitesinde metin dosyası halinde depolanmasında kullanılır. External File Storage uygulamadaki kullanıcı taraflı girdilerin / verilerin cihazın harici depolama ünitesinde metin dosyası halinde depolanmasında kullanılır.

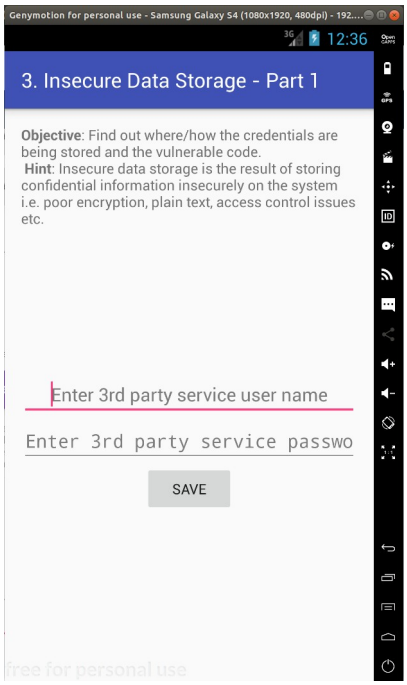
Android cihazlarda yerel veri depolama yöntemleri birbirlerinden **amaç ve kapsam** olarak farklıdırlar. Örneğin bu mini seride bahsedeceğimiz “Shared Preferences” tekniği sadece string, float, integer verileri depolar (çünkü amacı kullanıcı uygulama tercihlerini depolamaktır ve bu nedenle xml formatında veri depolar), resim-ses-video gibi verileri depolamaz, örneğin bu mini seride değinmeyeceğimiz “Saved Instance State” yerel depolama yöntemi sadece uygulama nesnelerinde yapılan değişiklikleri anlık kaydeder ve kazara sayfada geri gitme gibi durum olduğunda mevcut durumun / verilerin korunurluğunu sürdürür, veya bu mini seride değinmeyeceğimiz bir diğer yerel depolama yöntemi “Internal Cache Files” sadece kısıtlı süreli veri tutmaya yarar (çünkü amacı cihaz depolamasını verimli kullanmaktır)... gibi.

Bu güvensiz veri depolama mini (part 1 - 4) serisinde dört farklı yerel depolama yönteminin yanlış şekilde kullanımı nedeniyle kullanıcı hassas verilerinin (diva uygulamasındaki üçüncü taraf servis kullanıcı hesap bilgilerinin) ele geçirilebileceği gösterilecektir.

Dersin Çözümü

Diva uygulamasının “3. Insecure Data storage - Part 1” sayfasında senaryo gereği bizi uygulama içi üçüncü taraf bir servise kaydolma sayfası karşılamaktadır. Bu kayıt işlemi ile diva uygulaması içerisinden üçüncü taraf bir servisin işlevlerini / özelliklerini kullanabilir hale geleceğiz



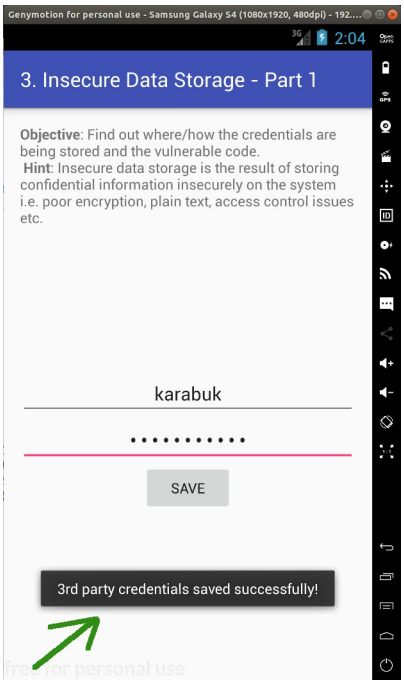


Bu sayfaya gireceğimiz kayıt bilgileri normal şartlarda uzak sunucuda kayıt altına alınacaktır. Fakat ders sayfasındaki senaryoya göre uygulamayı başka zamanlarda kullanırken üçüncü taraf servise her defasında elle giriş yapmayalım diye kayıtlı hesabımızın mobil cihazda yerel olarak depolanması söz konusudur. Bu şekilde diva uygulamasını kullanırken üçüncü taraf servise her

daim bağlı halde kalmış olacağız ve uygulamayı kullanırken her daim üçüncü taraf servisin işlevlerini / özelliklerini kullanabilir halde olacağız.

