

TUTORIAL

**Development of a framework to gather the data
from mobile devices for the purpose of user activity
/ behavior classification**

December 12, 2017

Department of CSIS
BITS Pilani

Team:

Devamalya Hazra [2016H1120169P]

Mohit Agarwal [2016H1120161P]

Rishabh Sharma [2016H1120154P]

Shubham Bansal [2016H1120157P]

Experiment setup

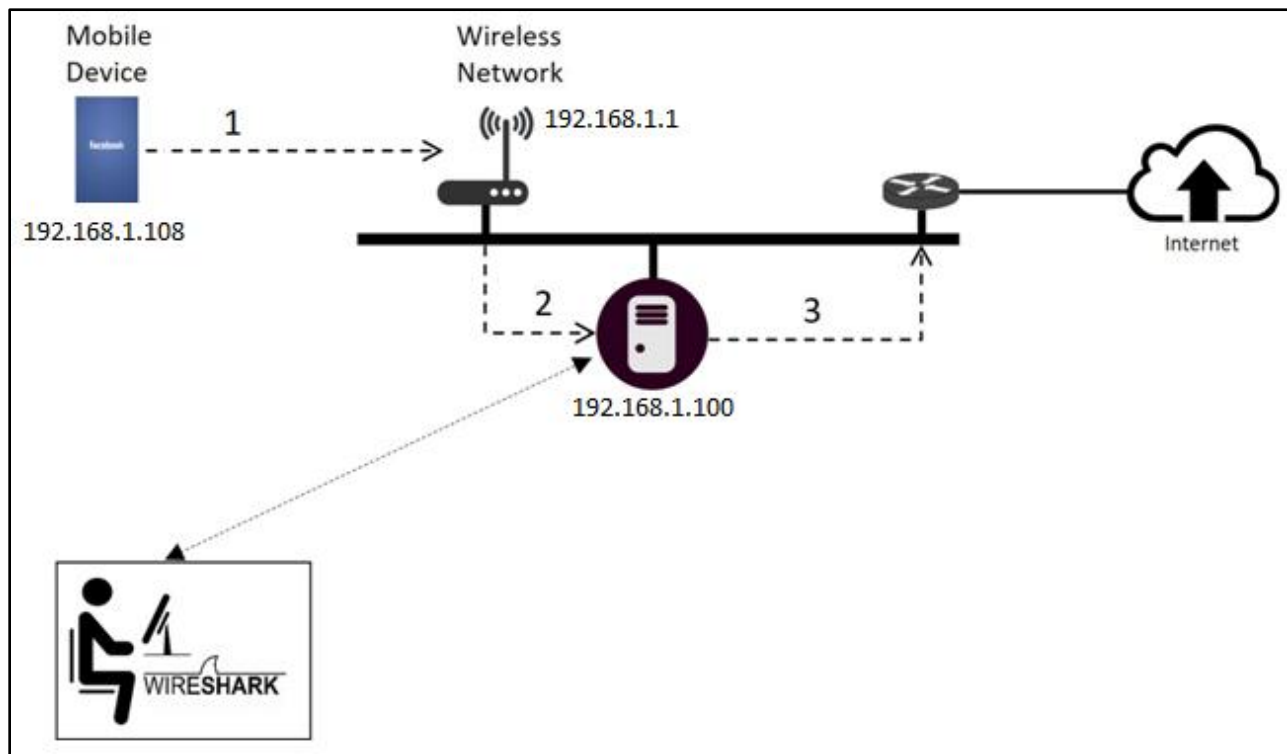


Figure 1: Experiment setup

Above figure shows the experimental setup for our tutorial.

In this tutorial, we are going to use a wifi router, an android phone and a laptop with ubuntu OS.

Router is configured with following DHCP ip range.

IP range: 192.168.1.1 to 192.168.1.254

Subnet: 192.168.1.0/24

So, all the devices - mobile phones, laptop and router are connected in same network. Whole mobile traffic which is generated from mobile devices is routed through laptop (proxy server) and that can be intercepted at this point.

