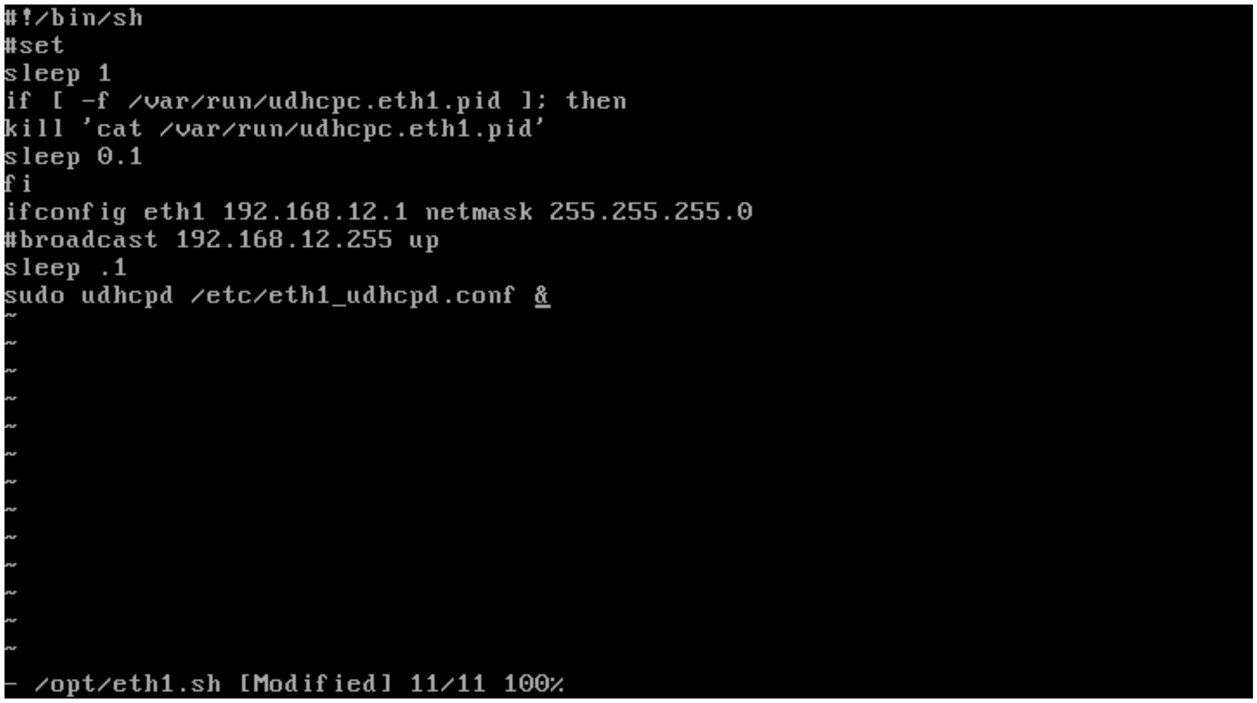
CS4516
Final Report
Team22
Ying Zhang & Yujia Qiu
04/28/2019

Overview: This project consist of traffic capture, analysis and classification. The TinyCore VM was configured as a gateway for Android VM. Pyshark was used to gather information and capture network packets as they are in transit through the gateway. Then a machine learning model for application identification was trained by collect traffic samples from a set of different types of applications. The trained classifier was then used to analyze live traffic, and perform live traffic classification.

