Cybersecurity Blog

Everything about threat intelligence, blue team, red team, pentesting, security audit, security review, testing and assessment.

Friday, March 3, 2017

Android Application Backup Vulnerabiility Testing

You must be already knowing about android application backup process. Beauty of this vulnerability is it works on non-rooted devices too sometimes. The vulnerability lies within the AndroidManifest.xml file.

Today we are going to test DIVA (Damn Insecure Vulnerable Application) against this vulnerability. First I had diva-beta.apk file. I unzip that using below command:

```
unzip diva-beta.apk
```

I got files and folders as listed in below screenshot. Then I tried to inspect AndroidManifest.xml file which was showing junk data. In order to see that in clear-text, I moved to apktool.

Translate Language

Search

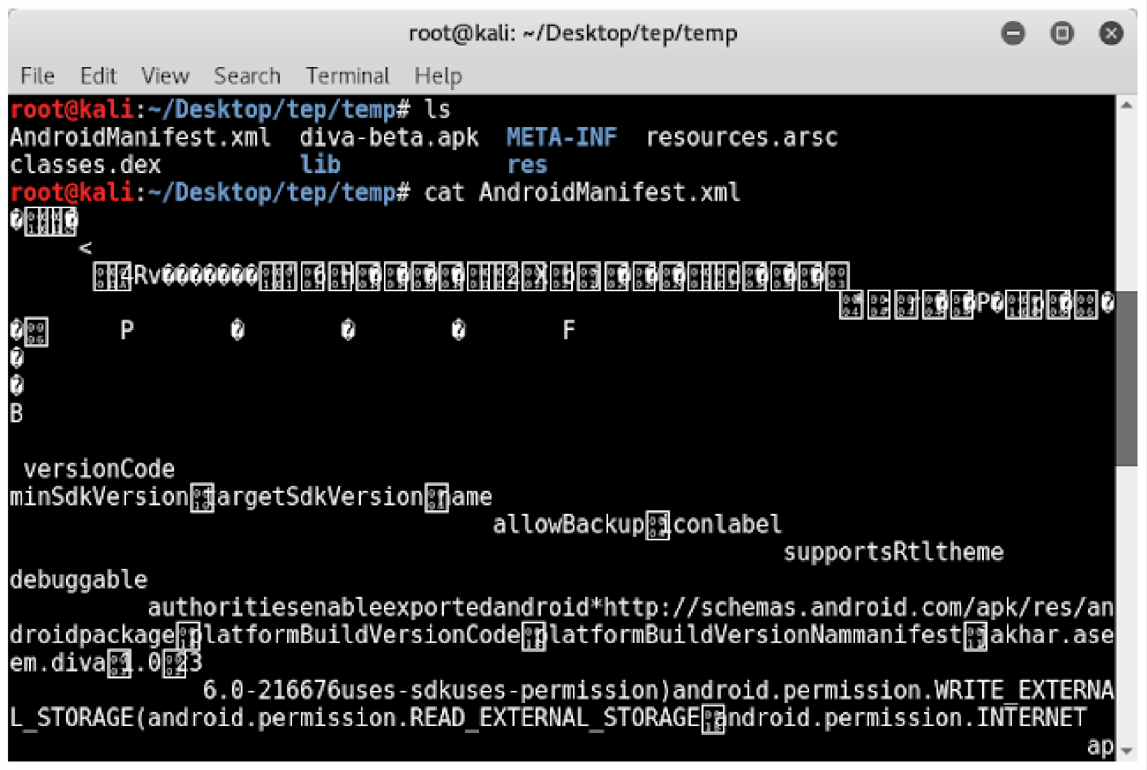
Search

Subscribe via email

Email address... Submit

Blog Archive

- ▶ 2020 (2)
- ▶ 2019 (6)
- ▶ 2018 (4)
- ▼ 2017 (5)
  - ▶ September 10 (1)
  - ▶ April 30 (1)
  - ▶ April 2 (1)
  - ▶ March 26 (1)
  - ▼ February 26 (1)
    - Android Application Backup Vulnerabiility Testing
- ▶ 2016 (11)
- ▶ 2015 (4)
- ▶ 2014 (22)