We will start the tutorial with configuration of laptop and mobile devices. First we will setup proxy-server on ubuntu-laptop and then we will use this proxy on our phones for internet.

1. PROXY-SERVER Setup (TinyProxy)

We will use TinyProxy for proxy-server setup. Tinyproxy is a HTTP proxy server daemon for linux operating systems which is designed to be small and fast solution for proxy-server. For our prototype, this is viable solution.

1.>We can install it via apt-get command. Execute following command on terminal:

```
sudo apt-get install tinyproxy
```

2.>We need to configure ip-permissions so that proxy server can allow connections from android phones.

open **/etc/tinyproxy.conf** with root user or [sudo permission] in text editor. Scroll down in file where the default IP rules are mentioned.

Add the IP list (whitelist)

```
//Allow 127.0.0.1  
//Allow 192.168.1.113    //  
//Allow 192.168.1.110    //  
Allow 192.168.1.0/24    // this will include whole network.
```

3.>Now, we can start the proxy Server (background process)

```
sudo /etc/init.d/tinyproxy restart
```

4.>Proxy server's maintains all logs in default location (/var/log/tinyproxy/tinyproxy.log)
We can see the logs related to connection-requests, requested urls, etc.

```
sudo cat /var/log/tinyproxy/tinyproxy.log
```

2. Android phone configuration

We will configure proxy and restrict background settings.

- **Proxy Settings**

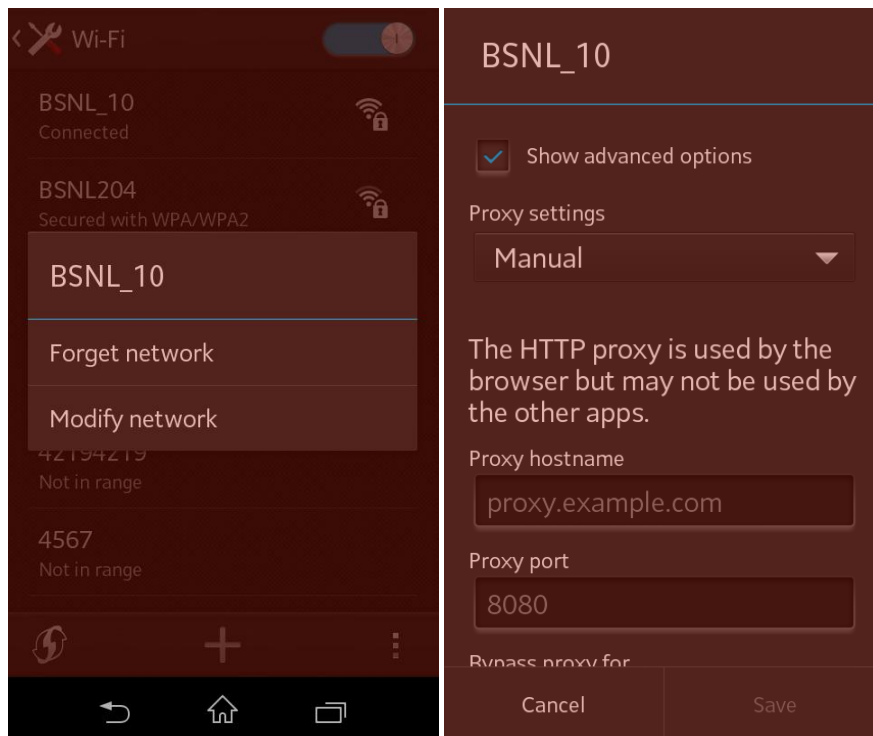
Open wifi from settings.