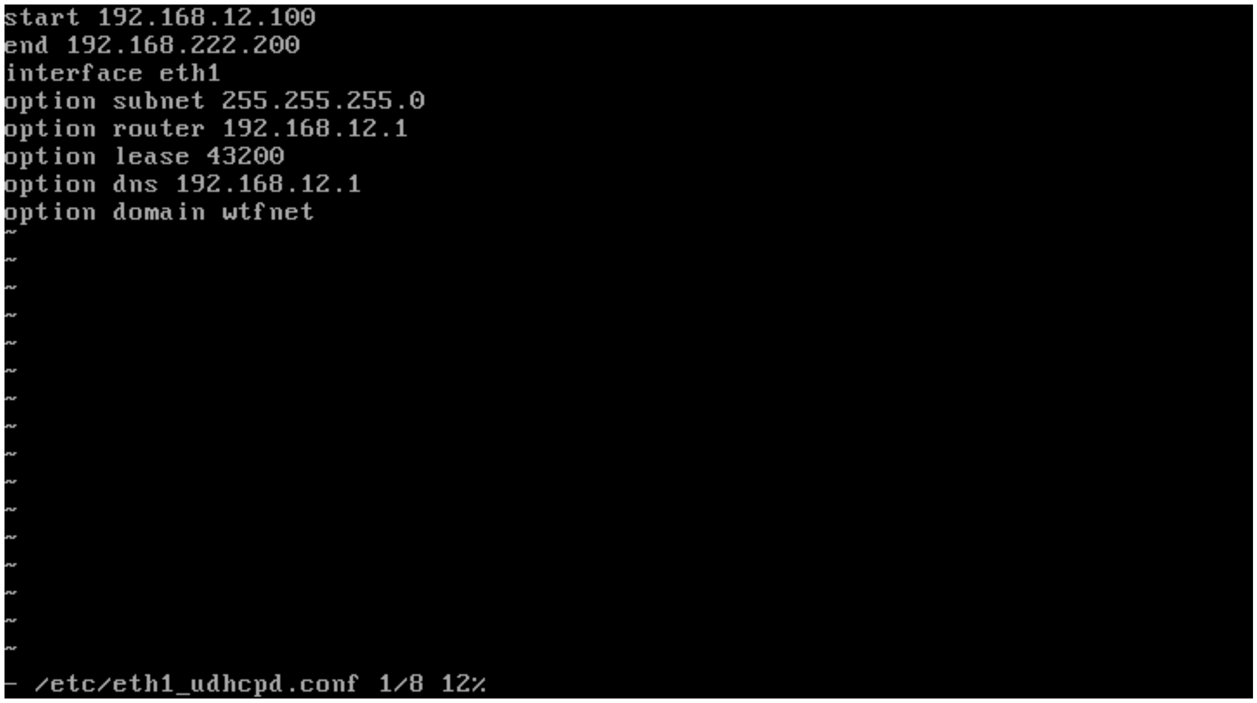
In **Phase1** , we configured the TinyCore VM so that it can act as gateway for Android VM. The gateway was configured as follows:

1. Configure interface eth1 by creating a new script file using “sudo vi /opt/eth1.sh” command, inside the file, type the following content:



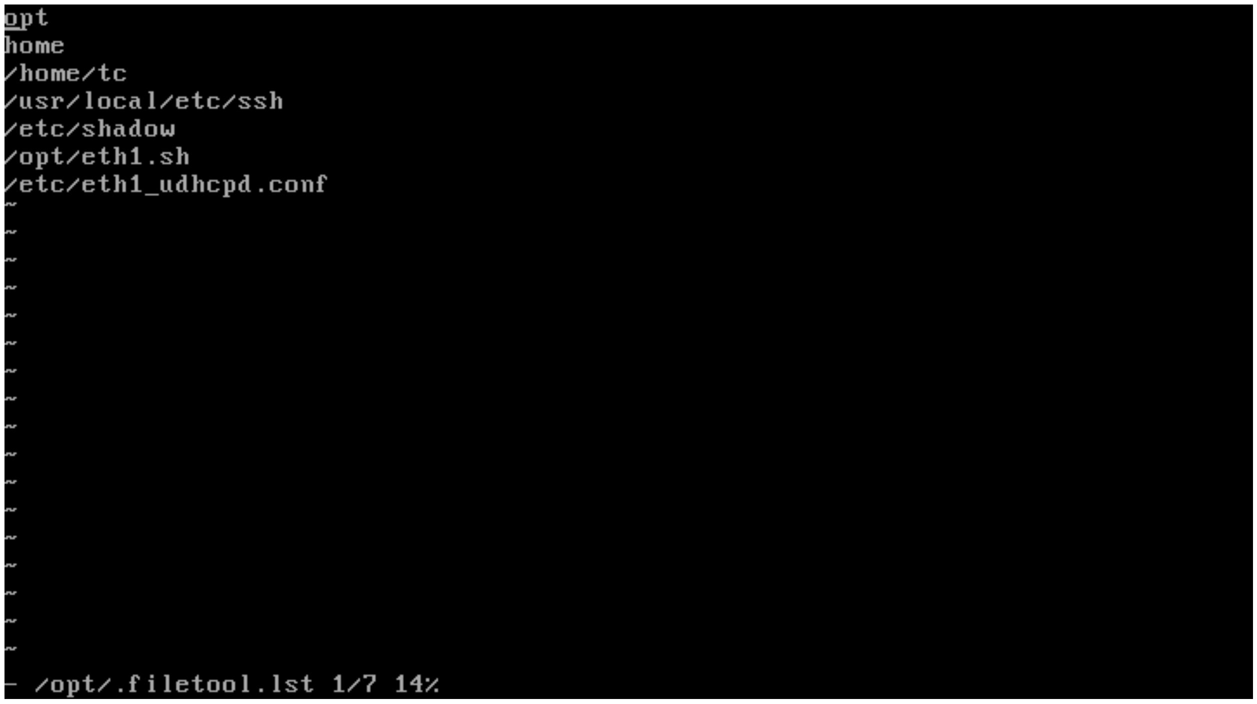
```
#!/bin/sh
#set
sleep 1
if [ -f /var/run/udhcpc.eth1.pid ]; then
kill 'cat /var/run/udhcpc.eth1.pid'
sleep 0.1
fi
ifconfig eth1 192.168.12.1 netmask 255.255.255.0
#broadcast 192.168.12.255 up
sleep .1
sudo udhcpd /etc/eth1_udhcpd.conf &
~
~
~
~
~
~
~
~
~
~
~
~
~
- /opt/eth1.sh [Modified] 11/11 100%
```