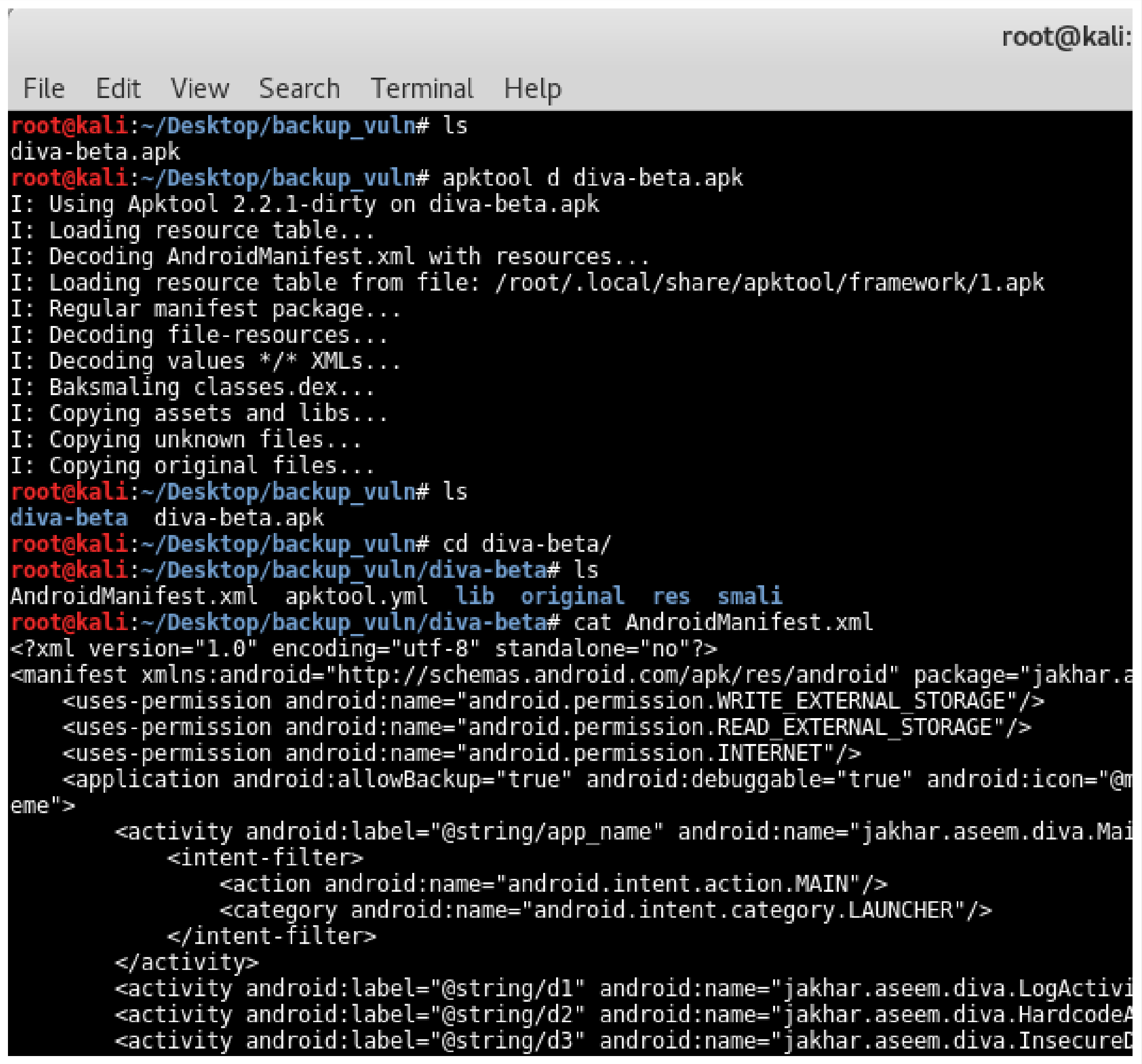
root@kali: ~/Desktop/tep/temp
File Edit View Search Terminal Help

```
root@kali:~/Desktop/tep/temp# ls
AndroidManifest.xml  diva-beta.apk  META-INF  resources.arsc
classes.dex          lib            res
root@kali:~/Desktop/tep/temp# cat AndroidManifest.xml
�����
     <
      ��4Rv�������������H�������2X�����������������
                                                            ����������P�������
��      P       �       �       �       F
�
�
B


 versionCode
minSdkVersion�argetSdkVersion�ame
                                   allowBackup�conlabel
                                                     supportsRtltheme
debuggable
          authoritiesenableexportedandroid*http://schemas.android.com/apk/res/an
droidpackage�latformBuildVersionCode�latformBuildVersionNammanifest�akhar.ase
em.diva�.0�3
              6.0-216676uses-sdkuses-permission)android.permission.WRITE_EXTERNA
L_STORAGE(android.permission.READ_EXTERNAL_STORAGE�android.permission.INTERNET
                                                                              ap
```

I used below command to decompile diva-beta.apk file.

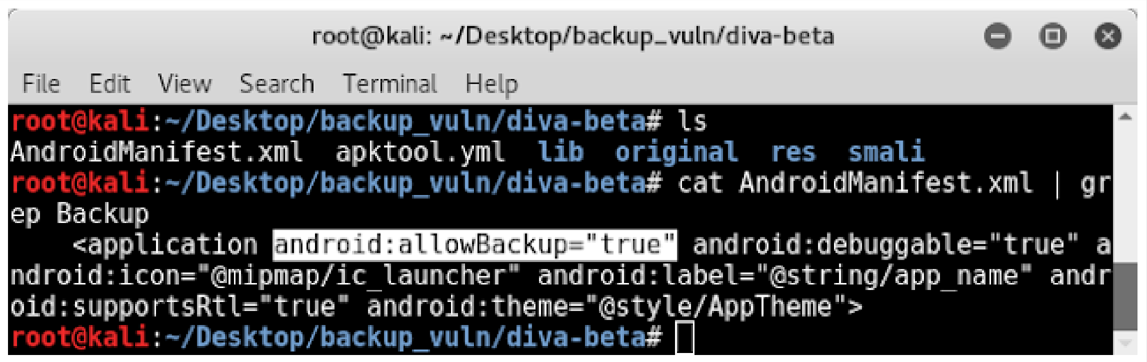
apktool d diva-beta.apk



```
root@kali:~/Desktop/backup_vuln# ls
diva-beta.apk
root@kali:~/Desktop/backup_vuln# apktool d diva-beta.apk
I: Using Apktool 2.2.1-dirty on diva-beta.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
root@kali:~/Desktop/backup_vuln# ls
diva-beta  diva-beta.apk
root@kali:~/Desktop/backup_vuln# cd diva-beta/
root@kali:~/Desktop/backup_vuln/diva-beta# ls
AndroidManifest.xml  apktool.yml  lib  original  res  smali
root@kali:~/Desktop/backup_vuln/diva-beta# cat AndroidManifest.xml
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="jakhar.a
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <application android:allowBackup="true" android:debuggable="true" android:icon="@m
eme">
        <activity android:label="@string/app_name" android:name="jakhar.aseem.diva.Mai
            <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.LogActivi
        <activity android:label="@string/d2" android:name="jakhar.aseem.diva.HardcodeA
        <activity android:label="@string/d3" android:name="jakhar.aseem.diva.InsecureD
```

From above screenshot it can be observed that now AndroidManifest.xml file is in cleartext. To check if application allows to take backup or not, lets inspect the file using grep. I will straightaway look for backup keyword using grep. Command for the same is as mentioned below:

cat AndroidManifest.xml | grep Backup



root@kali: ~/Desktop/backup_vuln/diva-beta
File Edit View Search Terminal Help

```
root@kali:~/Desktop/backup_vuln/diva-beta# ls
AndroidManifest.xml  apktool.yml  lib  original  res  smali
root@kali:~/Desktop/backup_vuln/diva-beta# cat AndroidManifest.xml | gr
ep Backup
    <application android:allowBackup="true" android:debuggable="true" a
ndroid:icon="@mipmap/ic_launcher" android:label="@string/app_name" andr
oid:supportsRtl="true" android:theme="@style/AppTheme">
root@kali:~/Desktop/backup_vuln/diva-beta# 
```

From the above screenshot you can observe that application allows backup. Now this is **by default** behavior of any application. Developers need to explicitly set this value to false in order to prevent this issue.