Bizim amacımız ise yerel android sistemde güvensiz şekilde depolanan üçüncü taraf servis hesabımızın nerede ve ne şekilde depolandığını tespit etmektir.

Şimdi uygulama sayfasında üçüncü taraf servise kayıt olmak için bir kullanıcı adı ve şifre girelim; Kullanıcı adı: karabuk, Şifre: password123



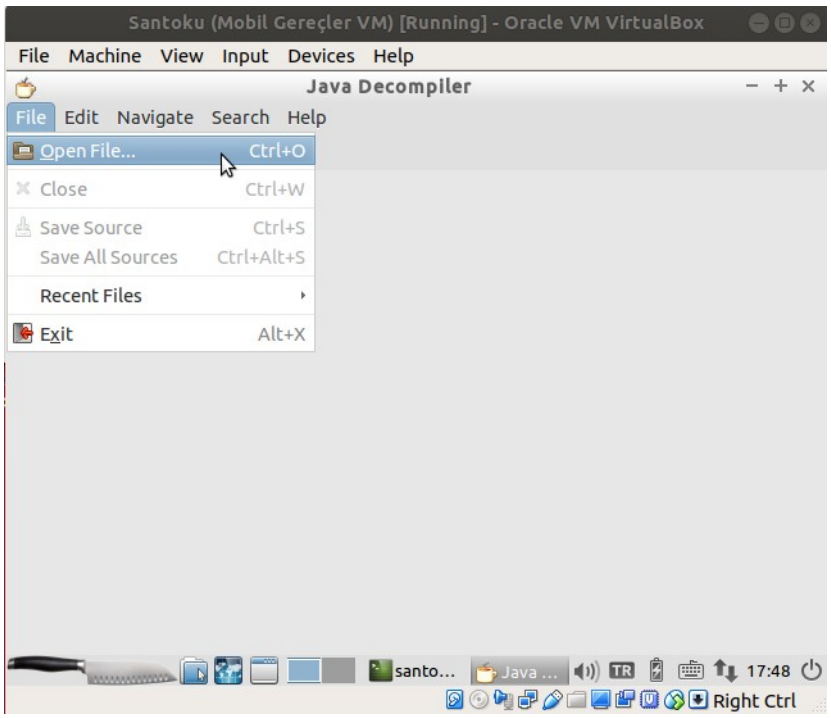
Bu bilgiler uzak sunucuda bir tür veritabanına kaydolur. Yerel mobil cihazda ise sonradan kullanım için bir dosya halinde saklanır / kaydedilir.