2. Configure udhcp.conf



```
start 192.168.12.100
end 192.168.222.200
interface eth1
option subnet 255.255.255.0
option router 192.168.12.1
option lease 43200
option dns 192.168.12.1
option domain wtfnet
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
- /etc/eth1_udhcpd.conf 1/8 12%
```

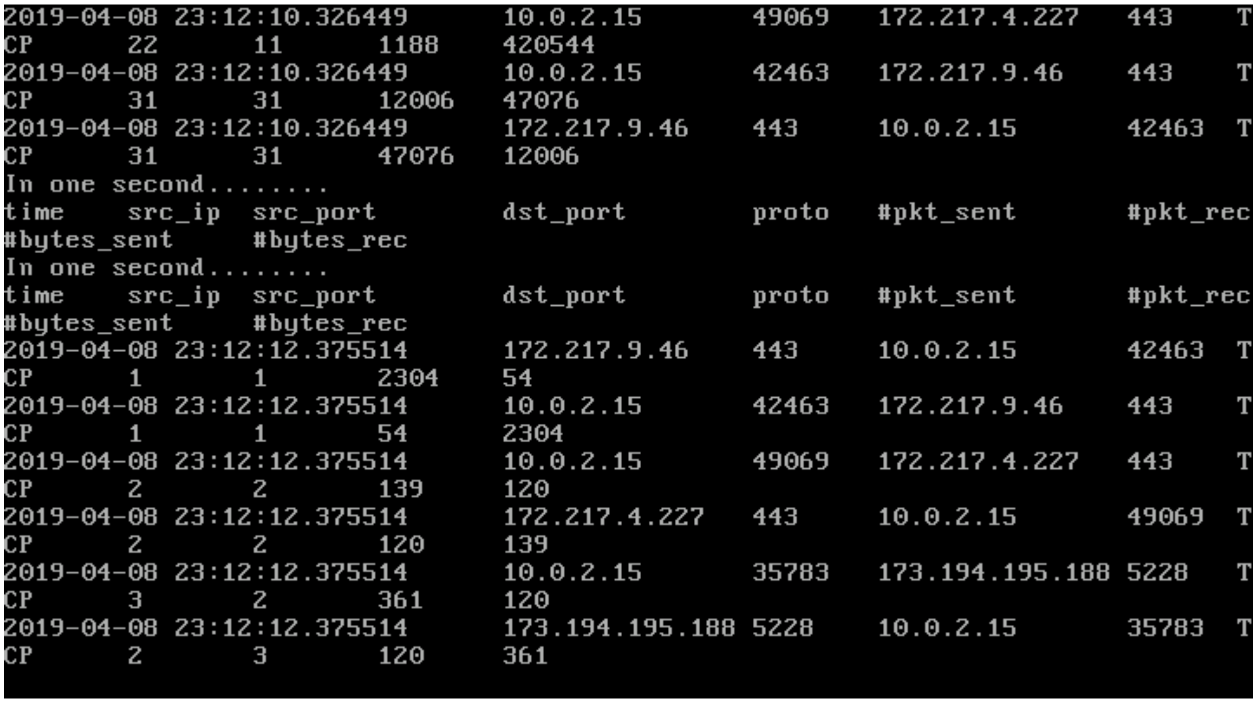
3. udhcpd /etc/eth1_udhcpd.conf
4. sudo sysctl net.ipv6.ip_forward(change the value to 1 sudo sysctl -w net.ipv6.ip_forward=1)
5. Rules about iptables
    1. *iptables -F*
    2. *iptables -A FORWARD -s 192.168.12.0/24 -j ACCEPT*
    3. *iptables -A FORWARD -d 192.168.12.0/24 -j ACCEPT*
    4. *iptables -t nat -A POSTROUTING -s 192.168.12.0/24 -o eth0 -j SNAT --to-source 10.0.2.15*
6. Change interface script file permission (rewrite several files in /opt)



```
opt
home
/home/tc
/usr/local/etc/ssh
/etc/shadow
/opt/eth1.sh
/etc/eth1_udhcpd.conf
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
- /opt/.filetool.lst 1/7 14%
```