Our information gathering part is over here. Lets move to quickly exploit this issue. In order to exploit this issue, you need to first take backup shown below screenshot. Command for the same as as follow:

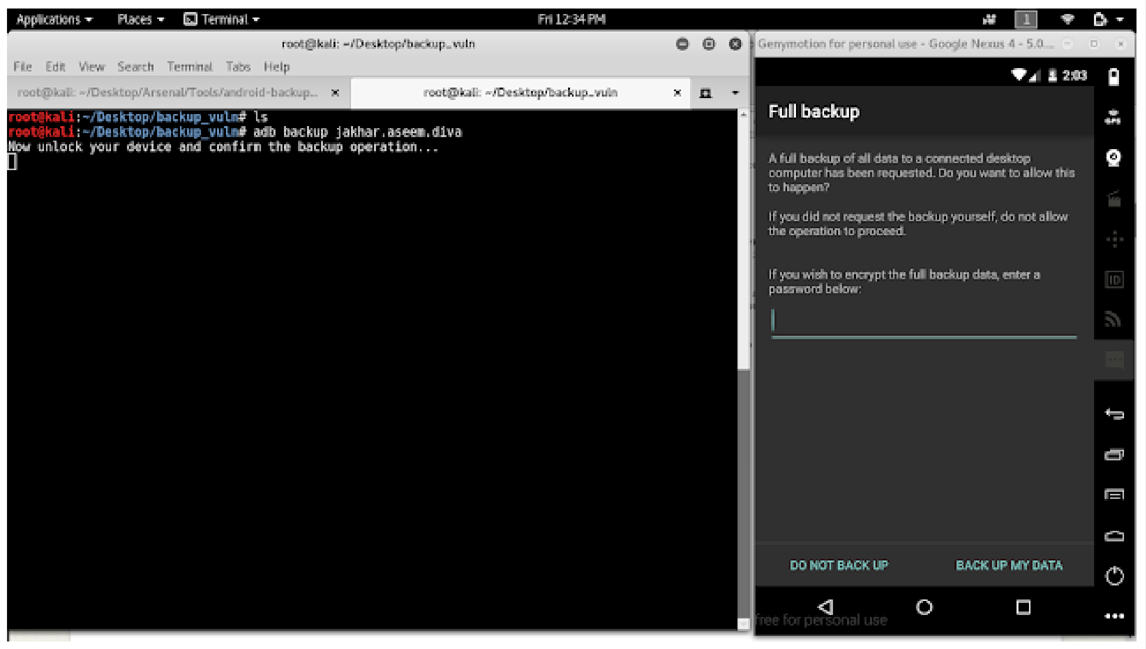
adb backup jakhar.aseem.diva

- adb (Android Debug Bridge)
- backup (Your output file name)
- Jakhar.aseem.diva (Target application name)

In case if you are confuse how to setup android pentest environment using genymotion and all required tool, you can visit this guide.

Setup pentest environment - https://www.youtube.com/watch?v=gwF3qxYxRFM

Setting up and using adb - https://www.youtube.com/watch?v=NPYCpbMoWkQ



Applications
Places
Terminal
Fri 12:34 PM
root@kali: ~/Desktop/backup_vuln
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/Arsenal/Tools/android-backup...
root@kali: ~/Desktop/backup_vuln
Genymotion for personal use - Google Nexus 4 - 5.0...
2:03
free for personal use

```
root@kali:~/Desktop/backup_vuln# ls
root@kali:~/Desktop/backup_vuln# adb backup jakhar.aseem.diva
Now unlock your device and confirm the backup operation...
```

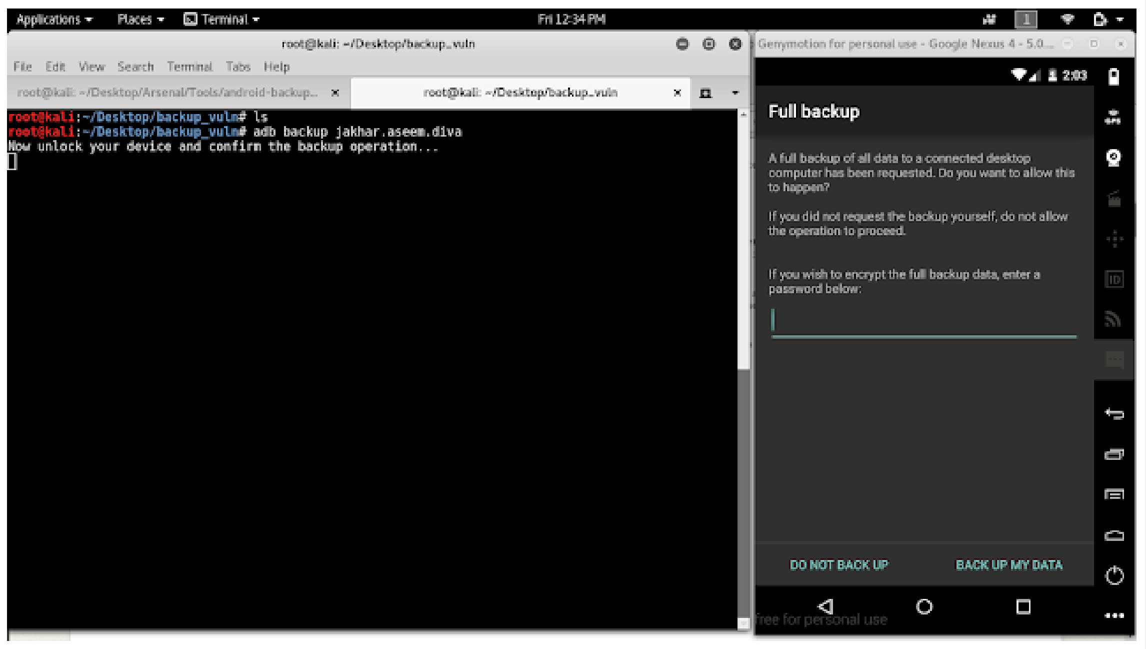
Full backup

A full backup of all data to a connected desktop computer has been requested. Do you want to allow this to happen?