Set proxy setting which is available under advanced options of wifi-connection:

Proxy settings: Manual

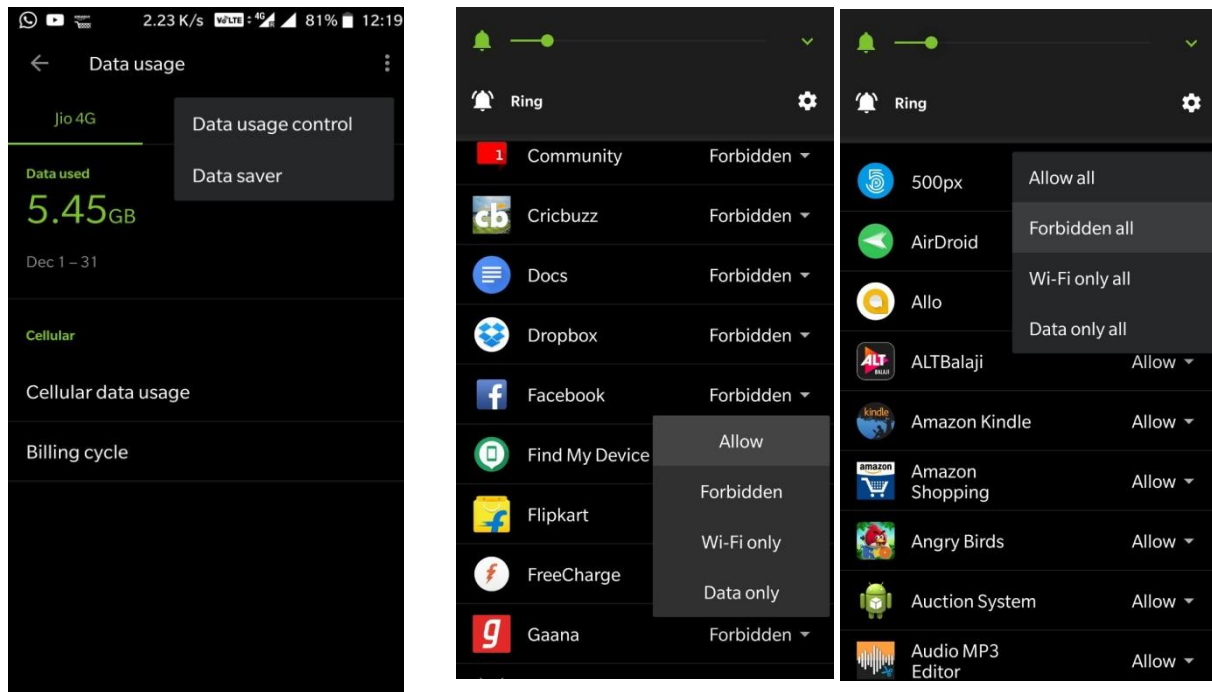
Proxy hostname: 192.168.1.100

Proxy port: 8888



- **Restrict background**

Go to **Settings => data usage => on the top right of the screen**



3. Wireshark (Packet capture)

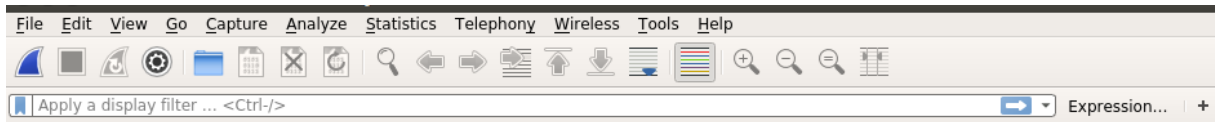
We have configured laptop with proxy-server and android phone with proxy setting and background data restrictions. Now. We are ready for the packet capturing part. We will use wireshark (Network sniffer tool).

