Android Testing Environment Cheatsheet (Part 2)

This part will cover installing Android testing tools on the Kali VM and their basic usage.

Configuring Burp

Burp is already installed on the Kali VM. The following changes are needed:

Configure Burp on Kali

Choose the `Proxy` -> `Options` tab.

Choose `Edit`

network)

Configure proxy settings in Genymotion

Choose `Settings` -> `Wi-Fi`.

Hold down `WiredSSID` and choose `Modify`.

In `Advanced Options`, choose `Manual` proxy.

Enter the Kali VM address for the `Proxy hostname` e.g. 192.168.56.101.

Enter Burp's port number for the `Proxy port` e.g. 8080.

Install Burp's cert in Genymotion

Install Chrome e.g. `$ sudo apt-get install chromium`

In the Kali VM Chrome browser: `Settings` -> `Network` -> `Network proxy`.

Add 127.0.0.1, port 8080 in for the `Manual` proxy settings.

If your burp is not listening on all interfaces, change your burp listener to listen on 127.0.0.1 temporarily.

Visit `http://burp`.

Download the portswigger cert from the `CA Certificate` link.

Rename the downloaded cert to `cacert.cer`.

Copy the cert onto Genymotion's sdcard:

Connect to Genymotion using adb as shown in Part 1, then:

```
$ adb push cacert.cer /sdcard/cacert.cer
```

Choose the `cacert.cer` at the bottom of the list and install.
Name the cert `portswigger`.

Finally, if your burp is not listening on all interfaces, change your burp listener back to listen on the address of your Kali VM vbox network.

Test your setup by visiting a web page in Genymotion and verifying that the traffic can be seen in Burp on the Kali VM.

Obtaining and installing android apps for testing

Install a plugin for chrome which allows downloading of APKs from the Google Play store e.g. `APK Downloader Appsofto`
Create a test google account for logging into the play store.
When visiting the play store, just click on the app's icon and paste in the name of the package to download.
Of course, make sure to download apps which are compatible with your target Genymotion device.

Installing an APK onto Genymotion
To install your downloaded APK from the app store onto Genymotion, run the following

```
$ adb install appname.apk
```

The installed app should now be in the Genymotion menu.

Install testing tools on Kali VM

Install Chrome browser

```
$ sudo apt-get install chromium
```

Install Drozer

Download Drozer.
Download drozer.apk agent.

Install Drozer server on Kali VM:

```
$ sudo apt-get install python-protobuf
$ sudo apt-get -f install (to install deps for python-protobuf)
$ sudo dpkg -i drozer.deb
```

Install Drozer agent on GenyMotion:
Connect to Genymotion using adb as shown in Part 1, then:

```
$ adb install drozer.apk
```

Open the drozer agent app on Genymotion and turn `ON` the server.

In Kali:

```
$ adb forward tcp:31415 tcp:31415
```

```
$ drozer console connect
```

Install APKtool

Download the apktool.

Download the apktool wrapper.

```
$ mv apktool-ver-number.jar apktool.jar
```

```
$ sudo cp apktool.jar /usr/local/bin
```

```
$ sudo cp apktool /usr/local/bin
```

```
$ sudo chmod u+x /usr/local/bin/apktool*
```

You should now be able to run the apktool against your target APK and browse the resulting directory to review the manifest file etc:

```
$ apktool d appname.apk
```

Install dex2jar

dex2jar: convert an APK file into a JAR file.

Unzip and add the dex2jar script to your PATH/.profile:

```
$ unzip dex2jar-ver-num.zip
```

```
$ PATH="$PATH:/home/android/tools/dex2jar-ver-num/"
```

You should now be able to run the dex2jar tool against your target APK to create a jar file for browsing with jd-gui tool.

```
$ dex2jar-ver-num appname.apk
```

Install jd-gui
jd-ui: browse a JAR file and save source files as .java.

Download the jd-gui jar file.
Create a file name jd-gui with the following command tailored for your location:

```
java -jar /home/android/tools/jd-ui-ver-num.jar
```

Make the file executable and add to your PATH/.profile:

```
$ chmod u+x jd-gui
```

```
$ PATH="$PATH:/home/android/tools/"
```

You should now be able to run the jd-gui tool and open your newly created jar file from dex2jar, and browse the reconstructed Java source:

```
$ jd-gui
```

jdax: converts an APK file into Java source code (also useful if dex2jar doesn't work).

Download jadx and unzip.
Add to your PATH/.profile:

```
$ PATH="$PATH:/home/android/tools/jadx-vernum/bin/"
```

Run jadx-gui and select your target APK or JAR file to browse:

```
$ jadx-gui
```

Install FindBugs & FindSecurityBugs
These plugins will work with Android Studio and can be used to scan source code for potential security issues.

Install FindBug:
Open Android Studio, `File` -> `Settings` -> `Plugins`.
Search for "FindBugs", install and restart Android Studio.

Install FindSecurityBugs:
Download FindSecurityBugs plugin for Android Studio.
In Android Studio, `File` -> `Settings` -> `Plugins` -> `Install plugin form disk`.
Select the plugin zip file and restart Android Studio.

0xsh / October 19, 2016 / android-cheatsheet

Leave a Reply

Your email address will not be published. Required fields are marked *

COMMENT

NAME *

EMAIL *

WEBSITE

POST COMMENT

PREVIOUS

Android Testing Environment Cheatsheet (Part 1)

NEXT

Data exfiltration via PSK31 with GNU Radio

RECENT POSTS

- Android Testing Environment Cheatsheet (Part 2)
- Android Testing Environment Cheatsheet (Part 1)
- Listening to Iridium satellite traffic on Ubuntu (16.04 LTS)

RECENT COMMENTS

CATEGORIES

- android
- cheatsheet
- infrastructure
- pentesting
- sdr
- wifi

ARCHIVES

- March 2017
- October 2016
- September 2016
- July 2016

Search …

META

- Log in
- Entries RSS
- Comments RSS
- WordPress.org