If you did not request the backup yourself, do not allow the operation to proceed.

If you wish to encrypt the full backup data, enter a password below:

As shown in screenshot below first it will ask you to create a backup using password so called as encrypted backup. As hackers we always do not want this. So simply click on 'BACK UP MY DATA' button to start the process.



Applications
Places
Terminal
Fri 12:34 PM
root@kali: ~/Desktop/backup_vuln
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/Arsenal/Tools/android-backup...
root@kali: ~/Desktop/backup_vuln
Genymotion for personal use - Google Nexus 4 - 5.0...
2:03
free for personal use

```
root@kali:~/Desktop/backup_vuln# ls
root@kali:~/Desktop/backup_vuln# adb backup jakhar.aseem.diva
Now unlock your device and confirm the backup operation...
```

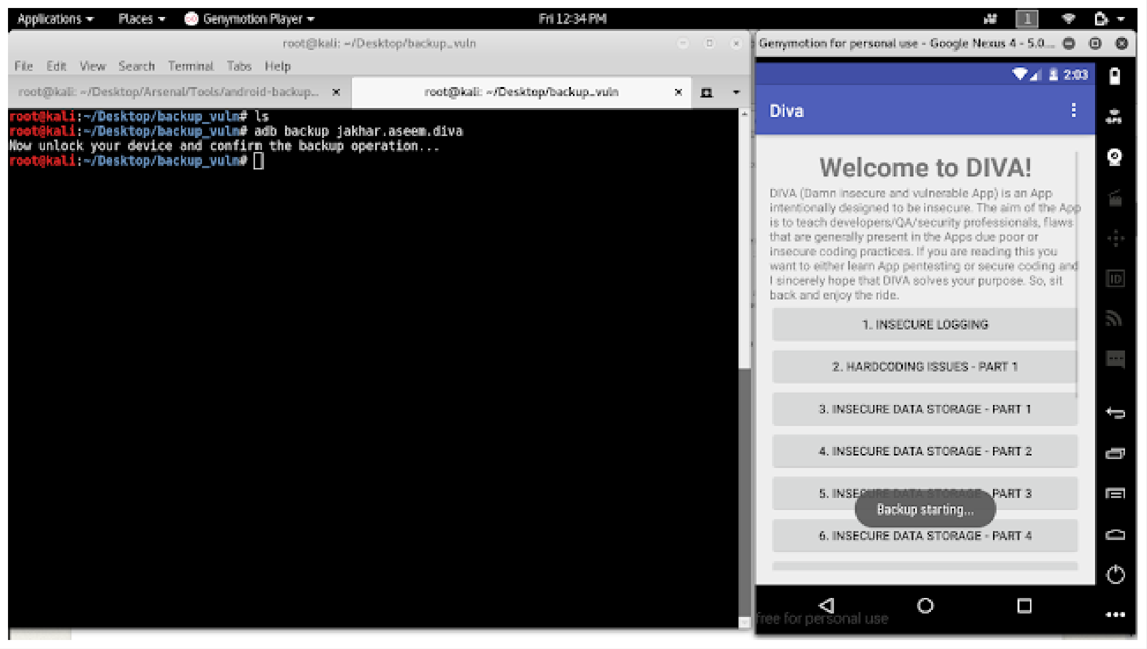
Full backup

A full backup of all data to a connected desktop computer has been requested. Do you want to allow this to happen?

If you did not request the backup yourself, do not allow the operation to proceed.

If you wish to encrypt the full backup data, enter a password below:

Backup in progress...



```
root@kali:~/Desktop/backup_vuln# ls
root@kali:~/Desktop/backup_vuln# adb backup jakhar.aseem.diva
Now unlock your device and confirm the backup operation...
root@kali:~/Desktop/backup_vuln# 
```

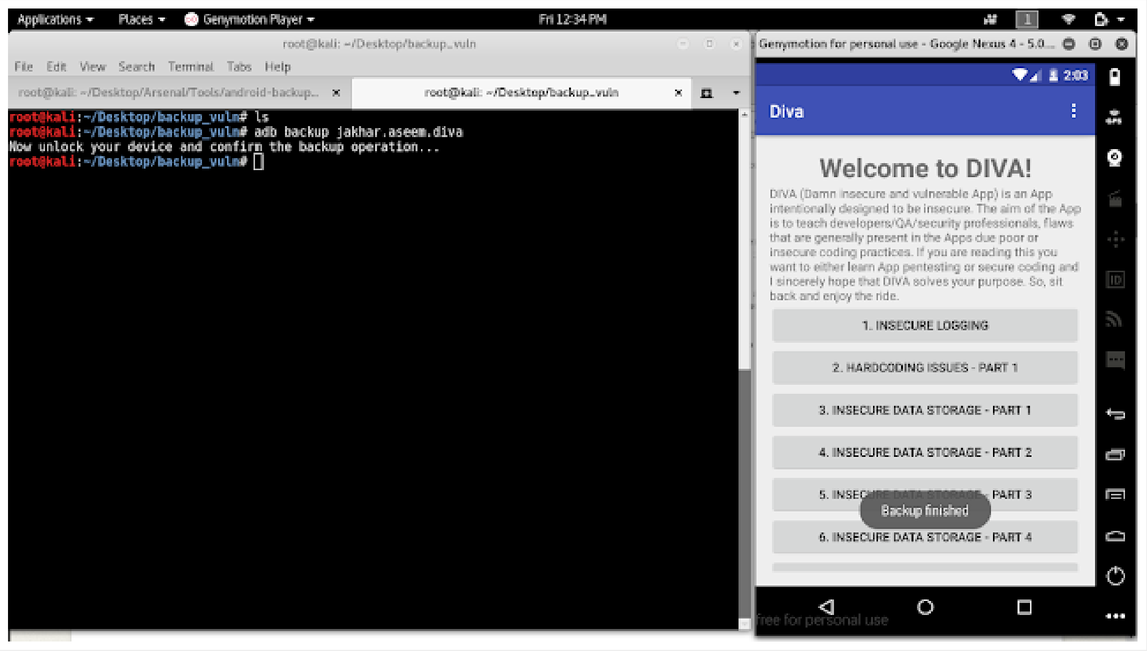
Welcome to DIVA!