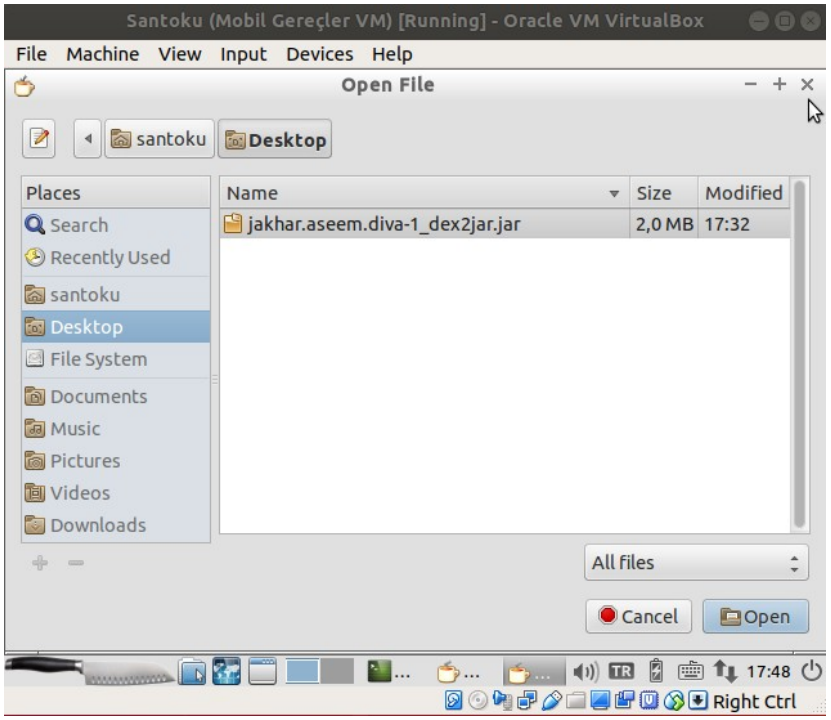
Şimdi bu makale dizisinin en başında yaptığımız uygulama binary dosyasını (.apk dosyasını) bilgisayara çekme, üzerinde tersine mühendislik yapma ve uygulamanın okunabilir java dosyalarını elde etme işini tekrarladığımızı varsayalım. Uygulamanın okunabilir java dosyalarını JD-GUI editörü ile inceleyerek bu uygulama sayfasını (“Insecure Data Storage - Part 1” sayfasını) sunan / kontrol eden java dosyasını tespit edelim. Bahsedilen işlemlerin ayrıntısı için bkz. Başlangıç: Android Uygulamalarda Tersine Mühendislik ile Okunabilir Java Kaynak Kodları Elde Etme (Dex2Jar, JD-GUI, ApkTool)

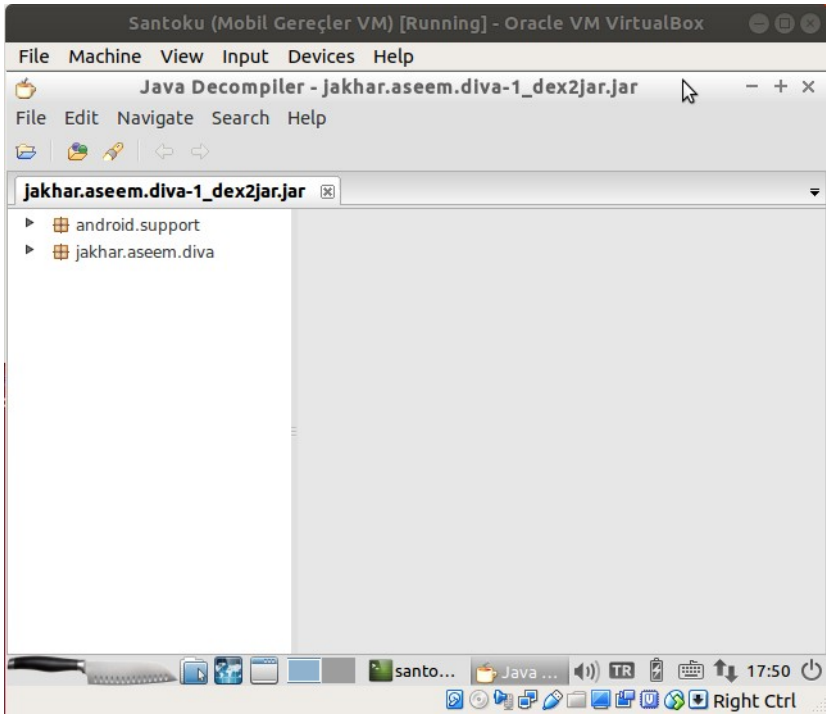
Santoku Linux Terminal:

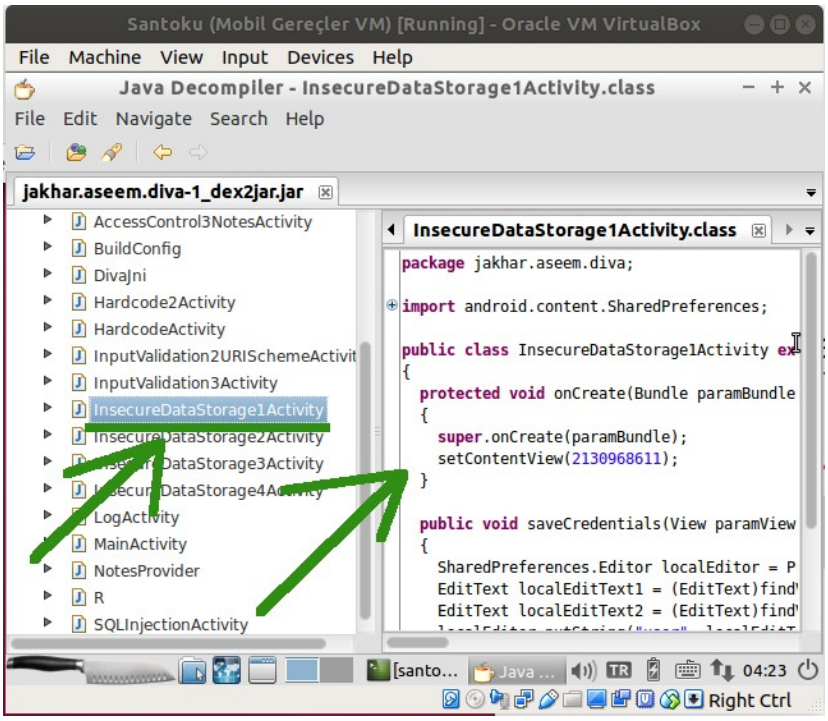
> jd-gui

Çıktı:


Santoku (Mobil Gereçler VM) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Java Decompiler
File Edit Navigate Search Help
Open File... Ctrl+O
Close Ctrl+W
Save Source Ctrl+S
Save All Sources Ctrl+Alt+S
Recent Files
Exit Alt+X
santo...
Java ...
TR
17:48
Right Ctrl


Santoku (Mobil Gereçler VM) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Open File
santoku
Desktop
Places
Search
Recently Used
santoku
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
jakhar.aseem.diva-1_dex2jar.jar
2,0 MB
17:32
All files
Cancel
Open
TR
17:48
Right Ctrl





Java kaynak kod dosyaları incelendiğinde uygulamada görüntülüyor olduğumuz ekranın hangi java dosyası tarafından sunulduğu / kontrol edildiği yukarıdaki gibi görülebilir. Bu örnek için java dosyasını bulmak daha önceki makalelerde bahsedildiği üzere uygulama ekranındaki sayfa ismi ile benzerliği dolayısıyla kolay olmuştur ama gerçek bir senaryoda java dosyaları arasında inceleme yapmak ve doğru java dosyasını bulmak için okumalar yapmak gerekecektir.

Üçüncü taraf servise kaydolduğumuz mobil uygulama sayfasını sunan / kontrol eden java dosyası yukarıdaki gibi InsecureDataStorage1Activity.class'tır. Bu java dosyası incelendiğinde saveCredentials() isimli (yani Türkçe ifadeyle hesap bilgilerini kaydet isimli) bir metot kullanımı görünecektir.

```
public void saveCredentials(View paramView)
{
        SharedPreferences.Editor localEditor =
PreferenceManager.getDefaultSharedPreferences(this).edit();
        EditText localEditText1 = (EditText)findViewById(21314930000);
        EditText localEditText2 = (EditText)findViewById(21314930001);
        localEditor.putString("user", localEditText1.getText().toString());
        localEditor.putString("passwod", localEditText2.getText().toString());
        localEditor.commit();
        Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
}
```

