Traffic Analysis for Android Devices

Traffic Analysis for Android Device

Android Traffic Interception

Ways to Analyse Android Traffic

Passive Analysis, Active Analysis

HTTP Proxy Interception

Other ways to intercept SSL traffic

Traffic Analysis for Android Devices

- Often applications leak sensitive information in their network data, so finding it is one of the most crucial tasks of a penetration tester.
- Also, you will often encounter applications that perform authentication and session management over insecure network protocols.
- So, here we will learn the ways to intercept and analyze traffic of various applications in an Android device.

Android Traffic Interception

Android traffic interception

- The insufficient transport layer protection is the third biggest risk in mobile devices according to OWASP Mobile Top10 (https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks).
- In fact, imagine a scenario where an application is submitting the user's login credentials via HTTP to the server.
- What if the user is sitting in a coffee shop or at an airport and is logging in to his application while someone is sniffing the network.
- The attacker will be able to get the entire login credentials of the particular user, which could be used for malicious purposes later.

Android Traffic Interception

Android traffic interception (Continue…)

- Let's say the application is doing the authentication over HTTPS, the session management over HTTP, and is passing the authentication cookies in the requests.
- In that case as well, the attacker will be able to get the authentication cookies by intercepting the network while performing a man-in-the middle attack.
- Using those authentication cookies, he could then directly log in to the application as the victim user.

Ways to Analyze Android Traffic

- There are two different ways of traffic capture and analysis in any scenario.
- We will be looking at the two different types that are possible in the Android environment and how to perform them in a real-world scenario.
- The Passive and Active analyses

Ways to Analyze Android Traffic

- **Passive analysis:**
  - This is a way of traffic analysis in which no active interception is done with the application sending the network data.
  - Instead, we will try to capture all the network packets and later open it up in a network analyzer, such as Wireshark, and then try to find out the vulnerabilities or the weak security issues in the application.
- **Active analysis:**
  - In Active analysis, the penetration tester will actively intercept all the network communications being made and can analyze, assess, and modify the data on the fly.
  - Here, he will be setting up a proxy and all the network calls being made and received by the application/device will pass through that proxy.

Ways to Analyze Android Traffic

Passive analysis:

- In Passive analysis, the concept is to save all the network information to a specific file and later view it using a packet analyzer.
- This is what we will be doing with Passive analysis in Android devices as well.
- We will be using tcpdump in order to save all the information to a location onto the device itself.
- Thereafter, we will pull that file to our system and then view it using Wireshark or Cocoa packet analyzer.

Passive Analysis

- **Step-1 - Installing TCPDump**
  - tcpdump is a command-line utility that captures the traffic on a particular network device and dumps it to the filesystem.
  - tcpdump can be downloaded from https://www.tcpdump.org/release/tcpdump-4.99.1.tar.gz
  - Once the *tcpdump* binary has been downloaded, all we need to do is use adb to push the file onto the device.
  - To be able to do so, your handset needs to be connected to and properly identified by your computer.
  - adb devices
  - adb push /home/tcpdump /data/local
  - adb shell
  - cd /data/local
  - chmod 777 tcpdump/
  - cd tcpdump/

Passive Analysis

- **Step-1 - Installing TCPDump**

Command Prompt

```
c:\Program Files\Genymobile\Genymotion\tools>adb.exe devices
List of devices attached
192.168.198.101:5555    device


c:\Program Files\Genymobile\Genymotion\tools>adb.exe push c:\users\Parag\Downloads\tcpdump-4.99.1\tcpdump /data/local
c:\users\Parag\Downloads\tcpdump-4.99.1\tcpdump\:...files pushed. 3.9 MB/s (13723558 bytes in 3.381s)
```

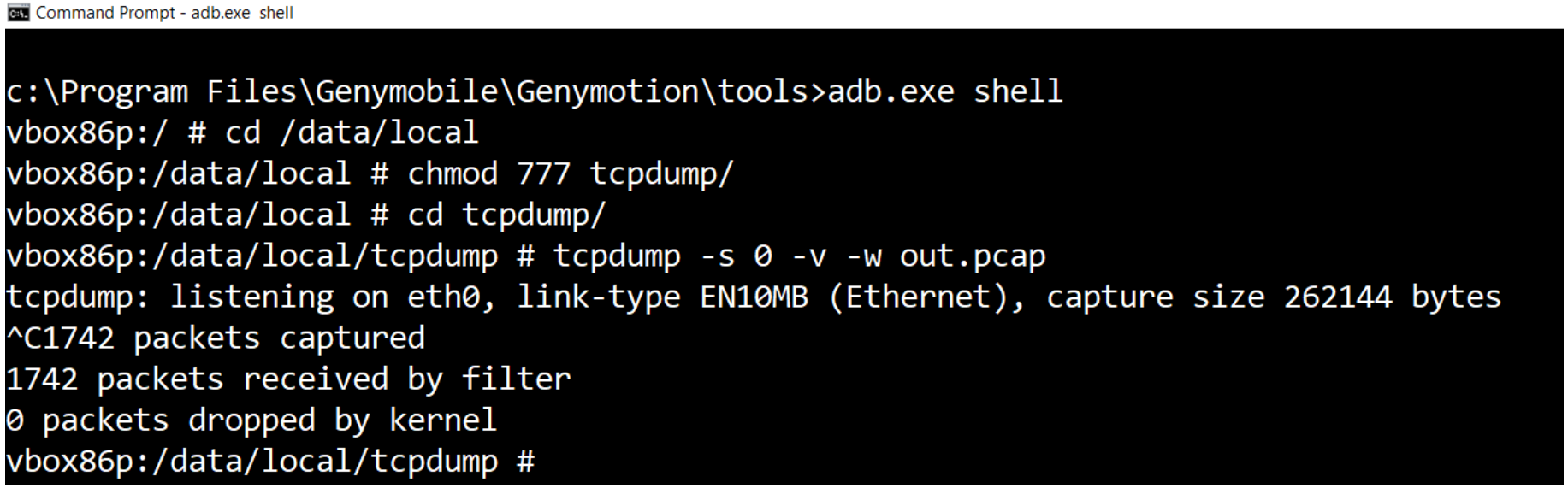
Command Prompt - adb.exe shell

```
c:\Program Files\Genymobile\Genymotion\tools>adb.exe shell
vbox86p:/ # cd /data/local
vbox86p:/data/local # chmod 777 tcpdump/
vbox86p:/data/local # cd tcpdump/
vbox86p:/data/local/tcpdump #
```

Passive Analysis

- **Step-2 – Saving the Traffic Dump to File**
  - Tcpdump can now be started from the same adb shell and the output saved to a file

    tcpdump -s 0 -v -w out.pcap

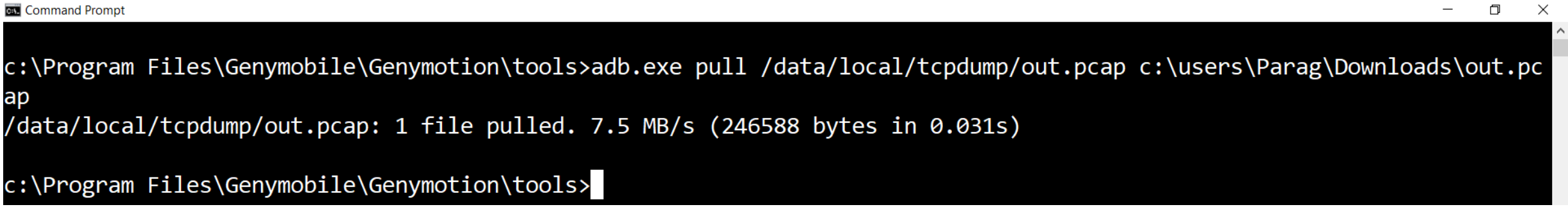


  - Once the dump is completed, we can stop capturing the data.
  - In order to do so, you simply need to press Ctrl+C.