Install Wireshark via apt-get.

```
sudo apt-get install wireshark
```

Open wireshark with sudo permission.

```
sudo wireshark
```

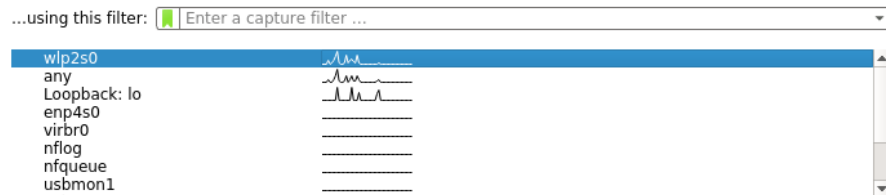


Welcome to Wireshark

Open

/home/shubham/Desktop/pcap/pcap_data/gmail.pcapng (407 KB)
/home/shubham/Desktop/pcap/pcap_data/amazon.pcapng (4003 KB)
/home/shubham/Desktop/pcap/filtered.pcap.pcapng (23 MB)
/home/shubham/Desktop/pcap/facebook_2.pcap.pcapng (24 MB)
/home/shubham/Downloads/2017_10_25_150557.pcap (11 MB)
/home/shubham/Desktop/pcap/facebook.pcap.pcapng (not found)

Capture



Select wifi-interface (wlp2s0)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help									
Apply a display filter ... <Ctrl-/>									
Expression...									
No.	Time	Source	Destination	Protocol	Length	Info			
91	8.028976219	91.189.91.23	192.168.1.100	TCP	1514	80 → 56666 [ACK] Seq=1461 Ack=1 Win=237 Len=...			
92	8.029000929	192.168.1.100	91.189.91.23	TCP	54	56666 → 80 [ACK] Seq=1 Ack=2921 Win=616 Len=0			
93	8.029085125	91.189.91.23	192.168.1.100	TCP	2254	80 → 56666 [PSH, ACK] Seq=2921 Ack=1 Win=237...			
94	8.029109440	192.168.1.100	91.189.91.23	TCP	54	56666 → 80 [ACK] Seq=1 Ack=5121 Win=651 Len=0			
95	8.029274316	91.189.91.23	192.168.1.100	TCP	1514	80 → 56666 [ACK] Seq=5121 Ack=1 Win=237 Len=...			
96	8.029305737	192.168.1.100	91.189.91.23	TCP	54	56666 → 80 [ACK] Seq=1 Ack=6581 Win=674 Len=0			
97	8.029325384	91.189.91.23	192.168.1.100	TCP	1514	80 → 56666 [ACK] Seq=6581 Ack=1 Win=237 Len=...			
98	8.029338453	192.168.1.100	91.189.91.23	TCP	54	56666 → 80 [ACK] Seq=1 Ack=8041 Win=696 Len=0			
99	8.029577200	91.189.91.23	192.168.1.100	TCP	1514	80 → 56666 [ACK] Seq=8041 Ack=1 Win=237 Len=...			
100	8.029600143	192.168.1.100	91.189.91.23	TCP	54	56666 → 80 [ACK] Seq=1 Ack=9501 Win=719 Len=0			
101	8.029944087	91.189.91.23	192.168.1.100	TCP	774	80 → 56666 [PSH, ACK] Seq=9501 Ack=1 Win=237...			
102	8.029966738	192.168.1.100	91.189.91.23	TCP	54	56666 → 80 [ACK] Seq=1 Ack=10221 Win=742 Len=...			
103	8.304092448	192.168.1.108	192.168.1.100	TCP	74	46144 → 8888 [SYN] Seq=0 Win=65535 Len=0 MSS...			
104	8.304166537	192.168.1.100	192.168.1.108	TCP	74	8888 → 46144 [SYN, ACK] Seq=0 Ack=1 Win=2896...			
105	8.304173705	192.168.1.108	192.168.1.100	TCP	74	46145 → 8888 [SYN] Seq=0 Win=65535 Len=0 MSS...			
106	8.304194619	192.168.1.100	192.168.1.108	TCP	74	8888 → 46145 [SYN, ACK] Seq=0 Ack=1 Win=2896...			
107	8.306721092	192.168.1.108	192.168.1.100	TCP	66	46144 → 8888 [ACK] Seq=1 Ack=1 Win=87808 Len=...			
108	8.306763017	192.168.1.108	192.168.1.100	TCP	66	46145 → 8888 [ACK] Seq=1 Ack=1 Win=87808 Len=...			