7. Backup-make the Ethernet interface configuration persistent

The core of **Phase 2** is to use pyshark to capture the information of the network interface card. We used “pyshark.LiveCapture(interface='en0', bpf_filter='ip and tcp’)” to get information about “en0” and its ip and TCP type packets. Then we user “sniff(100)” to deal with the first 100 packet. The next step is to encapsulate gathered information, which include the number of received and sent packets number, byte number, IP and Port. The result is stored in a dictionary (Key-Value set). The “Key” is the send IP + port + received IP + port. The “Value” is the encapsulated information. For every packet gathered from pyshark, we get its send & received IP and port, and we get two values from dictionary: assume we set send IP as IP1, received IP as IP2(ignore the port), then for a “Key” as IP1 + IP2, we changed the number of bytes it send. Then for a “Key” as IP2 + IP1, we changed the number of bytes it received. We also used an extra thread to print the result every second, then clear the result so that we could gather information about the next second. Below is a screenshot for phase2.



```
2019-04-08 23:12:10.326449      10.0.2.15       49069   172.217.4.227   443     T
CP      22      11      1188    420544
2019-04-08 23:12:10.326449      10.0.2.15       42463   172.217.9.46    443     T
CP      31      31      12006   47076
2019-04-08 23:12:10.326449      172.217.9.46    443     10.0.2.15       42463   T
CP      31      31      47076   12006
In one second........
time    src_ip  src_port        dst_port        proto   #pkt_sent       #pkt_rec
#bytes_sent     #bytes_rec
In one second........
time    src_ip  src_port        dst_port        proto   #pkt_sent       #pkt_rec
#bytes_sent     #bytes_rec
2019-04-08 23:12:12.375514      172.217.9.46    443     10.0.2.15       42463   T
CP      1       1       2304    54
2019-04-08 23:12:12.375514      10.0.2.15       42463   172.217.9.46    443     T
CP      1       1       54      2304
2019-04-08 23:12:12.375514      10.0.2.15       49069   172.217.4.227   443     T
CP      2       2       139     120
2019-04-08 23:12:12.375514      172.217.4.227   443     10.0.2.15       49069   T
CP      2       2       120     139
2019-04-08 23:12:12.375514      10.0.2.15       35783   173.194.195.188 5228    T
CP      3       2       361     120
2019-04-08 23:12:12.375514      173.194.195.188 5228    10.0.2.15       35783   T
CP      2       3       120     361
```