Passive Analysis

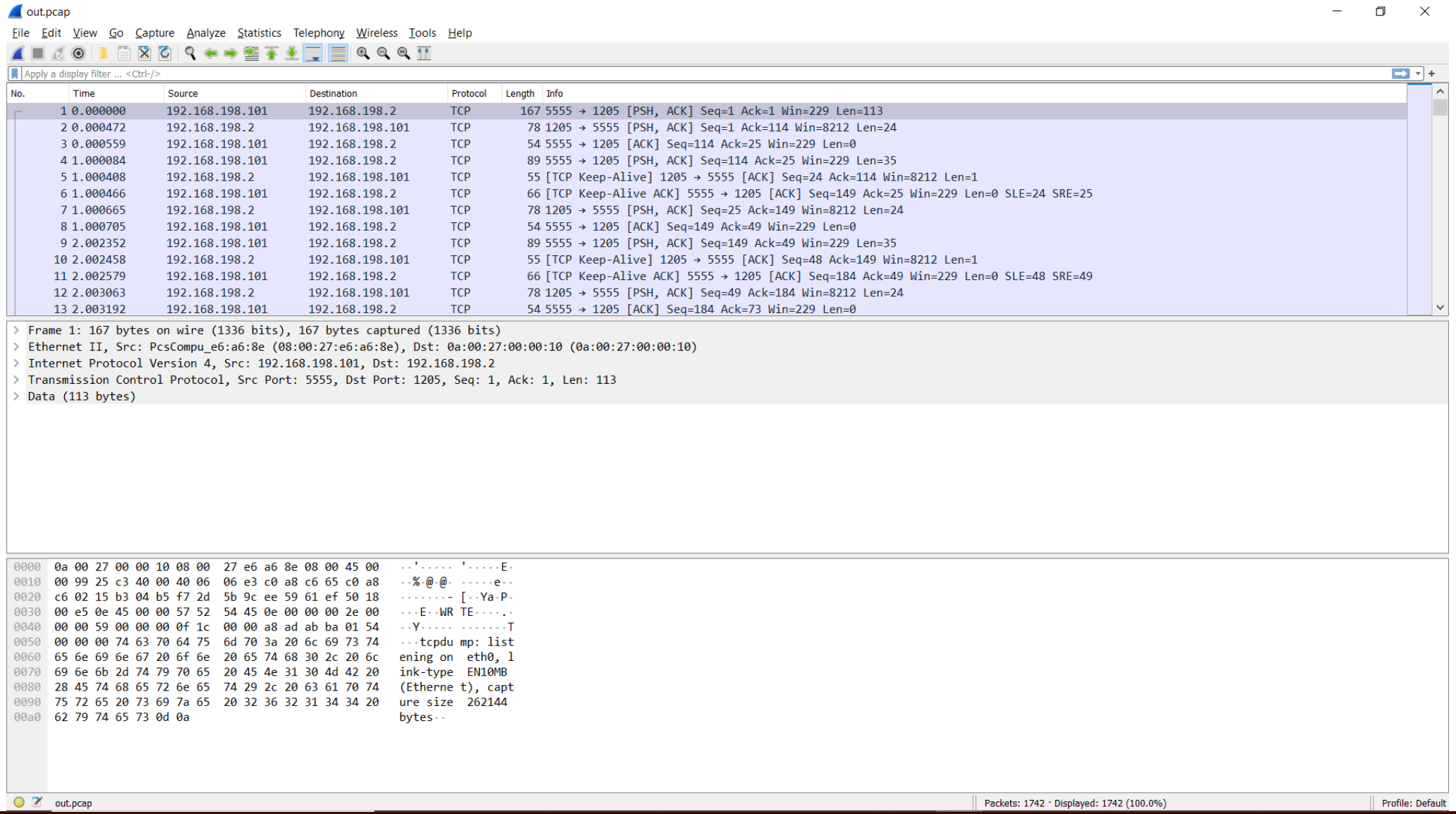
- **Step-2 – Saving the Traffic Dump to File (Continue)**
  - The resulting file can be pulled out of the device and saved locally, so that it can get analyzed using *Wireshark*.

adb pull /data/local/tcpdump/out.pcap D:/Parag/out.pcap



Passive Analysis

- **Step-3 Open the pcap file in Wireshark**



Active Analysis

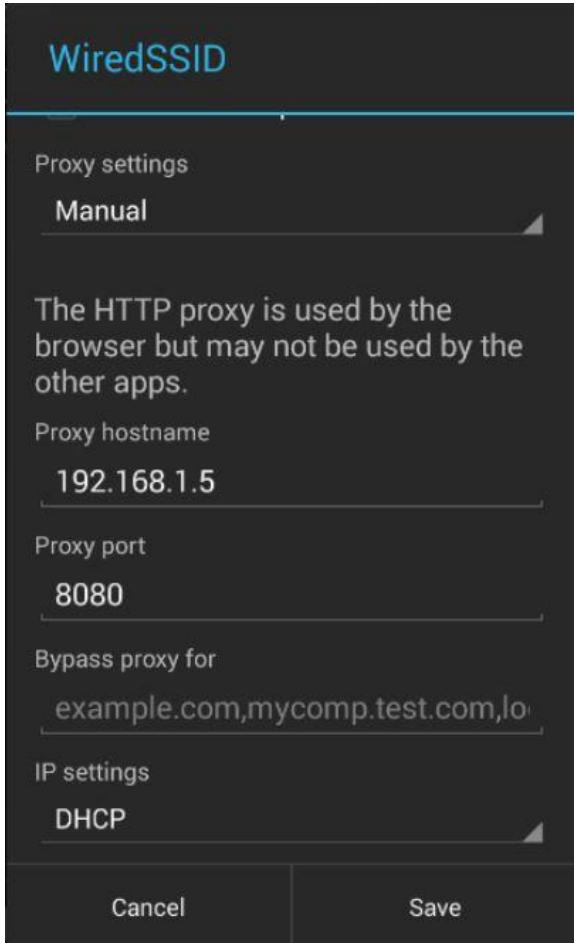
- In Active analysis, the fundamental rule is to make every request and response pass through an intermediate stage defined by us.
- In this case, we need to set up a proxy and make all the requests and responses go through that particular proxy.
- Also, we have an option to manipulate and modify both the packets in the requests and response, and thus assess the application's security.
- In order to create a proxy for HTTP, start up the emulator with the -httpproxy flag specifying the proxy IP and Port.
- Since we are running the emulator on the same system, we will use the IP 127.0.0.1 and any port that is available.
- In this case, we will be using the port 8080.

Active Analysis

- On a device, we could also set up the proxy by navigating to Settings | Wi-Fi and then long tapping on the network Wi-Fi that we are connected to.
- Also, the system that is used for interception should be on the same network if we are doing it using an actual device.
- Once we long tap on the Wi-Fi connection, we will have a screen similar to the one shown in the following screenshot.
- Also, if you're performing this analysis with a real device, the device needs to be on the same network as the proxy