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.

Backup starting...

Applications
Places
Genymotion Player
Fri 12:34 PM
root@kali: ~/Desktop/backup_vuln
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/Arsenal/Tools/android-backup...
root@kali: ~/Desktop/backup_vuln
Genymotion for personal use - Google Nexus 4 - 5.0...
2:03
Diva
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
free for personal use

Backup finished...


Applications
Places
Genymotion Player
Fri 12:34 PM
root@kali: ~/Desktop/backup_vuln
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/Arsenal/Tools/android-backup...
root@kali: ~/Desktop/backup_vuln
Genymotion for personal use - Google Nexus 4 - 5.0...
2:03
Diva
free for personal use

```
root@kali:~/Desktop/backup_vuln# ls
root@kali:~/Desktop/backup_vuln# adb backup jakhar.aseem.diva
Now unlock your device and confirm the backup operation...
root@kali:~/Desktop/backup_vuln# 
```

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

Backup finished

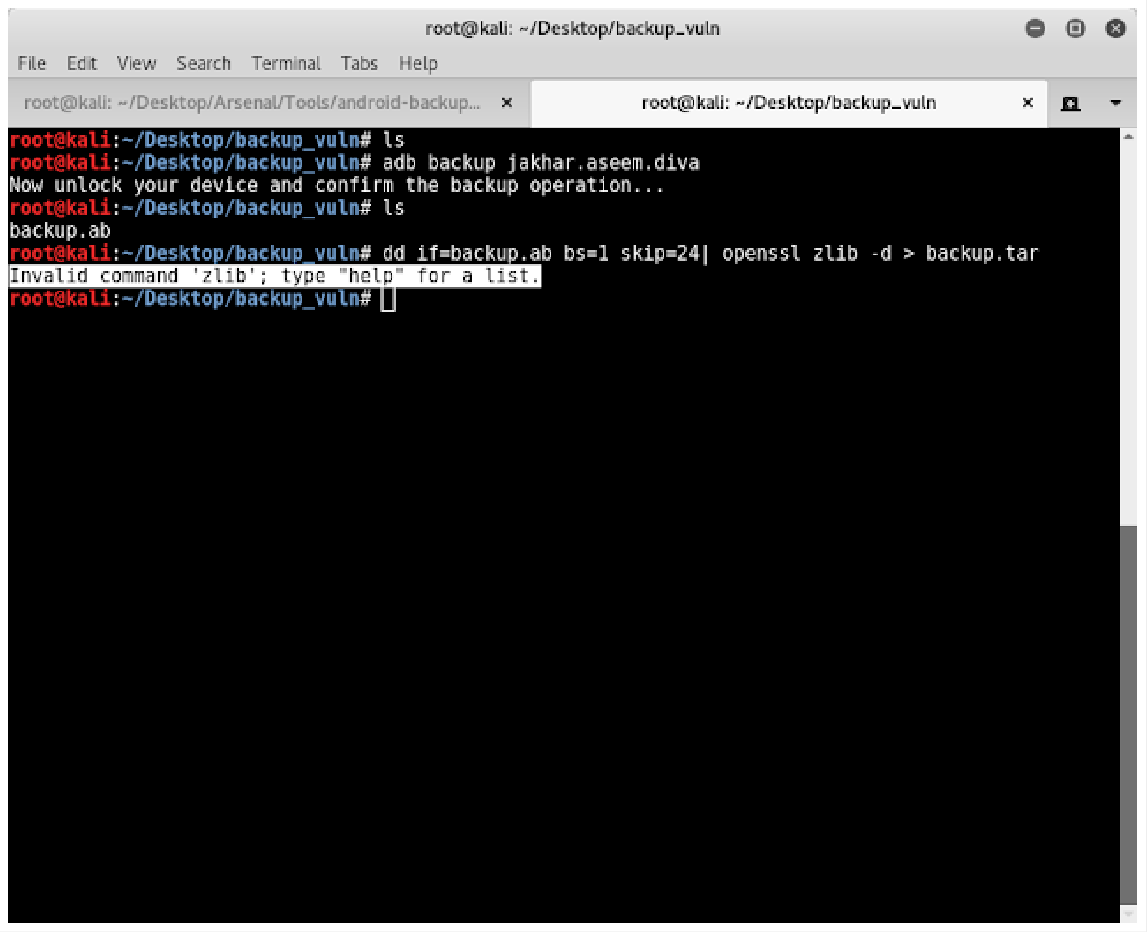
From here onwards its been tricky for beginners. Now as you can see that 'backup.ab' file has been created.