In **Phase3**, the classification algorithm we use is RandomForest, a technique based on the bagging. RandomForest algorithm works as a large collection of de-correlated decision trees. A lot of decision trees were used to make a classfication. We train a random forest classifier on the feature vectors as follows:
* Protocol (1 for TCP, 0 for UDP)
* Byte ratio (bytes sent / bytes received, or reciprocal)
* Packet ratio (packets sent / packets received, or reciprocal)
* Mean packet length

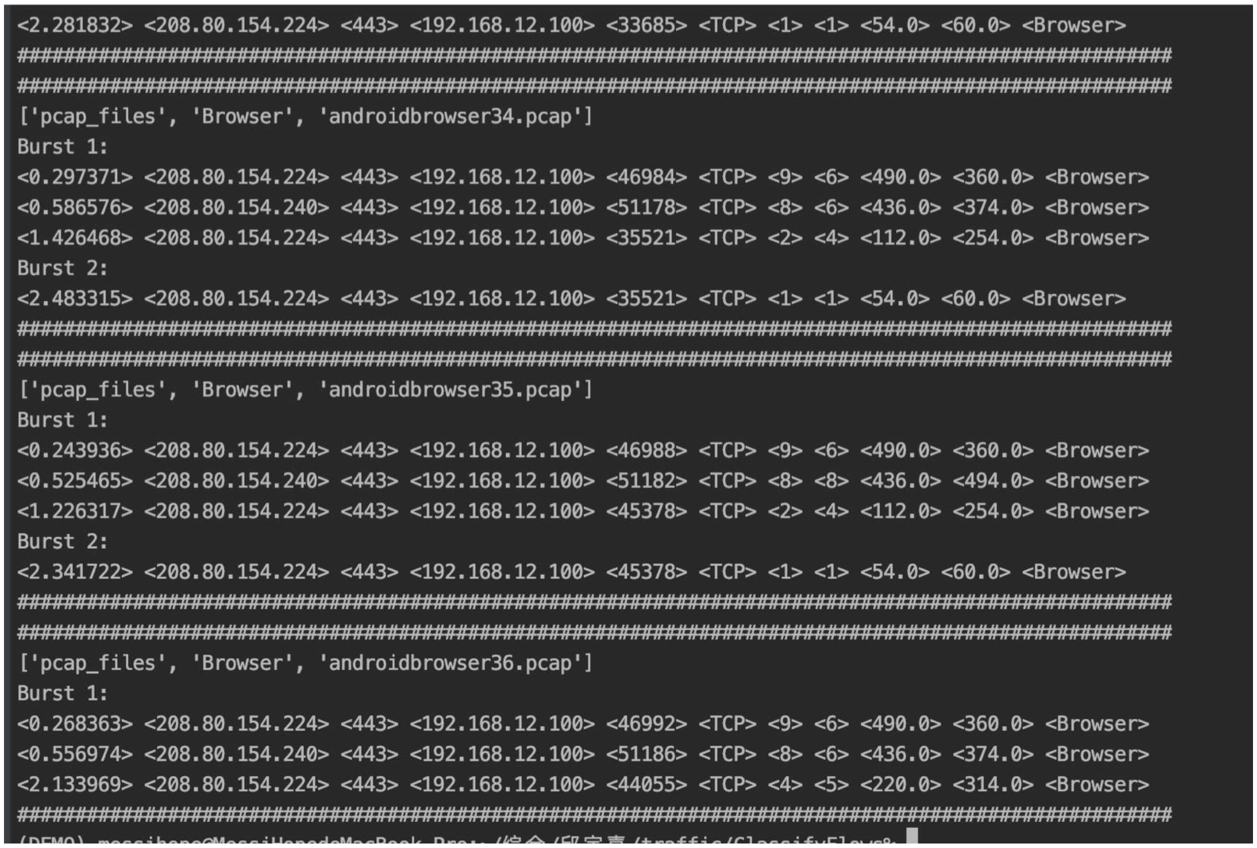
* Min packet length
* Max packet length
* Standard deviation of packet lengths (zero if n <= 2)
* Packet length kurtosis
* Packet length skew
* Mean time gap between each packet (zero if n <= 1)
* Min time gap
* Max time gap
* Standard deviation of packet lengths (zero if n <= 2)
* Time gap kurtosis
* Time gap skew

