

# 1. Preparation for running the program

The program can only run on a rooted Android System. Make sure the phone has been rooted.

## 1.1 Download

To run this program, you firstly need to download two tools to the phone. These two tools are:

- Tcpcat <http://www.strazzere.com/android/tcpcat>
- LSOFL <http://www.roman10.net/src/lsofl>

## 1.2 push tools to your phone

To put these tools into your Android phone, you can use ADB commands. First, connect your phone to your PC by USB cable. Then, open the command line tool and run the following command “adb push [source file] [destination file]”, just as shown in the following figure:

```
C:\Users\user>adb push c:/tcpcat /data/local/tcpcat
2130 KB/s (645840 bytes in 0.296s)
```

Make sure you have got the write permission of “/data/local”. Otherwise, you need to use shell command to change the mode of “/data/local” first:

```
C:\Users\user>adb shell
shell@jflte:/ $ su
su
root@jflte:/ # chmod 777 /data/local
chmod 777 /data/local
```

After you push Tcpcat and LSOFL to “/data/local”, you need to change their mode to get them executable:

```
root@jflte:/ # chmod 777 /data/local/tcpcat
chmod 777 /data/local/tcpcat
```

The operation of LSOFL is the same.

You can check if you successfully finish above operation by running these two tool in

ADB shell command:

```
root@jflte:/ # /data/local/tcpdump
/data/local/tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 96 bytes
19:56:08.922669 IP 172.20.168.95.56478 > 172.20.168.113.10201: S 1857642472:1857642472<0> win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
19:56:08.959779 IP 172.20.168.113.10201 > 172.20.168.95.56478: S 2146082949:2146082949<0> ack 1857642473 win 14600 <mss 1460,nop,nop,sackOK,nop,wscale 6>
19:56:08.926484 IP 172.20.168.95.56478 > 172.20.168.113.10201: . ack 1 win 16425
19:56:08.926728 IP 172.20.168.95.56478 > 172.20.168.113.10201: P 1:9<8> ack 1 win 16425
19:56:08.926942 IP 172.20.168.113.10201 > 172.20.168.95.56478: . ack 9 win 229
19:56:08.928132 IP 172.20.168.95.56478 > 172.20.168.113.10201: P 9:163(154) ack 1 win 16425
19:56:08.928315 IP 172.20.168.113.10201 > 172.20.168.95.56478: . ack 163 win 245
19:56:08.926860 IP 172.20.168.113.22438 > resolver.net.ed.ac.uk.domain: 8073* PTR? 95.168.20.172.in-addr.arpa. <44>
19:56:08.940248 IP resolver.net.ed.ac.uk.domain > 172.20.168.113.22438: 8073 NXDomain* 0/1/0 <104>
19:56:08.954347 IP 172.20.168.113.2335 > resolver.net.ed.ac.uk.domain: 47057* PTR? 113.168.20.172.in-addr.arpa. <45>
19:56:08.958619 IP resolver.net.ed.ac.uk.domain > 172.20.168.113.2335: 47057 NXDomain* 0/1/0 <105>
19:56:08.972840 IP 172.20.168.113.10201 > 172.20.168.95.56478: P 1:9<8> ack 163 win 245
19:56:08.973237 IP 172.20.168.113.29783 > resolver.net.ed.ac.uk.domain: 25121* PTR? 191.205.215.129.in-addr.arpa. <46>
```

## 2. Some settings in the code

### 2.1 get particular type of packets

In default setting, Network Warden only capture TCP packet. You can find the line of code in *RunTCP.java* :

```
os.writeBytes("/data/local/tcpdump -v -n -s 0 tcp\n");
```

If you want to capture UDP packets as well, change this code to:

```
os.writeBytes("/data/local/tcpdump -v -n -s 0 tcp or udp\n");
```

### 2.2 network

In default setting, Network Warden captures the packet in WiFi network. You can find this line of code in *HashTable.java*:

```
os.writeBytes("ifconfig wlan0\n");
```

"wlan0" is the network name of the WiFi network. If you want to capture the packets from other network, you need the name of that network. You can find the name of the networks by ADB command:

```
127|root@jflte:/ # netcfg
netcfg
rmnet_smsmxi DOWN
0.0.0.0/0 0x000001002 66:c7:01:8e:6b:af
rmnet_smsmxi DOWN
0.0.0.0/0 0x000001002 3e:70:06:eb:3b:5b
dummy0 DOWN
0.0.0.0/0 0x000000082 a6:ce:68:24:dc:58
wlan0 UP
172.20.168.106/20 0x000001043 40:f3:08:17:04:47
lo UP
127.0.0.1/8 0x000000049 00:00:00:00:00:00
sit0 DOWN
0.0.0.0/0 0x000000080 00:00:00:00:00:00
p2p0 UP
0.0.0.0/0 0x000001003 42:f3:08:17:04:47
rmnet_usb0 DOWN
0.0.0.0/0 0x000000000 00:00:00:00:00:00
rmnet_usb1 DOWN
0.0.0.0/0 0x000000000 00:00:00:00:00:00
rmnet_usb2 DOWN
0.0.0.0/0 0x000000000 00:00:00:00:00:00
```

While the first column is network name. Then, for example, you can replace the code with:

```
os.writeBytes("ifconfig rmnet_usb1\n");
```