109	8.307062783	192.168.1.100	172.24.2.71	DNS	86	Standard query 0x72e0 PTR 108.1.168.192.in-a...			
110	8.307109542	192.168.1.100	172.24.2.71	DNS	86	Standard query 0xdc39 PTR 108.1.168.192.in-a...			
111	8.307972453	172.24.2.71	192.168.1.100	DNS	163	Standard query response 0xdc39 No such name ...			
112	8.307996004	172.24.2.71	192.168.1.100	DNS	163	Standard query response 0x72e0 No such name ...			
113	8.312667629	192.168.1.108	192.168.1.100	HTTP	163	CONNECT graph.facebook.com:443 HTTP/1.1			
114	8.312707595	192.168.1.100	192.168.1.108	TCP	66	8888 → 46144 [ACK] Seq=1 Ack=98 Win=29056 Le...			
115	8.312718757	192.168.1.108	192.168.1.100	HTTP	163	CONNECT graph.facebook.com:443 HTTP/1.1			
116	8.312741674	192.168.1.100	192.168.1.108	TCP	66	8888 → 46145 [ACK] Seq=1 Ack=98 Win=29056 Le...			
117	8.519436073	192.168.1.100	172.24.2.71	DNS	78	Standard query 0x13e3 A graph.facebook.com			
118	8.519475885	192.168.1.100	172.24.2.71	DNS	78	Standard query 0x89c5 A graph.facebook.com			
119	8.519502332	192.168.1.100	172.24.2.71	DNS	78	Standard query 0x98aa AAAA graph.facebook.com			
▶ Frame 113: 163 bytes on wire (1304 bits), 163 bytes captured (1304 bits) on interface 0									
▶ Ethernet II, Src: Oneplus_d3:79:38 (c0:ee:fb:d3:79:38), Dst: Sony_6a:74:a8 (54:42:49:6a:74:a8)									
▶ Internet Protocol Version 4, Src: 192.168.1.108, Dst: 192.168.1.100									
▶ Transmission Control Protocol, Src Port: 46144, Dst Port: 8888, Seq: 1, Ack: 1, Len: 97									
▶ Hypertext Transfer Protocol									
0000	54 42 49 6a 74 a8 c0 ee	fb d3 79 38 08 00 45 00	TBIjt... .y8..E.						
0010	00 95 64 36 40 00 40 06	52 0c c0 a8 01 6c c0 a8	..d6@.@. R....l..						
0020	01 64 b4 40 22 b8 14 be	d6 5c f6 9d 67 5f 80 18	.d.@"... ..\..g...						
0030	01 57 be 2d 00 00 01 01	08 0a 00 e8 b2 70 00 20	.W.-.... ..p.						
0040	de 25 43 4f 4e 4e 45 43	54 20 67 72 61 70 68 2e	.%CONNEC T graph.						
0050	66 61 63 65 62 6f 6f 6b	2e 63 6f 6d 3a 34 3a 33	facebook .com:443						
0060	20 48 54 54 50 2f 31 2e	31 0d 0a 48 6f 73 74 3a	HTTP/1. 1..Host:						

Now, in above picture we can see all the wifi network traffic. We can easily track packets related to the mobile application. In this case we are using facebook on phone and packet number 113 confirms the application (CONNECT graph.facebook.com).

This captured data contains all communication between android phone and facebook. It consists packets of http, dns, TLSv1.2, tcp, etc.

We will be capturing the n/w traffic for few minutes then we will stop and export data to pcap file. In next steps, we will use this pcap file for the analysis.