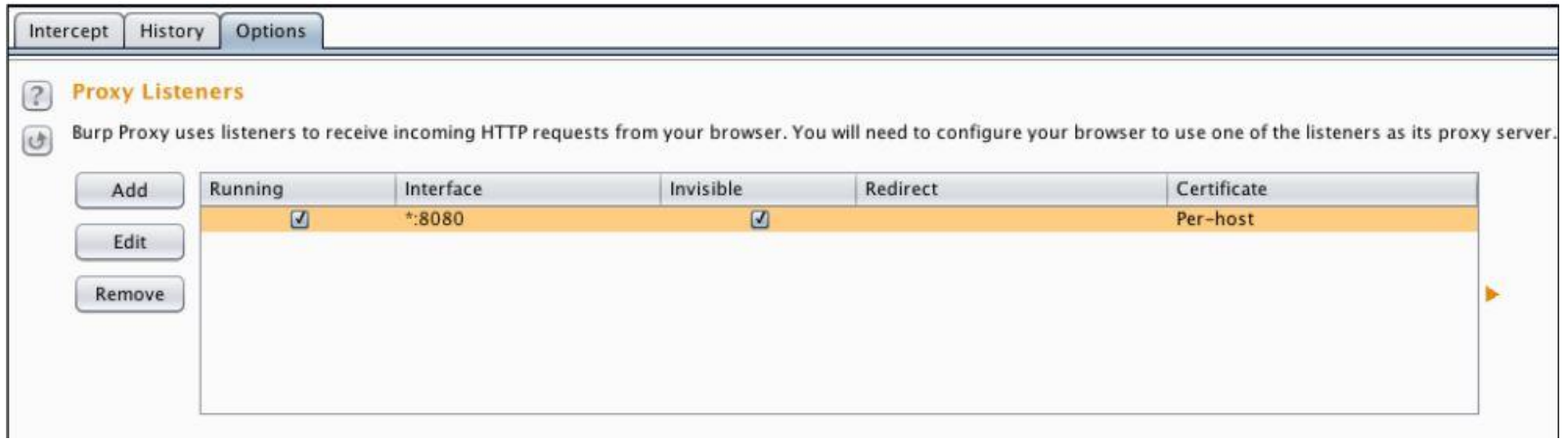
Active Analysis

- Once into the modify connection screen, while going down, notice the proxy configurations asking for the IP address of the device on the network and the port of the proxy system.
- However, these settings are only in the latest versions of Android starting from 4.0. If we want to implement a proxy on a device less than 4.0, we will have to install a third-party application, such as ProxyDroid available on Play Store



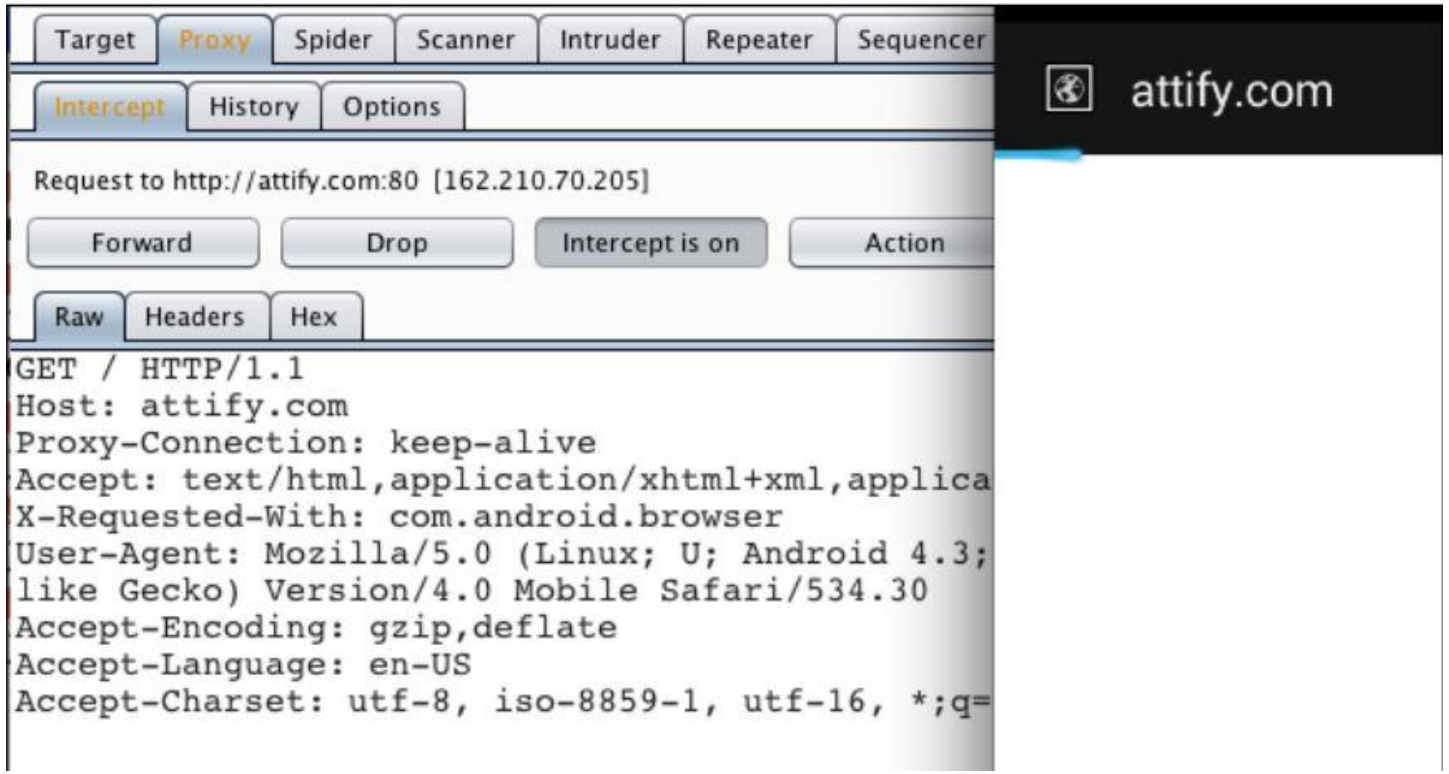
Active Analysis

- Once we have set up the proxy in the device/emulator, go ahead and launch the Burp Proxy in order to intercept the traffic.
- Here is how the Burp setting should look in the Options tab in order to effectively intercept the traffic of both the browser and the application.
- We also need to check the invisible proxy in order to make sure that our proxy is also capturing the nonproxy requests.



Active Analysis

- In order to check whether the proxy is working or not, open up the browser and launch a website.



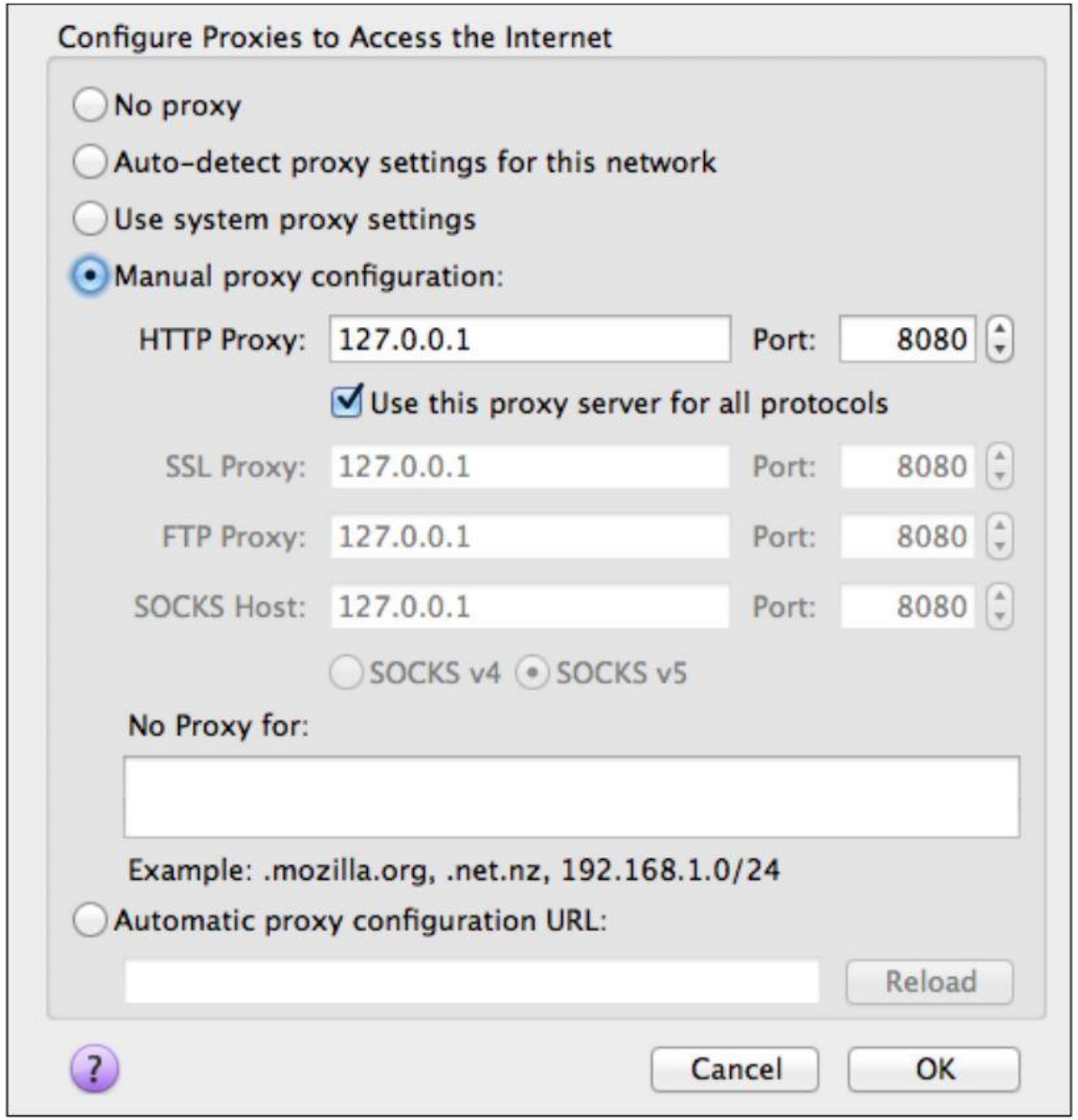
- As we can see in the preceding screenshot, we are opening up a URL, http://attify.com, and the request is right now being displayed in the Burp Proxy screen.
- So, we have managed to successfully intercept all the HTTP-based requests from the device and the application.