Metot içerisi incelendiğinde uygulama sayfasındaki metin kutuları nesneler halinde, içerdikleri verilerle beraber localEditText1 ve localEditText2'e atanmaktadırlar. Yani bu nesnelerden biri kullanıcı adını, diğeri parolayı kelime halinde tutmaktadır. Sonra en başta oluşturulan "SharedPreferences" nesnesinin yazma metodu putString() ile kullanıcı adı ve parola belirli bir hedef dosyaya yazılmaktadır. Burada bu bilgilerin hangi tür dosyaya / hedefe yazıldığını kullanılan sınıf isminden anlayabiliriz: SharedPreferences. Dolayısıyla kullanıcı hesap bilgilerini saklamada kullanılan yerel depolama yönteminin SharedPreferences olduğunu söyleyebiliriz.

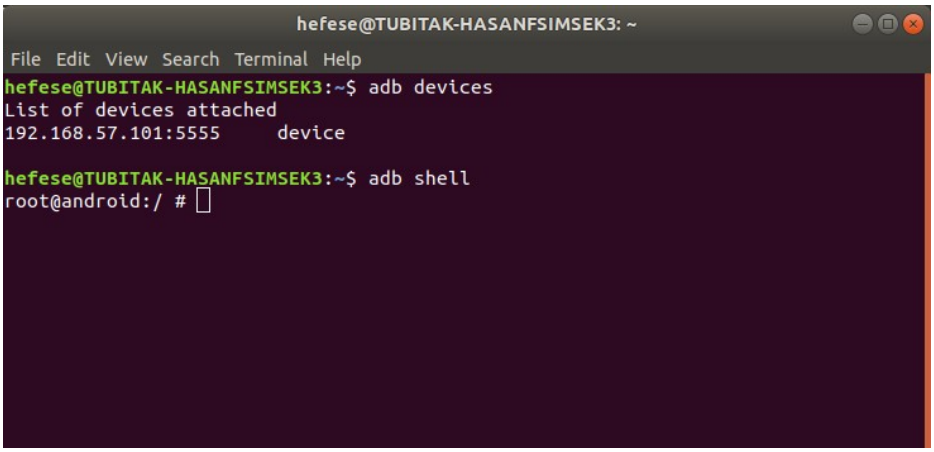
Uygulama java kaynak kodunu inceleyerek kullanıcı hassas verilerinin nerede tutulduğu bilgisine (SharedPreferences dosyalarının bulunduğu konumda olduğu bilgisine) eriştik. Şimdi nasıl tutulduğu bilgisine erişmek için bilgisayardan mobil cihaza bağlanalım ve kullanıcı hassas verilerinin tutulduğu dosya yoluna gidip dosyayı görüntüleyelim.

Öncelikle linux bilgisayardan mobil cihaza resmi Android SDK aracı adb ile bağlanalım ve shell komutu ile mobil cihazın komut satırını alalım.

Ubuntu 18.04 LTS Linux Terminal:

```
> adb devices
> adb shell
```

Çıktı:

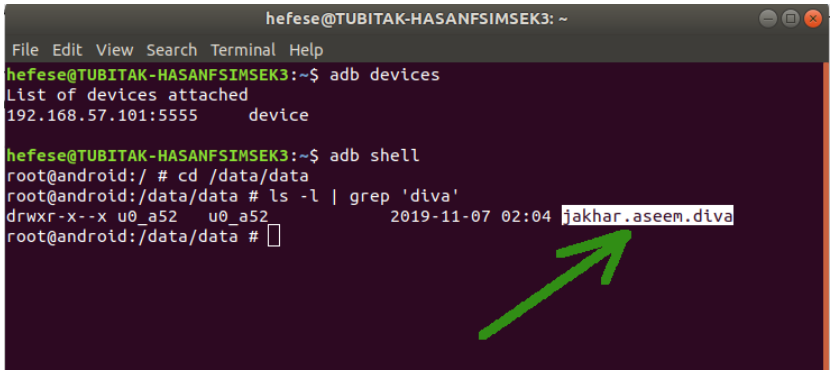


Ardından yapılacak şey mobil cihazdaki diva uygulamasının paket ismini öğrenmektir, sonra android sistemdeki /data/data dizini altında öğrendiğimiz paket ismindeki dizinin içerisine girmektir. Bu şekilde uygulama paket ismindeki klasör altında SharedPreferences klasörü ekrana yansıyacaktır ve klasörün içerisinde ise kullanıcı hesap verilerinin tutulduğu dosya listelenecektir.