4. Packet Filtration

This is data pre-processing step. In this step, we will remove unwanted packets (like: non-IP, proxy-server background data)

Ex:

non-IP: SSDP, ARP

ubuntu repository data: us.archive.ubuntu.com, security.ubuntu.com,etc.

We will use following python code ({Project_dir}/**filter_packet_code.py**) for this step.

```
from scapy.allimport*

#reading from pcap file
packets = rdpcap('facebook_2.pcap.pcapng')
filtered_packets = PacketList()

# Filter-Rules:

# 91.189.91.13, 91.189.91.14, 91.189.91.15, more ==> us.archive.ubuntu.com
# 91.189.88.161, 91.189.92.181, 91.189.92.200 ==> security.ubuntu.com
# 91.189.95.83 ==> ppa.launchpad.net
# 91.189.92.152 ==> extras.ubuntu.com

ignore_list
=['239.255.255.250','91.189.88.161','91.189.92.181','91.189.91.13','91.189.91.14','91.189.91.15','91.189
.95.83','91.189.92.152']

num =-1

for pk in packets:

    num = num +1

    ifnot pk.haslayer(IP):

        print'packet without IP: ', num

        continue

    ifnot(pk['IP'].src in ignore_list or pk['IP'].dst in ignore_list):

        filtered_packets.append(pk)
```



```

#writing to pcap file
wrpcap('filtered.pcap.pcapng',filtered_packets)

print'Packets count in original pcap = ',len(packets)
print'Packets count in filtered pcap = ',len(filtered_packets)

```

this will generate a filtered pcap file.

5. Feature extraction

Now, we will use above filtered pcap file for feature extraction.

We will use following python code ({Project_dir}/**flow_code.py**) for this step.

First we extract flow.

```

'''
@Parameter 1: packetList object
@Parameter 2: output pickleFile path, this is optional parameter. If valid path exists then function will
return flowList by de-serialising the pickle file data

return: flowList which is list of list object.. ==> [ [srcIP, dstIP], [srcPort, dstPort], [packetIndex1,
packetIndex2,.....]]
'''

def extractFlows(packets, outputPickleFile=None):

if outputPickleFile !=Noneand os.path.exists(outputPickleFile):
fp=open('flows.pik','rb')
flowList = pickle.load(fp)
fp.close()

```

```
return flowList

num =0
flowList =[]

for pk in packets:
    if pk.haslayer(IP)and pk.haslayer(TCP):
        src = pk['IP'].src
        dst = pk['IP'].dst
        sport = pk['IP'].sport
        dport = pk['IP'].dport

#rules
#Ex: filter protocol, remove ack, threshold on minimum packets,

#aggregating packets into flow
if len(flowList)==0:
    flowList.append([[src,dst],[sport,dport],[num]])
else:
    flag =False
    for flow in flowList:
        if src in flow[0]and dst in flow[0]and sport in flow[1]and dport in flow[1]:
            flow[2].append(num)
            flag =True
    break
ifnot flag:
    flowList.append([[src,dst],[sport,dport],[num]])
```

```

num +=1

#writing flowList into pickle
if outputPickleFile !=None:
    fout =open(outputPickleFile,"wb")
    pickle.dump(flowList, fout )
    fout.close()

return flowList

```

Now, we extract 22 features for each flow.

```

'''
@Parameter 1: packetList object
@Parameter 2: flowList which is list of list object.. ==> [ [srcIP, dstIP], [srcPort, dstPort], [packetIndex1,
packetIndex2,.....]]

return: function return the feature vector which is list of numeric values
'''

def extractFeatures(packets, flow):

    #creating featureVector list
    featureVector =[]

    #Adding features to feature vector

    #1
    totalPacketCount = feature_TotalPackets(packets, flow)

```

```
featureVector.append(totalPacketCount)
```

#2

```
avgPacketSize = feature_AveragePacketSize(packets, flow)
```

```
featureVector.append(avgPacketSize)
```