HTTPS Proxy Interception

- The preceding method will work in the normal traffic interception of application and browser when they are communicating via the HTTP protocol.
- In HTTPS, we will get an error due to the certificate mismatch, and thus we won't be able to intercept the traffic.
- However, in order to solve the challenge, we can create our own certificate or Burp/PortSwigger and installing it on the device.
- In order to create our own certificate, we will need to set up a proxy in Firefox (or any other browser or global proxy)
- However, in order to solve the challenge, we will be creating our own certificate or Burp/PortSwigger and installing it on the device.
- In order to create our own certificate, we will need to set up a proxy in Firefox (or any other browser or global proxy)

HTTPS Proxy Interception

- Once in the Network tab, we need to click on Settings in order to configure the proxy with Firefox.
- Once done, go to the HTTPS website on our system browser of which we would want to intercept the traffic on our device.
- Here we will receive a The Network is Untrusted message. Click on I understand the Risks and hit Add Exception.



HTTPS Proxy Interception

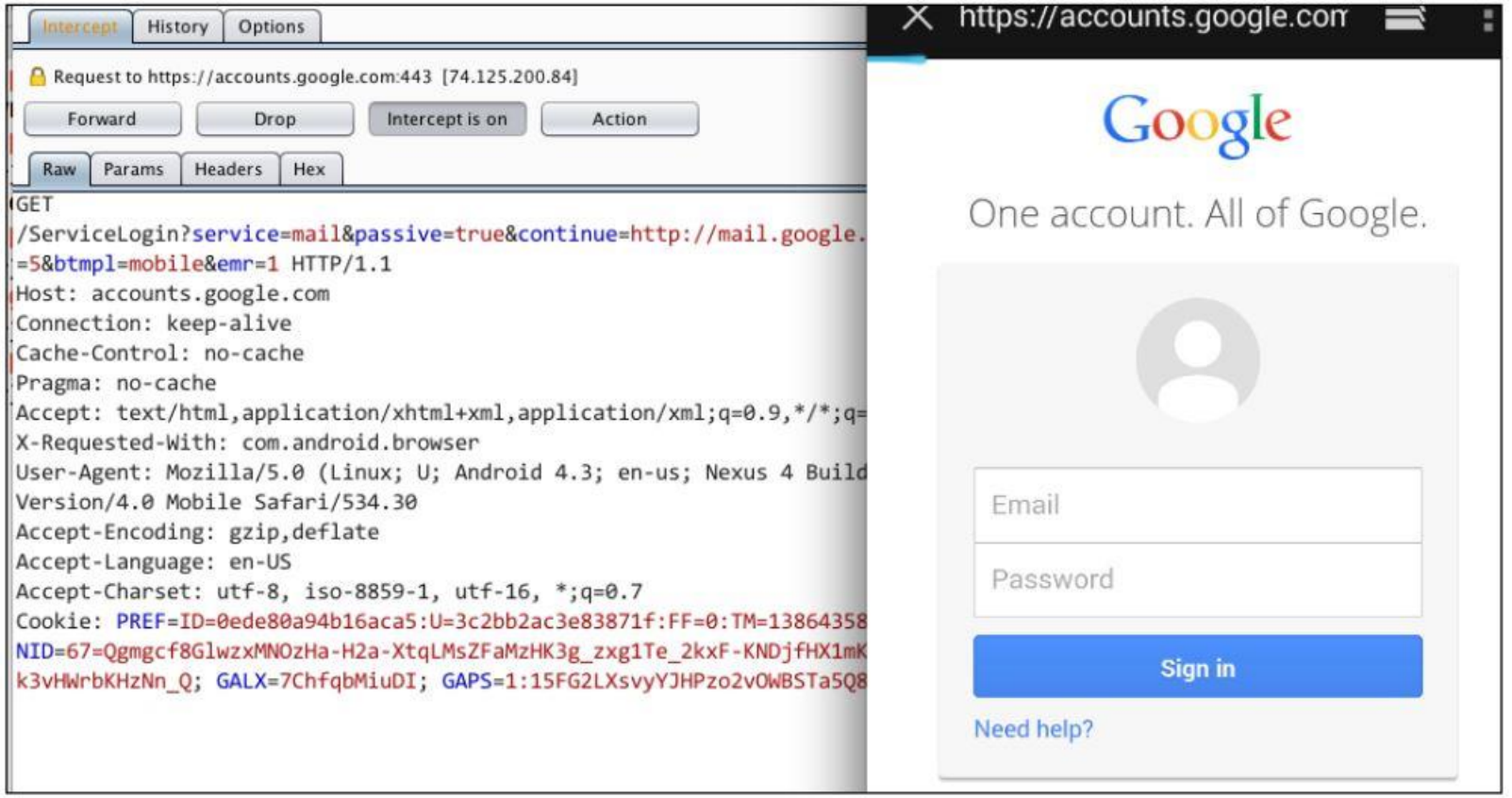
- Thereafter, click on Get Certificate and finally click on View and then on Export in order to save the certificate.
- Once the certificate is saved on our system, we could now push this to our device using adb.

```
adb push portswiggerca.crt /mnt/sdcard/portswiggerca.crt
```

- Now, in our device, go to Settings, and under the Personal category, we will find Security. Once we go into Security, notice that there is an option to install certificates from the SD card. Clicking on that will lead us to finally save the certificate with a given name, which will be applicable for all the applications and browsers for even the HTTPS websites.

HTTPS Proxy Interception

- Confirm this by going back to our browser and opening an HTTPS website, such as https://gmail.com in this case. As we can see in the following screenshot, we have successfully intercepted the communication in this case as well:



Other Ways to intercept the SSL traffic

- There are other ways to do SSL traffic interception as well as different ways to install certificates on the device.
- One of the other ways include pulling the cacerts.bks file from the /system/etc/ security location of the Android device.
- Once we have pulled it out, we could then use the key tool along with Bouncy Castle (located in the Java installation directory) to generate the certificate.
- If you're unable to find Bouncy Castle in the Java installation directory, you could also download it from http://www.bouncycastle.org/latest_ releases.html and place it at a known path.
- Thereafter, we will need to mount the /system partition as read/write in order to push the updated cacerts.bks certificate back to the device.

Other Ways to intercept the SSL traffic

- However, in order to make this change permanent in case we are using an emulator, we will need to use mks.yaffs2 in order to create a new system. img and then use it.
- Also, there are other tools you can use to intercept traffic of Android devices, such as Charles Proxy and MITMProxy (http://mitmproxy.org).
- I highly recommend you to try out both of them on the basis of the knowledge of Burp proxying, as they are quite the same when it comes to usability, but are much more powerful.
- While using Charles Proxy, we could directly download the certificate from www.charlesproxy.com/charles.crt.