> Bilgi:
>
> Android sistemlerde kurulumla beraber gelen apk uygulamalar /system/data dizini altında tutulurken android sistem kurulumu sonrası kullanıcının yüklediği apk uygulamalar ise /data/data dizini altında tutulurlar.

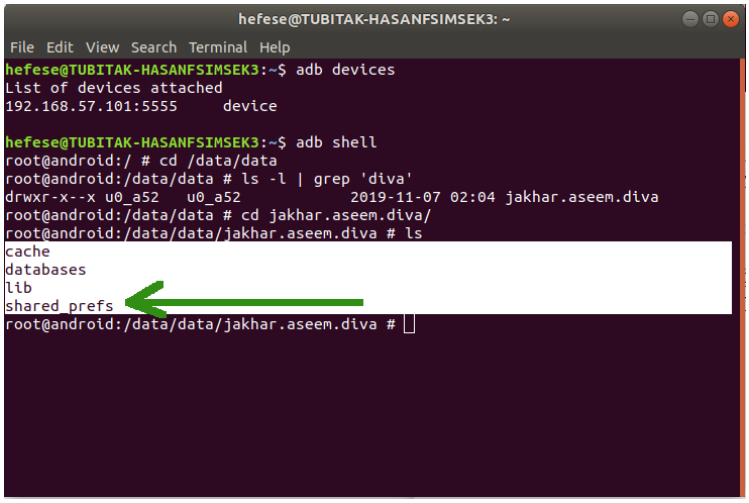
Linux bilgisayardan mobil cihazla olan komut satırı oturumuzda diva uygulamasının android sistemdeki paket ismini tespit edelim.

> Ubuntu 18.04 LTS Terminal:
>
> root@android:/ # cd /data/data/
> root@android:/data/data # ls -l | grep 'diva'
>
> Çıktı:



Ardından /data/data dizini altındaki diva ugulamasının android sistemdeki paket isminde olan klasöre girelim ve SharedPreferences klasörünü görüntüleyelim.

> Ubuntu 18.04 LTS Terminal:
>
> root@android:/data/data # cd jakhar.aseem.diva/
> root@android:/data/data/jakhar.aseem.diva # ls
>
> Çıktı:

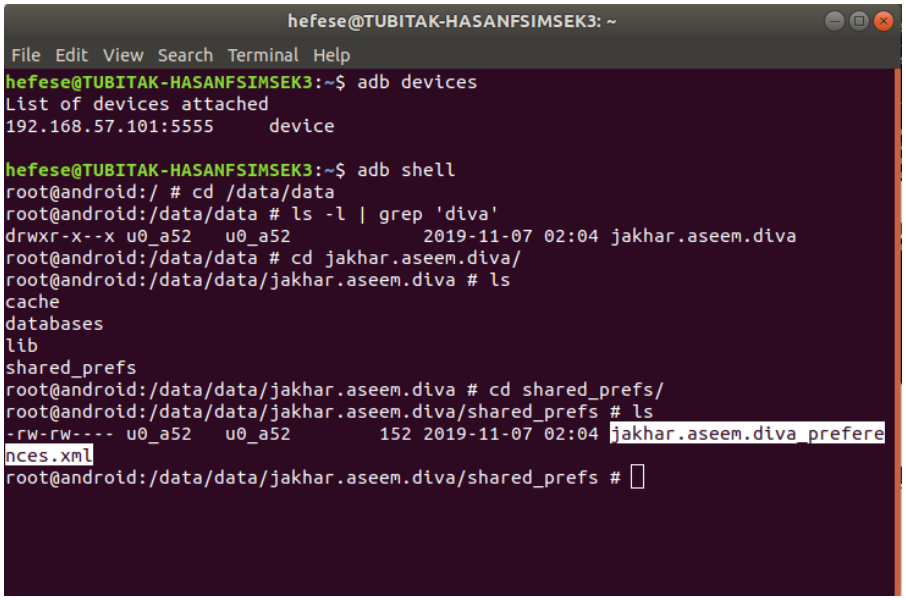


Son olarak shared_prefs klasörü içerisinde kullanıcı hesap bilgilerinin saklandığı xml dosyasını görüntüleyelim.

Ubuntu 18.04 LTS Terminal:

root@android:/data/data/jakhar.aseem.diva # cd shared_prefs/
root@android:/data/data/jakhar.aseem.diva/shared_prefs # ls

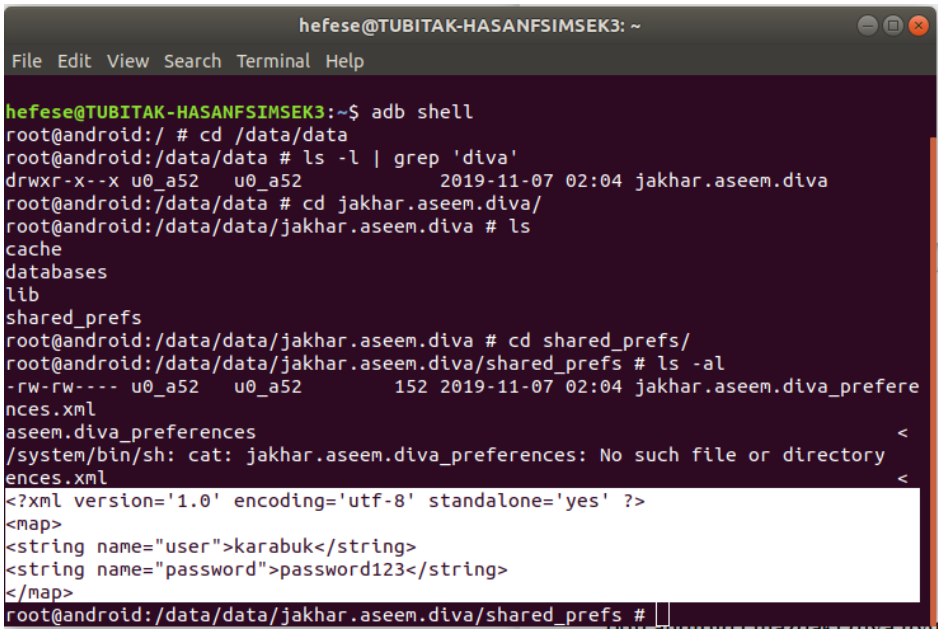
Çıktı:



Ubuntu 18.04 LTS Terminal:

root@android:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva_preferences.xml

Çıktı:



Görüldüğü üzere üçüncü taraf servis hesap bilgileri mobil cihazda yerel olarak açık bir şekilde (güvensiz bir şekilde) depolanmıştır. Bu mobil cihaza veya başka söz gelimi diva uygulamasını kullanan mobil cihaza sızmayı başaran saldırganlar diva uygulaması klasörü içerisinde kritik bir veri elde edebilir miyim diye inceleme yaptıklarında shared_prefs klasörü onlara kullanıcının diva uygulamasındaki bağlı olduğu üçüncü taraf servis hesabının bilgilerini verecektir.