#3

```
minfpktl, maxfpktl, meanfpktl, stdfpktl, fpackets, fbytes, minfiat, maxfiat, meanfiat, stdfiat =  
DirectionFeatures(packets, flow, "Forward")
```

```
featureVector.append(minfpktl)
```

```
featureVector.append(maxfpktl)
```

```
featureVector.append(meanfpktl)
```

```
featureVector.append(stdfpktl)
```

```
featureVector.append(fpackets)
```

```
featureVector.append(fbytes)
```

```
featureVector.append(minfiat)
```

```
featureVector.append(maxfiat)
```

```
featureVector.append(meanfiat)
```

```
featureVector.append(stdfiat)
```

```
minbpktl, maxbpktl, meanbpktl, stdbpktl, bpackets, bbytes, minbiat, maxbiat, meanbiat, stdbiat =  
DirectionFeatures(packets, flow, "Backward")
```

```
featureVector.append(minbpktl)
```

```
featureVector.append(maxbpktl)
```

```
featureVector.append(meanbpktl)
```

```
featureVector.append(stdbpktl)
```

```
featureVector.append(bpackets)
```

```
featureVector.append(bbytes)
```

```

featureVector.append(minbiat)

featureVector.append(maxbiat)

featureVector.append(meanbiat)

featureVector.append(stdbiat)


return featureVector

```

Output:

```

shubham@shubham@node0: /var/log/tinyproxy$ flow_code.py
parsing complete...
('Total packets = ', 35765)
('Total no flows = ', 158)
shubham@node0:~/Desktop/pcap$

```

>> `cat {Project_folder}/out_features.csv`

```

facebook,26,156.115384615,52,298,168.0,121.028095912,15,2533,-1.93347574198,2.7658
facebook,4,113.0,52,320,186.0,134.0,2,372,5.80230355263e-05,5.80230355263e-05,5.80
facebook,4,100.5,52,270,161.0,109.0,2,322,3.65759730339e-05,3.65759730339e-05,3.65
facebook,24,142.25,40,286,145.0,121.739182564,14,2036,-1.93458503199,2.76874491102
facebook,14,98.9285714286,52,270,117.0,96.7699776347,7,820,4.4243991375e-05,1.5713
facebook,14,106.071428571,52,320,131.0,119.340449375,7,920,2.47520208359e-05,1.474
facebook,13,67.3076923077,40,221,77.0,62.0886463051,7,540,0.000900973021984,0.0640
facebook,13,69.2307692308,40,221,80.0,63.9385866058,7,565,0.00076639598608,0.12792
facebook,47,723.404255319,40,2960,1436.0,568.251628281,23,33040,-0.896489958048,2.
facebook,18,101.666666667,40,467,111.0,122.716747023,10,1110,-0.748293097019,0.434
facebook,16,102.8125,40,467,117.0,127.684071921,9,1058,-1.19041515601,0.3707019959
facebook,16,87.25,40,455,97.0,127.580040236,9,877,-0.755388317049,0.433376551986,0
facebook,13,98.1538461538,40,455,113.0,140.411639729,7,797,-1.19568071997,0.292060
facebook,18,101.666666667,40,467,111.0,122.716747023,10,1110,-2.28213872403,0.6270
facebook,16,102.8125,40,467,117.0,127.684071921,9,1058,-3.09156250399,0.6082511360
facebook,1447,669.525224603,52,2948,407.0,675.16828114,789,321655,-3.93343553001,2
facebook,21,98.2380952381,52,467,106.0,117.674435317,11,1174,-2.29226303399,0.8199
facebook,16,87.25,40,455,97.0,127.580040236,9,877,-2.33087508005,0.425637515008,-0
facebook,1275,742.249411765,40,14640,480.0,1384.6625431,642,308178,-4.10595768899,
facebook,14,84.0,40,455,104.0,122.504067007,9,827,-2.11200702,0.420715770006,-0.20

```