Other Ways to intercept the SSL traffic

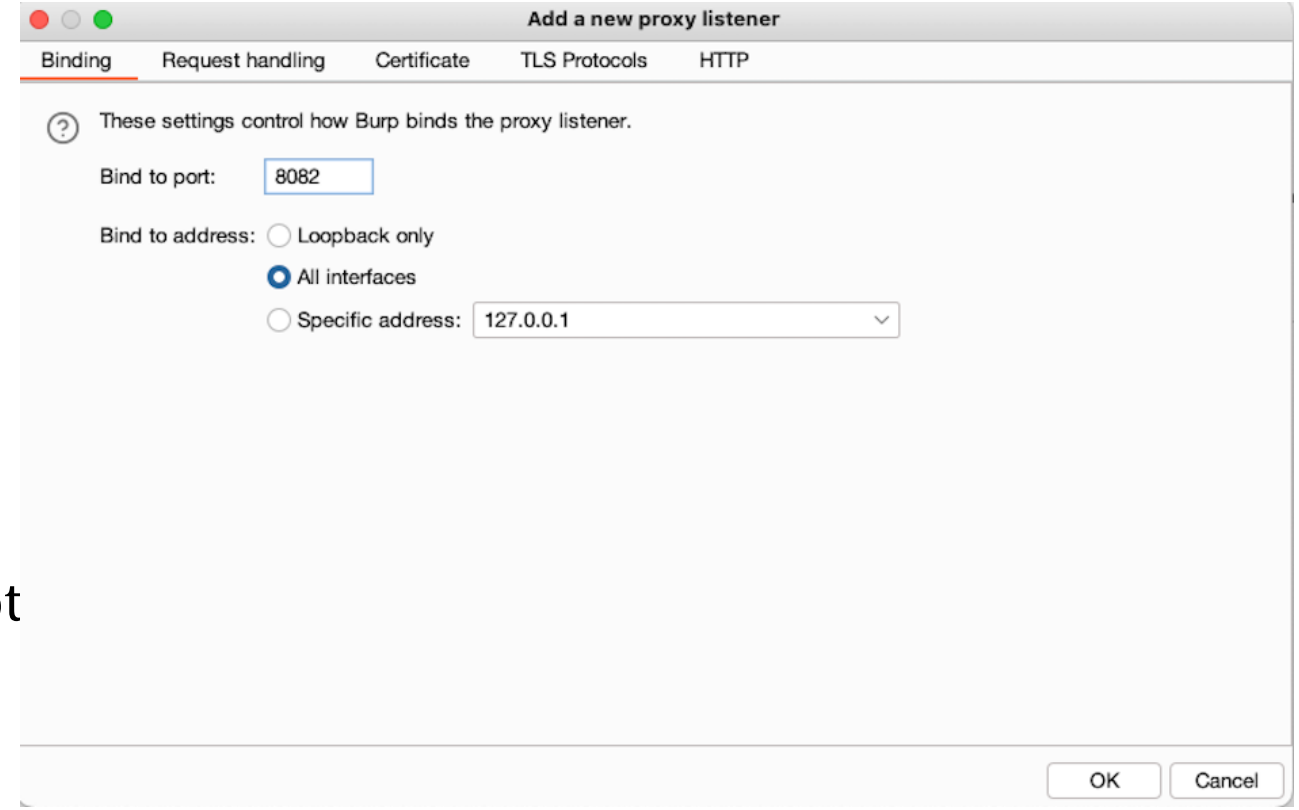
- A secure way of implementing traffic securely in the application is to have everything go over HTTPS and at the same time include a certificate in the app itself.
- This is done so that when the application tries to communicate with the server, it will verify if the server certificate corresponds with the one present in the application.
- However, if someone is doing a penetration test and is intercepting the traffic, the new certificate used by the device that has been added by the penetration tester, such as the portswigger certificate, won't match the one present in the application.
- In those cases, we will have to reverse engineer the application and analyze how the app is verifying the certificates.
- We might even need to modify and recompile the application

Configuring Android Device to work with Burp Suite

Step 1: Configure the Burp Proxy listener

To configure the proxy settings for Burp Suite Professional:

1. Open Burp Suite Professional and click **Settings** to open the **Settings** dialog.
2. Go to **Tools > Proxy**.
3. In **Proxy Listeners**, click **Add**.
4. In the **Binding** tab, set **Bind to port** to 8082 (or another port that is not in use).
5. Select **All interfaces** and click **OK**.

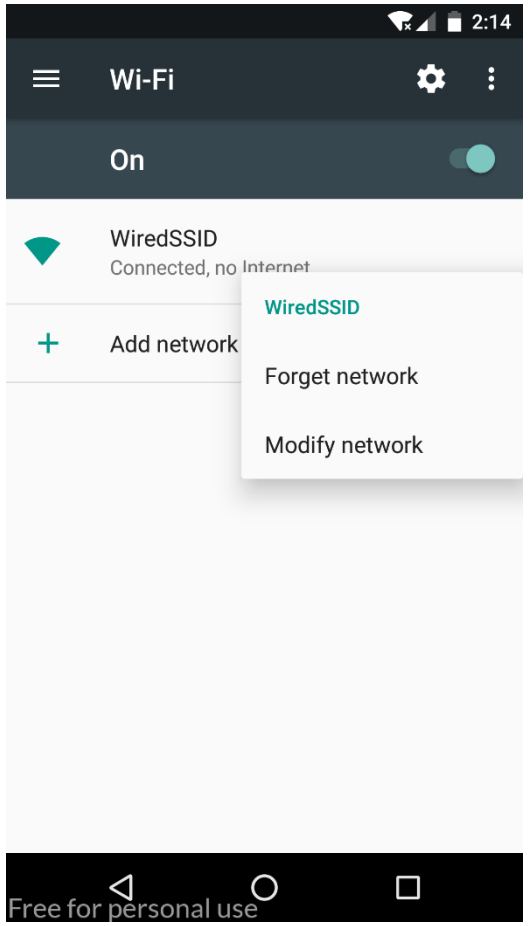


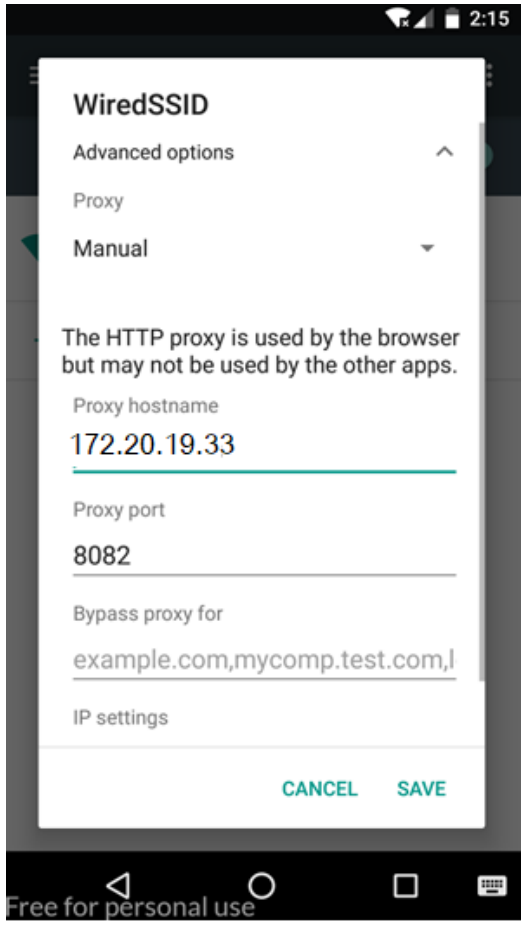
Configuring Android Device to work with Burp Suite

Step 2: Configure your device to use the proxy

Make sure that your Android device is disconnected from the Wi-Fi network before you attempt to configure the proxy settings:

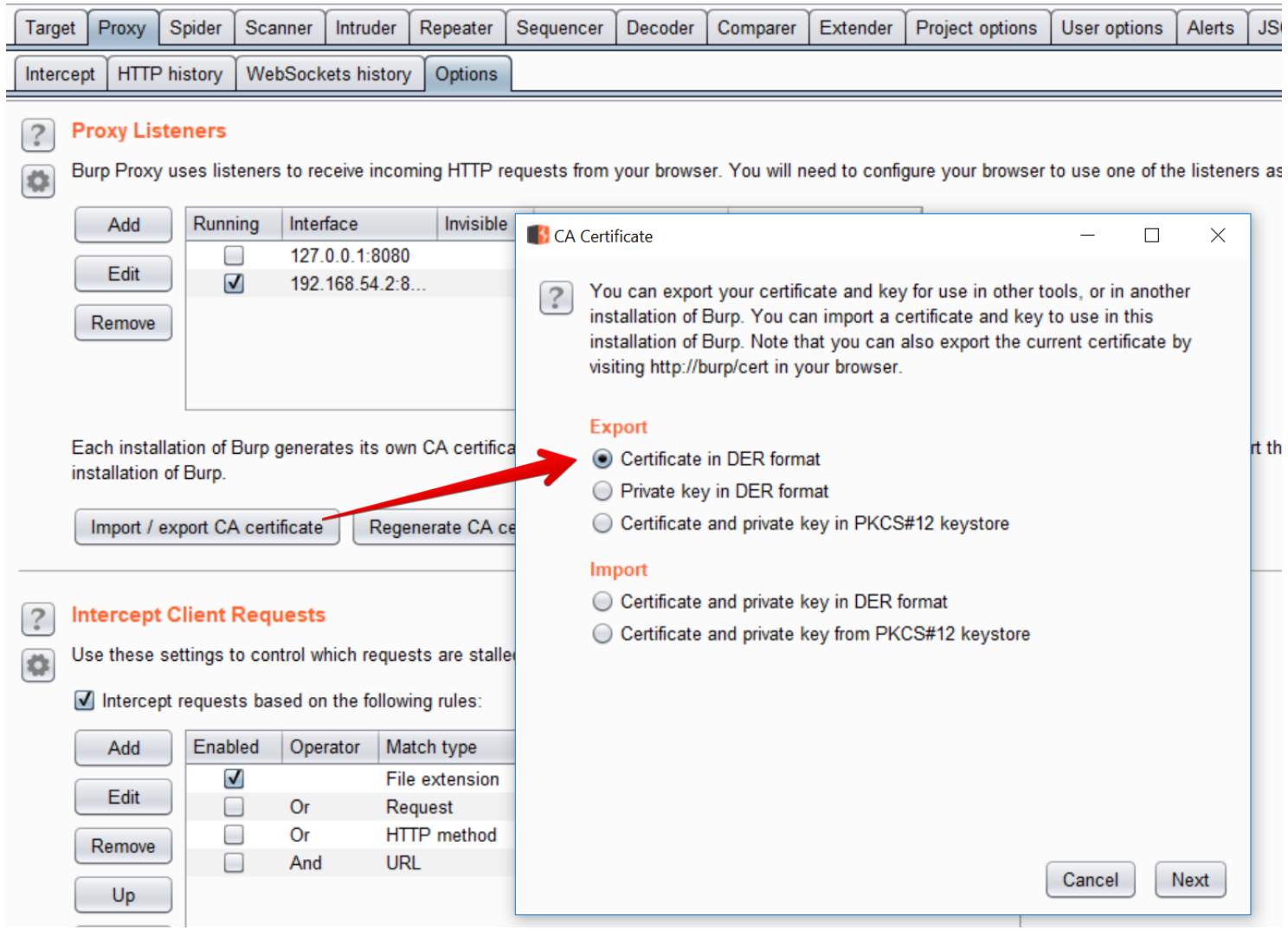
1. In your Android device, go to **Settings > Network & internet.**
2. Select **Internet** and long-press the name of your Wi-Fi network.
3. Select **Modify.**
4. From the **Advanced options** menu, select **Proxy > Manual.**
5. Set **Proxy hostname** to the IP of the computer running Burp Suite Professional.
6. Set **Proxy port** to the port value that you configured for the Burp Proxy listener, in this example 8082.
7. Touch **Save.**





Configuring Android Device to work with Burp Suite

Step 3: Install CA Certificate on your Android Device

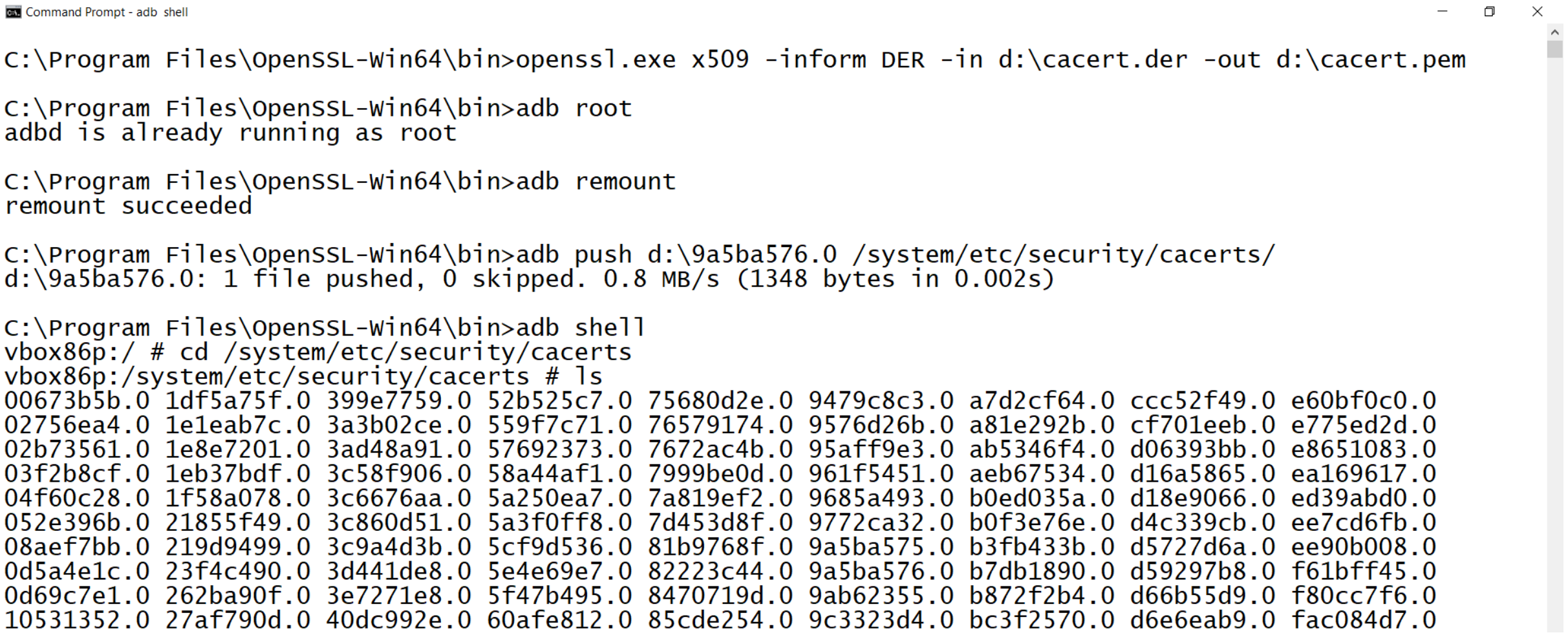


Export the certificate. Certificate will be saved as .der extension

Configuring Android Device to work with Burp Suite

Step 4: Push Certificate to Android Device

1. Convert .der to .pem using openssl
2. Rename the .pem file to .0 extension
3. Push the .0 file using adb command to /system/security/etc/cacerts of android device