Now if you follow online well known references mentioned below, then they will ask you to convert your .ab file into extractable tar file using below command:

dd if=mybackup.ab bs=24 skip=1| openssl zlib -d > mybackup.tar

Reference 1 - http://resources.infosecinstitute.com/android-hacking-security-part-15-hacking-android-apps-using-backup-techniques/
Reference 2 - http://nelenkov.blogspot.in/2012/06/unpacking-android-backups.html

However, due to issues with OpenSSL zlib library, you may get error message as shown in below screenshot.



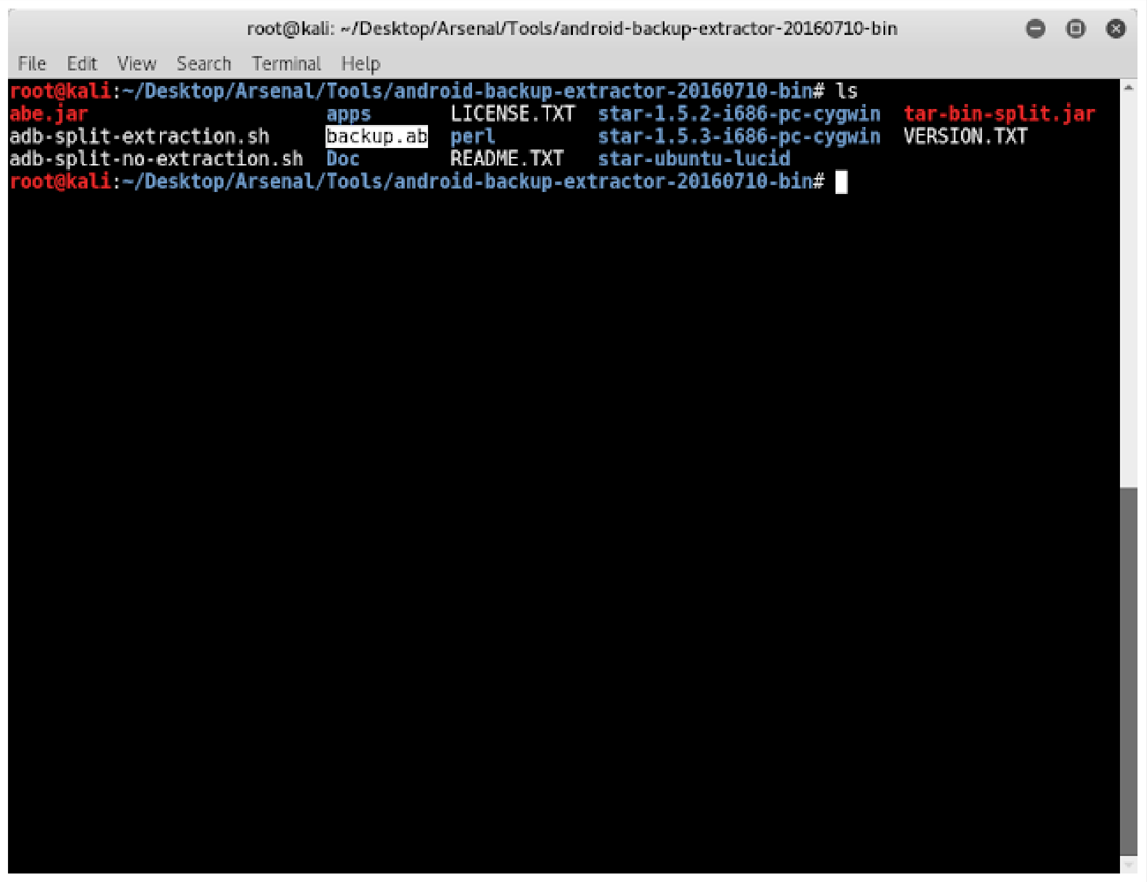
root@kali: ~/Desktop/backup_vuln
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/Arsenal/Tools/android-backup...
root@kali: ~/Desktop/backup_vuln

```
root@kali:~/Desktop/backup_vuln# ls
root@kali:~/Desktop/backup_vuln# adb backup jakhar.aseem.diva
Now unlock your device and confirm the backup operation...
root@kali:~/Desktop/backup_vuln# ls
backup.ab
root@kali:~/Desktop/backup_vuln# dd if=backup.ab bs=1 skip=24| openssl zlib -d > backup.tar
Invalid command 'zlib'; type "help" for a list.
root@kali:~/Desktop/backup_vuln# 
```

As a hacker your obivious try will be to copy paste this error and search online if there is any solution or not. You will find a solution to not to use default dd and openssl commands. Rather than that you can use android backup extractor tool from below referenced link:

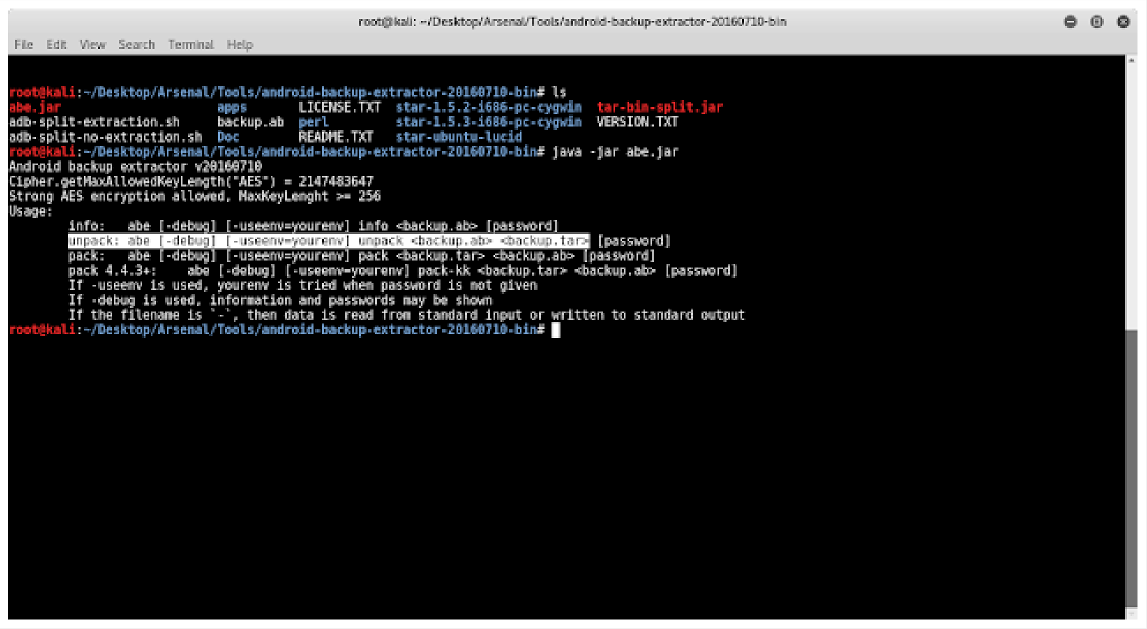
Reference: https://sourceforge.net/projects/adbextractor/