50 traces were used in total, 25 for training and the other 25 for evaluation. There are 858 flows in total, 429 for training and the other 429 for evaluation.With only 50 traces in total, we didn’t expect it can work well. Here are the results:

The accuracies of the test dataset are shown as follows:

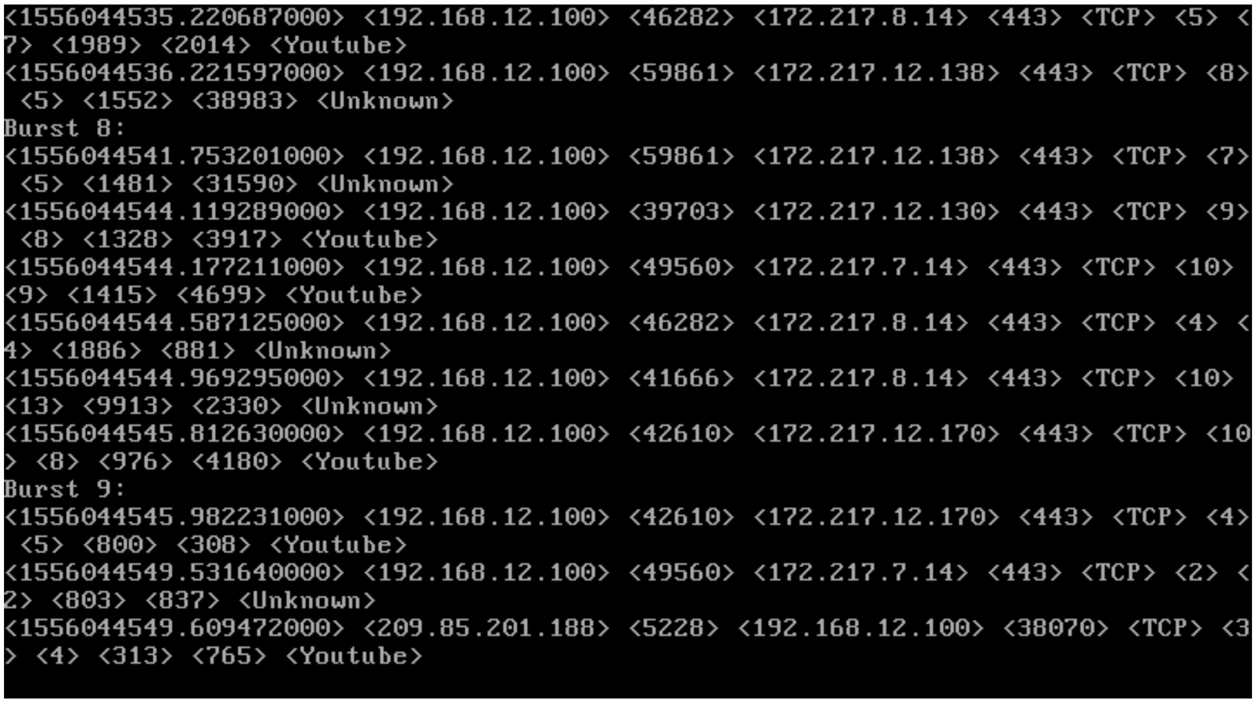
```
          precision    recall  f1-score   support

   Browser      0.83      0.75      0.79        20
     Fruit      0.70      0.84      0.76       169
      News      0.78      0.39      0.52        18
   Weather      0.74      0.70      0.72       155
   Youtube      0.82      0.63      0.71        67

avg / total     0.74      0.73      0.73       429
```