Configuring Android Device to work with Burp Suite

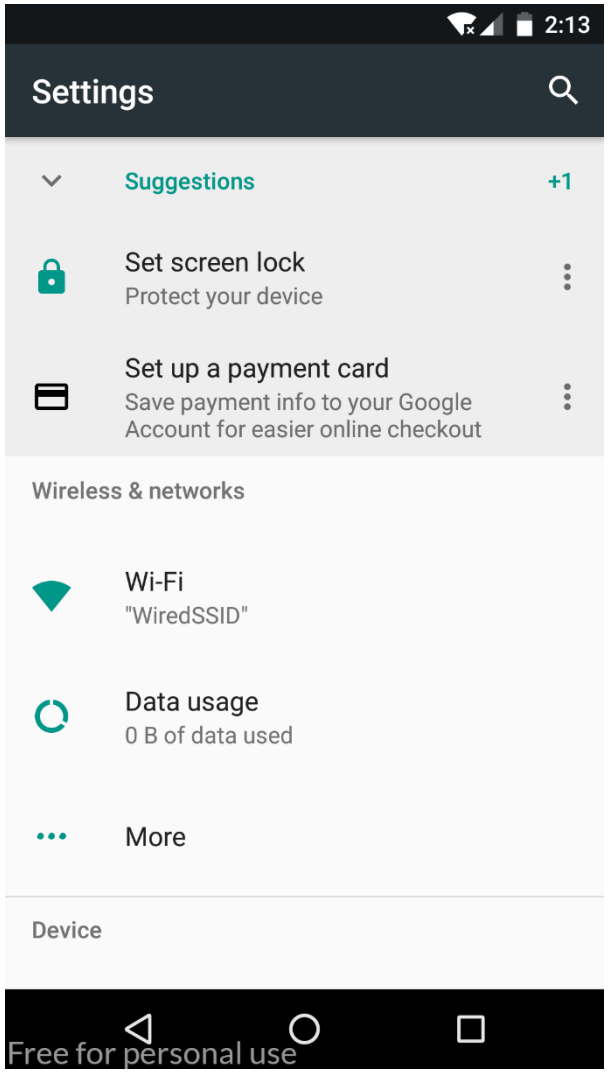
Step 5: Give read/write permission to file but don't give executable permission

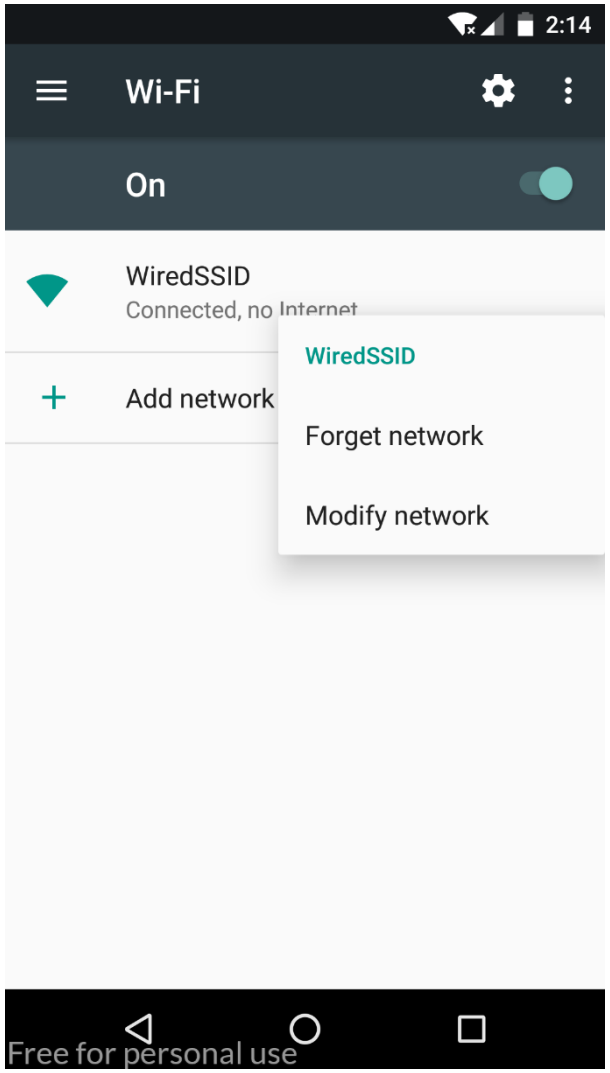
1. chmod 644 9a5ba576.0

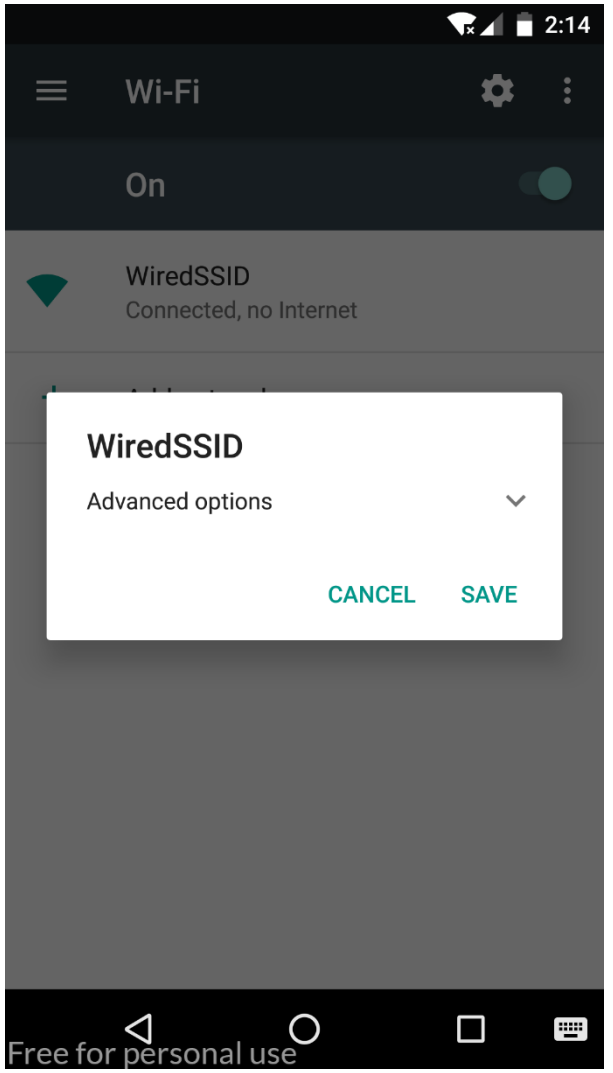
```
C:\Program Files\OpenSSL-Win64\bin>adb shell
vbox86p:/ # cd /system/etc/security/cacerts
vbox86p:/system/etc/security/cacerts # ls
00673b5b.0 1df5a75f.0 399e7759.0 52b525c7.0 75680d2e.0 9479c8c3.0 a7d2cf64.0 ccc52f49.0 e60bf0c0.0
02756ea4.0 1e1eab7c.0 3a3b02ce.0 559f7c71.0 76579174.0 9576d26b.0 a81e292b.0 cf701eeb.0 e775ed2d.0
02b73561.0 1e8e7201.0 3ad48a91.0 57692373.0 7672ac4b.0 95aff9e3.0 ab5346f4.0 d06393bb.0 e8651083.0
03f2b8cf.0 1eb37bdf.0 3c58f906.0 58a44af1.0 7999be0d.0 961f5451.0 aeb67534.0 d16a5865.0 ea169617.0
04f60c28.0 1f58a078.0 3c6676aa.0 5a250ea7.0 7a819ef2.0 9685a493.0 b0ed035a.0 d18e9066.0 ed39abd0.0
052e396b.0 21855f49.0 3c860d51.0 5a3f0ff8.0 7d453d8f.0 9772ca32.0 b0f3e76e.0 d4c339cb.0 ee7cd6fb.0
08aef7bb.0 219d9499.0 3c9a4d3b.0 5cf9d536.0 81b9768f.0 9a5ba575.0 b3fb433b.0 d5727d6a.0 ee90b008.0
0d5a4e1c.0 23f4c490.0 3d441de8.0 5e4e69e7.0 82223c44.0 9a5ba576.0 b7db1890.0 d59297b8.0 f61bff45.0
0d69c7e1.0 262ba90f.0 3e7271e8.0 5f47b495.0 8470719d.0 9ab62355.0 b872f2b4.0 d66b55d9.0 f80cc7f6.0
10531352.0 27af790d.0 40dc992e.0 60afe812.0 85cde254.0 9c3323d4.0 bc3f2570.0 d6e6eab9.0 fac084d7.0
111e6273.0 2add47b6.0 418595b9.0 6187b673.0 86212b19.0 9d6523ce.0 bdacca6f.0 d7746a63.0 facacbc6.0
119afc2e.0 2d9dafe4.0 450c6e38.0 63a2c897.0 87753b0d.0 9dbefe7b.0 bf64f35b.0 d8317ada.0 fb126c6d.0
124bbd54.0 2fa87019.0 455f1b52.0 6645de82.0 882de061.0 9f533518.0 c491639e.0 dbc54cab.0 fde84897.0
12d55845.0 33815e15.0 48a195d8.0 67495436.0 89c02a45.0 a0bc6fbb.0 c51c224c.0 dc99f41e.0 ff783690.0
1676090a.0 33815e15.1 4be590e0.0 69105f4f.0 8d6437c3.0 a2c66da8.0 c7e2a638.0 dfc0fe80.0
17b51fe6.0 343eb6cb.0 4e18c148.0 6e8bf996.0 91739615.0 a2df7ad7.0 c90bc37d.0 e268a4c5.0
1dac3003.0 35105088.0 5046c355.0 6fcc125d.0 9282e51c.0 a3896b44.0 cb156124.0 e442e424.0
1dcd6f4c.0 3929ec9f.0 524d9b43.0 72f369af.0 9339512a.0 a7605362.0 cb1c3204.0 e48193cf.0
vbox86p:/system/etc/security/cacerts # chmod 644 9a5b
9a5ba575.0   9a5ba576.0
vbox86p:/system/etc/security/cacerts # chmod 644 9a5ba57
9a5ba575.0   9a5ba576.0
vbox86p:/system/etc/security/cacerts # chmod 644 9a5ba576.0
vbox86p:/system/etc/security/cacerts #
```