I have downloaded and extracted the same in my box. Also copied backup.ab file in the same folder as mentioned below:


root@kali: ~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin
File Edit View Search Terminal Help

```
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# ls
abe.jar                    apps       LICENSE.TXT  star-1.5.2-i686-pc-cygwin  tar-bin-split.jar
adb-split-extraction.sh    backup.ab  perl         star-1.5.3-i686-pc-cygwin  VERSION.TXT
adb-split-no-extraction.sh Doc        README.TXT   star-ubuntu-lucid
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# 
```

Out of all usage options we will use below highlighted option in our case:

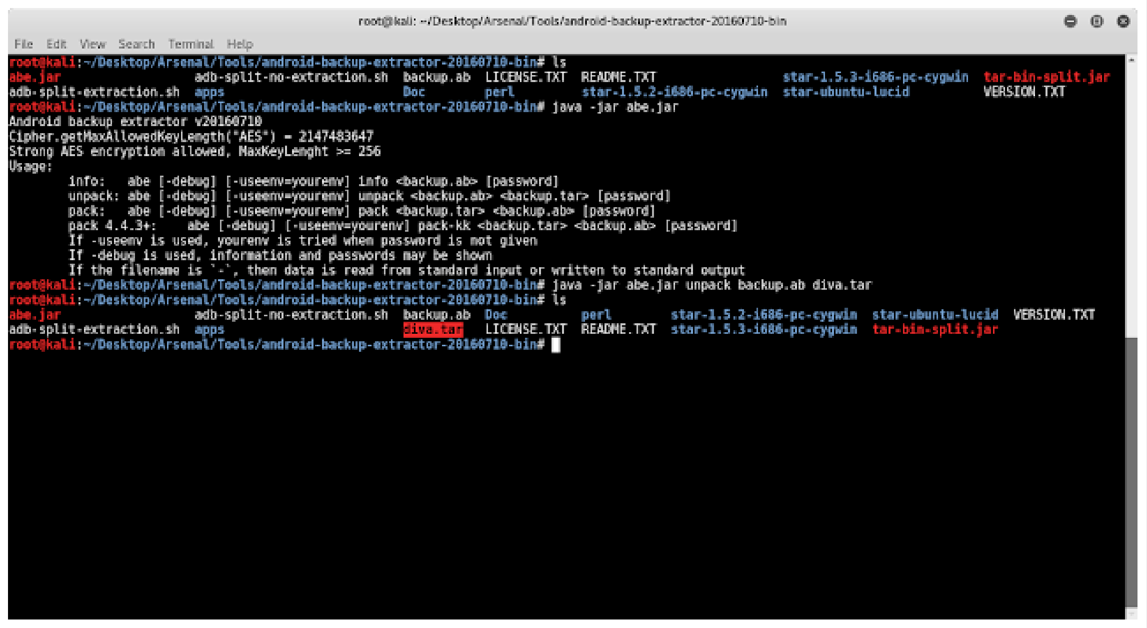


root@kali: ~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin
File Edit View Search Terminal Help

```
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# ls
abe.jar                     apps       LICENSE.TXT  star-1.5.2-i686-pc-cygwin  tar-bin-split.jar
adb-split-extraction.sh     backup.ab  perl         star-1.5.3-i686-pc-cygwin  VERSION.TXT
adb-split-no-extraction.sh  Doc        README.TXT   star-ubuntu-lucid
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# java -jar abe.jar
Android backup extractor v20160710
Cipher.getMaxAllowedKeyLength("AES") = 2147483647
Strong AES encryption allowed, MaxKeyLenght >= 256
Usage:
        info:   abe [-debug] [-useenv=yourenv] info <backup.ab> [password]
        unpack: abe [-debug] [-useenv=yourenv] unpack <backup.ab> <backup.tar> [password]
        pack:   abe [-debug] [-useenv=yourenv] pack <backup.tar> <backup.ab> [password]
        pack 4.4.3+:    abe [-debug] [-useenv=yourenv] pack-kk <backup.tar> <backup.ab> [password]
        If -useenv is used, yourenv is tried when password is not given
        If -debug is used, information and passwords may be shown
        If the filename is `-`, then data is read from standard input or written to standard output
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# 
```

I used below command to convert .ab file into extractable tar file using abe (Android Backup Extractor) tool.