Another thing we didn't expect is that quite a part of examples were predicted as “Unknown". Maybe it is caused by the similarity between the feature of different classes. Below is a screenshot for phase3.



```
<2.281832> <208.80.154.224> <443> <192.168.12.100> <33685> <TCP> <1> <1> <54.0> <60.0> <Browser>
############################################################################################
############################################################################################
['pcap_files', 'Browser', 'androidbrowser34.pcap']
Burst 1:
<0.297371> <208.80.154.224> <443> <192.168.12.100> <46984> <TCP> <9> <6> <490.0> <360.0> <Browser>
<0.586576> <208.80.154.240> <443> <192.168.12.100> <51178> <TCP> <8> <6> <436.0> <374.0> <Browser>
<1.426468> <208.80.154.224> <443> <192.168.12.100> <35521> <TCP> <2> <4> <112.0> <254.0> <Browser>
Burst 2:
<2.483315> <208.80.154.224> <443> <192.168.12.100> <35521> <TCP> <1> <1> <54.0> <60.0> <Browser>
############################################################################################
############################################################################################
['pcap_files', 'Browser', 'androidbrowser35.pcap']
Burst 1:
<0.243936> <208.80.154.224> <443> <192.168.12.100> <46988> <TCP> <9> <6> <490.0> <360.0> <Browser>
<0.525465> <208.80.154.240> <443> <192.168.12.100> <51182> <TCP> <8> <8> <436.0> <494.0> <Browser>
<1.226317> <208.80.154.224> <443> <192.168.12.100> <45378> <TCP> <2> <4> <112.0> <254.0> <Browser>
Burst 2:
<2.341722> <208.80.154.224> <443> <192.168.12.100> <45378> <TCP> <1> <1> <54.0> <60.0> <Browser>
############################################################################################
############################################################################################
['pcap_files', 'Browser', 'androidbrowser36.pcap']
Burst 1:
<0.268363> <208.80.154.224> <443> <192.168.12.100> <46992> <TCP> <9> <6> <490.0> <360.0> <Browser>
<0.556974> <208.80.154.240> <443> <192.168.12.100> <51186> <TCP> <8> <6> <436.0> <374.0> <Browser>
<2.133969> <208.80.154.224> <443> <192.168.12.100> <44055> <TCP> <4> <5> <220.0> <314.0> <Browser>
############################################################################################
(DEMO) massihepo@MassiHepodeMacBook-Pro:~/综合/段宇嘉/traffic/ClassifyFlows%
```

In phase4, we implemented live traffic classification by configured the TinyCore gateway to identify applications generating the traffic from the Android VM. We integrated the classifier into the traffic logger as follows: at first, packets captured by the LiveCapture object are placed into flows. When a burst occurs, feature vector includes Protocol, Byte ratio, Packet ratio, Mean packet length, Min packet length, Max packet length and so on will be extracted according to some information of the flows. After that, the pre-trained classifier will predict the class the current flow belongs. Below is a screenshot for phase4.



```
<1556044535.220687000> <192.168.12.100> <46282> <172.217.8.14> <443> <TCP> <5> <
7> <1989> <2014> <Youtube>
<1556044536.221597000> <192.168.12.100> <59861> <172.217.12.138> <443> <TCP> <8>
 <5> <1552> <38983> <Unknown>
Burst 8:
<1556044541.753201000> <192.168.12.100> <59861> <172.217.12.138> <443> <TCP> <7>
 <5> <1481> <31590> <Unknown>
<1556044544.119289000> <192.168.12.100> <39703> <172.217.12.130> <443> <TCP> <9>
 <8> <1328> <3917> <Youtube>
<1556044544.177211000> <192.168.12.100> <49560> <172.217.7.14> <443> <TCP> <10>
<9> <1415> <4699> <Youtube>
<1556044544.587125000> <192.168.12.100> <46282> <172.217.8.14> <443> <TCP> <4> <
4> <1886> <881> <Unknown>
<1556044544.969295000> <192.168.12.100> <41666> <172.217.8.14> <443> <TCP> <10>
<13> <9913> <2330> <Unknown>
<1556044545.812630000> <192.168.12.100> <42610> <172.217.12.170> <443> <TCP> <10
> <8> <976> <4180> <Youtube>
Burst 9:
<1556044545.982231000> <192.168.12.100> <42610> <172.217.12.170> <443> <TCP> <4>
 <5> <800> <308> <Youtube>
<1556044549.531640000> <192.168.12.100> <49560> <172.217.7.14> <443> <TCP> <2> <
2> <803> <837> <Unknown>
<1556044549.609472000> <209.85.201.188> <5228> <192.168.12.100> <38070> <TCP> <3
> <4> <313> <765> <Youtube>
```