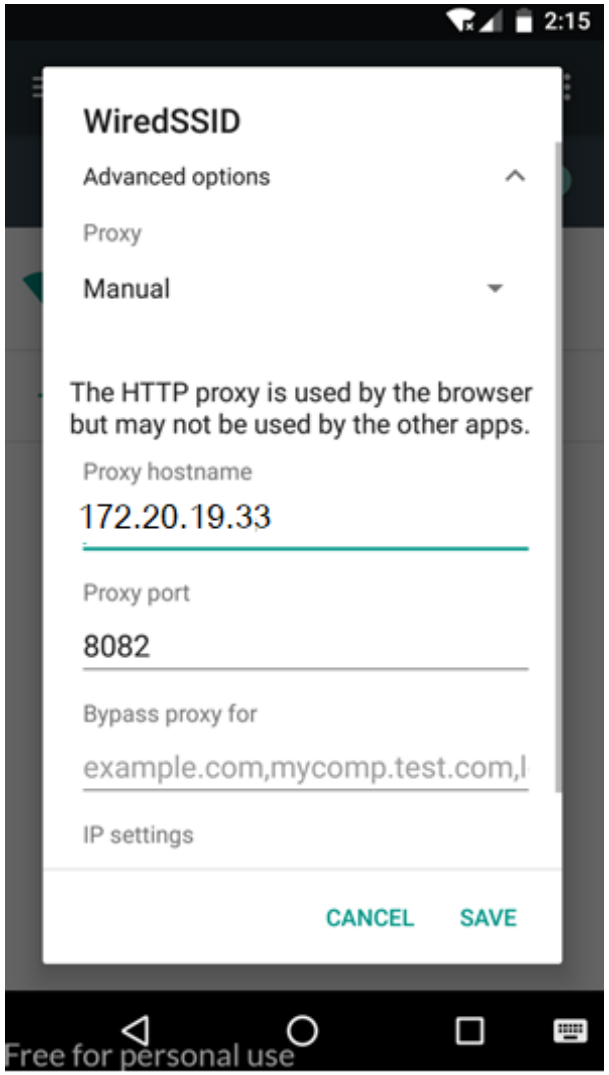
Configuring Android Device to work with Burp Suite

Step 6: Set IP Address of Computer to Android Device







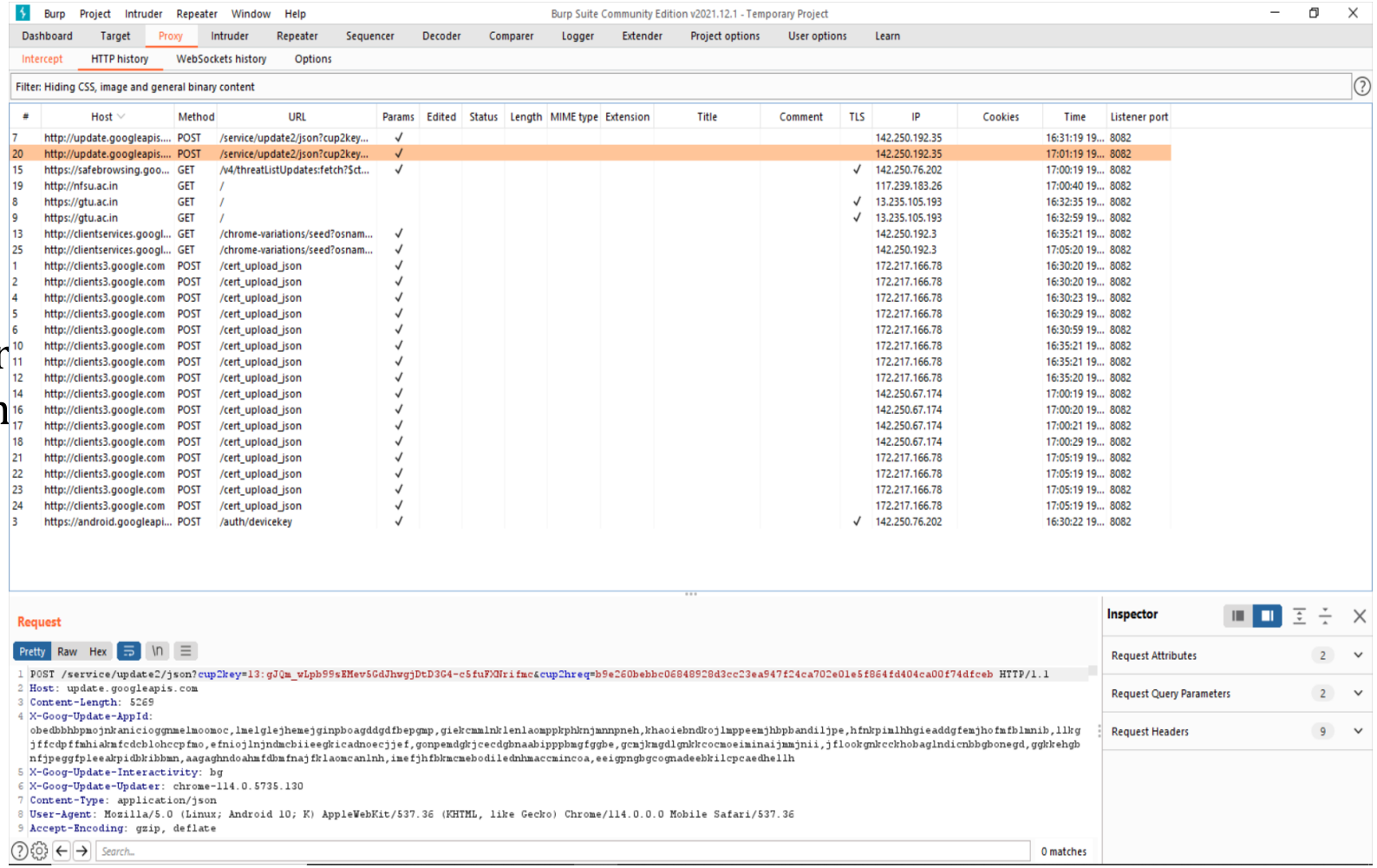


Configuring Android Device to work with Burp Suite

Step-7 Test the configuration

To test the configuration:

1. Open Burp Suite Professional.
2. Go to **Proxy > Intercept** and click **Intercept is off** to switch intercept on.
3. Open the browser on your Android device and go to an HTTPS web page.



Thank You