java -jar abe.jar unpack backup.ab diva.tar



root@kali: ~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin
File Edit View Search Terminal Help

```
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# ls
abe.jar                  adb-split-no-extraction.sh  backup.ab  LICENSE.TXT  README.TXT                 star-1.5.3-i686-pc-cygwin  tar-bin-split.jar
adb-split-extraction.sh  apps                        Doc        perl         star-1.5.2-i686-pc-cygwin  star-ubuntu-lucid          VERSION.TXT
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# java -jar abe.jar
Android backup extractor v20160710
Cipher.getMaxAllowedKeyLength("AES") = 2147483647
Strong AES encryption allowed, MaxKeyLenght >= 256
Usage:
        info:   abe [-debug] [-useenv=yourenv] info <backup.ab> [password]
        unpack: abe [-debug] [-useenv=yourenv] unpack <backup.ab> <backup.tar> [password]
        pack:   abe [-debug] [-useenv=yourenv] pack <backup.tar> <backup.ab> [password]
        pack 4.4.3+:    abe [-debug] [-useenv=yourenv] pack-kk <backup.tar> <backup.ab> [password]
        If -useenv is used, yourenv is tried when password is not given
        If -debug is used, information and passwords may be shown
        If the filename is `-`, then data is read from standard input or written to standard output
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# java -jar abe.jar unpack backup.ab diva.tar
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# ls
abe.jar                  adb-split-no-extraction.sh  backup.ab  Doc          perl        star-1.5.2-i686-pc-cygwin  star-ubuntu-lucid  VERSION.TXT
adb-split-extraction.sh  apps                        diva.tar   LICENSE.TXT  README.TXT  star-1.5.3-i686-pc-cygwin  tar-bin-split.jar
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# 
```

You can observe that diva.tar file was as mentioned in above screenshot. Now lets extract it using below command:

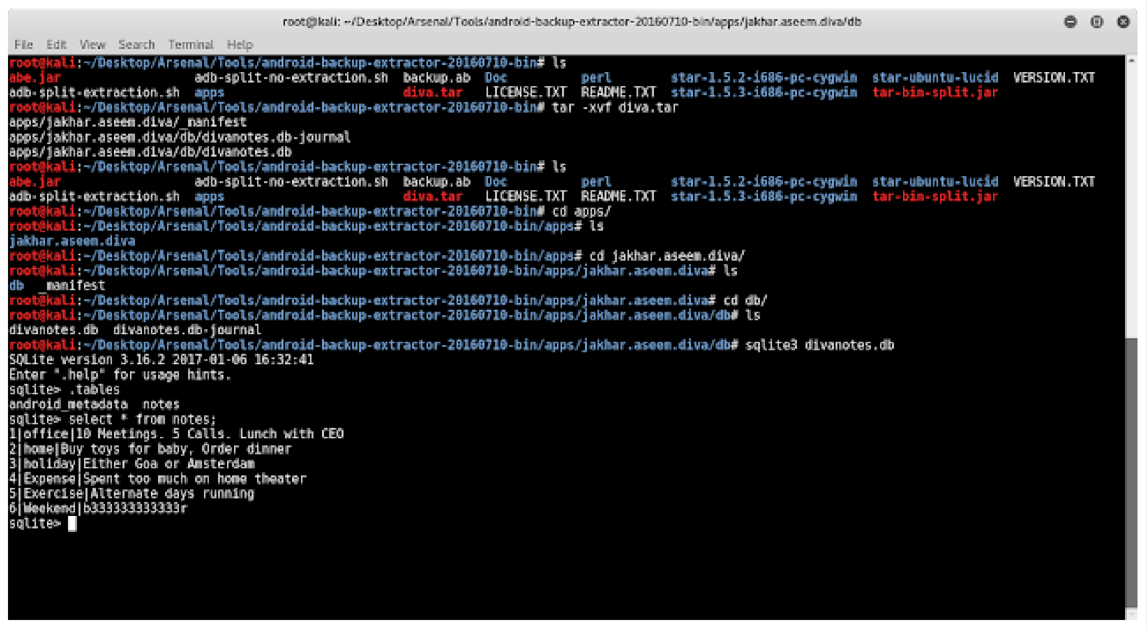
tar -xvf diva.tar

You can observed that it has created few files under apps/ folder. Now lets navigate and access those files. I was able to access divanotes.db (Database) file using Sqlite3 command as mentioned below:

sqlite3 divanotes.db (to open database)

.tables (to list all tables)

select * from notes; (to view complete 'notes' table data)



root@kali: ~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps/jakhar.aseem.diva/db
File Edit View Search Terminal Help

```
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# ls
abe.jar                 adb-split-no-extraction.sh  backup.ab  Doc          perl        star-1.5.2-i686-pc-cygwin  star-ubuntu-lucid  VERSION.TXT
adb-split-extraction.sh  apps                        diva.tar   LICENSE.TXT  README.TXT  star-1.5.3-i686-pc-cygwin  tar-bin-split.jar
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# tar -xvf diva.tar
apps/jakhar.aseem.diva/_manifest
apps/jakhar.aseem.diva/db/divanotes.db-journal
apps/jakhar.aseem.diva/db/divanotes.db
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# ls
abe.jar                 adb-split-no-extraction.sh  backup.ab  Doc          perl        star-1.5.2-i686-pc-cygwin  star-ubuntu-lucid  VERSION.TXT
adb-split-extraction.sh  apps                        diva.tar   LICENSE.TXT  README.TXT  star-1.5.3-i686-pc-cygwin  tar-bin-split.jar
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin# cd apps/
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps# ls
jakhar.aseem.diva
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps# cd jakhar.aseem.diva/
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps/jakhar.aseem.diva# ls
db  _manifest
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps/jakhar.aseem.diva# cd db/
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps/jakhar.aseem.diva/db# ls
divanotes.db  divanotes.db-journal
root@kali:~/Desktop/Arsenal/Tools/android-backup-extractor-20160710-bin/apps/jakhar.aseem.diva/db# sqlite3 divanotes.db
SQLite version 3.16.2 2017-01-06 16:32:41
Enter ".help" for usage hints.
sqlite> .tables
android_metadata  notes
sqlite> select * from notes;
1|office|10 Meetings. 5 Calls. Lunch with CEO
2|home|Buy toys for baby, Order dinner
3|holiday|Either Goa or Amsterdam
4|Expense|Spent too much on home theater
5|Exercise|Alternate days running
6|Weekend|b333333333333r
sqlite> 
```

So this is how we can check for android application backup vulnerability for any target application.

Thanks

Posted by Frogy at 3/03/2017

Labels: allowbackup, android, android app, android application, android security, backup

No comments:

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)