Şimdi uygulama sayfasındaki kullanıcı hesap bilgilerini kaydetme metodundaki güvensiz veri depolayan kod satırlarını gösterelim ve başlığı bitirelim.

```
public void saveCredentials(View paramView)
{
        SharedPreferences.Editor localEditor =
PreferenceManager.getDefaultSharedPreferences(this).edit();
        EditText localEditText1 = (EditText)findViewById(21314930000);
        EditText localEditText2 = (EditText)findViewById(21314930001);
        localEditor.putString("user", localEditText1.getText().toString());
        localEditor.putString("passwod", localEditText2.getText().toString());
        localEditor.commit();
        Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
}
```

İki satırda da görüldüğü üzere uygulama sayfasındaki metin kutularından gelen veriler olduğu gibi SharedPreferences dosyasına yazdırılmaktadır. Bu iki satır normalde bir çeşit şifreleme metoduyla metin kutusu verilerini şifreleyip o şekilde verileri dosyaya yazdırmalıydı.

>> Güvensiz Veri Depolama Kod Satırları

```
localEditor.putString("user", localEditText1.getText().toString());
localEditor.putString("passwod", localEditText2.getText().toString());
```

Bunun yerine direk olduğu gibi metin kutusundan gelen kullanıcı adı ve şifre bilgileri dosyaya yazdırıldığından SharedPreferences yerel depolama yöntemi kullanıcı adı ve şifreyi ifşa eder durumda kalmıştır.

Sonuç

Diva uygulamasının Insecure Data Storage - Part 1 sayfasındaki girilen üçüncü taraf servise kaydolma hesap bilgilerinin nerede ve ne şekilde kaydolduğu bilgisi tersine mühendislik ile elde edilen uygulama kaynak kodlarının incelenmesiyle tespit edilmiştir. Nerede olduğu bilgisi kaynak kodlarda kullanılan SharedPreferences java sınıfı isminden, ne şekilde tutulduğu bilgisi ise (yani şifreli mi, yoksa açık metin halinde mi bilgisi) android sistemdeki diva uygulaması klasörü altında yer alan shared_prefs klasörü içerisindeki xml dosyası görüntülenerek anlaşılmıştır.

Bu derste vurgulanmak istenen açıklık diva uygulamasındaki kullanıcı servis hesap bilgilerinin yerel diskte dosya içerisinde açık metin halinde / şifrelenmeden tutulmasıdır. Kullanıcı servis hesap bilgileri bu şekilde yerel diskte tutulduğu için mobil cihaza ilerde sızacak bir saldırgan kullanıcının hassas bu verilerini alabilir durumdadır.

Örneğin uygulama / sistemsel bir açıklıkla mobil cihaza sızan ve bu derste gösterilen şifrelenmemiş üçüncü taraf servis hesap bilgilerini elde eden bir saldırgan kullanıcı adına, o yapıyormuşçasına diva uygulamasındaki servisin sunduğu olanakları kullanabilir. Onun adına servisin sunduğu çeşit çeşit faaliyetleri yerine getirebilir ve servisin sunduğu yetkiler ölçüsünde kullanıcıya zarar verebilir. Daha farklı açıdan yaklaşacak olursak örneğin bu derste gösterilen şifrelenmemiş üçüncü taraf servis hesap bilgilerini elde eden bir saldırgan aynı hesap bilgileriyle başka sık kullanılan / popüler mobil veya web site uygulamalarına giriş yapmayı deneyebilir. Çünkü kullanıcıların çoğunluğu aynı hesap bilgileriyle birçok uygulamaya kayıt olmaktadırlar.

Sonuç olarak mobil cihazlardaki android uygulamaların yerel diskte kullanıcı hassas verilerini şifrelemeden tutmaları kullanıcının mobil cihazına ilerde sızacak saldırganlara karşı kaptırma imkanı doğurur. Bu nedenle mobil cihazlardaki android uygulamalarda yerel olarak depolanacak kullanıcı verileri hassas niteliğindeyse şifreli olarak yerel sistemde tutulmalıdırlar.

Kaynak

https://www.loginworks.com/blogs/top-10-ways-know-store-data-android/
https://stackoverflow.com/questions/16691437/when-are-java-temporary-files-deleted
https://source.android.com/devices/tech/connect/third-party-call-apps
https://developer.android.com/guide/topics/data/data-storage
https://developer.android.com/training/data-storage/shared-preferences
https://www.tutorialspoint.com/android/android_shared_preferences.htm