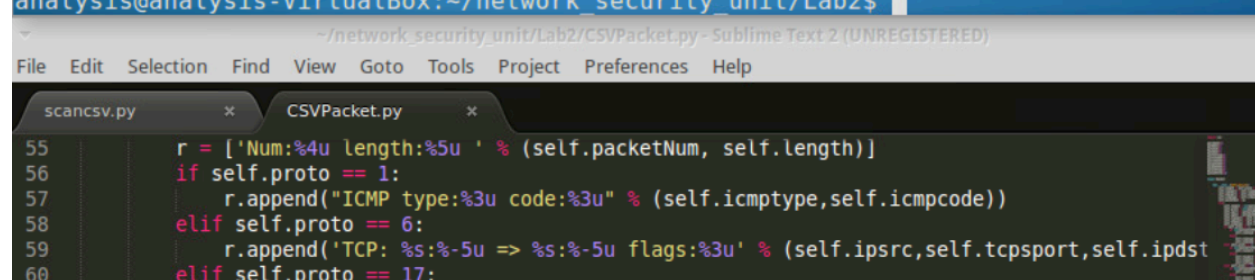


Student: Joaquin Saldana
Week 6 / Lab 2
CS373 – Defense Against the Dark Arts
Winter 2018

In this week's lab we were asked to examine network traffic found in 2 .csv files. In order to analyze the data in the .csv files, we were asked to write code in a Python or Perl script to help us dissect the data.

While doing an initial run of the scancsv.py Python script to the R.csv file (the smaller file), I found it provided me the following data:

```
analysis@analysis-VirtualBox:~/network_security_unit/Lab2$ python scancsv.py R.csv
numPackets:99142 numBytes:71683046
1:      7
2:      2
6:    39138
17:   59995
analysis@analysis-VirtualBox:~/network_security_unit/Lab2$
```

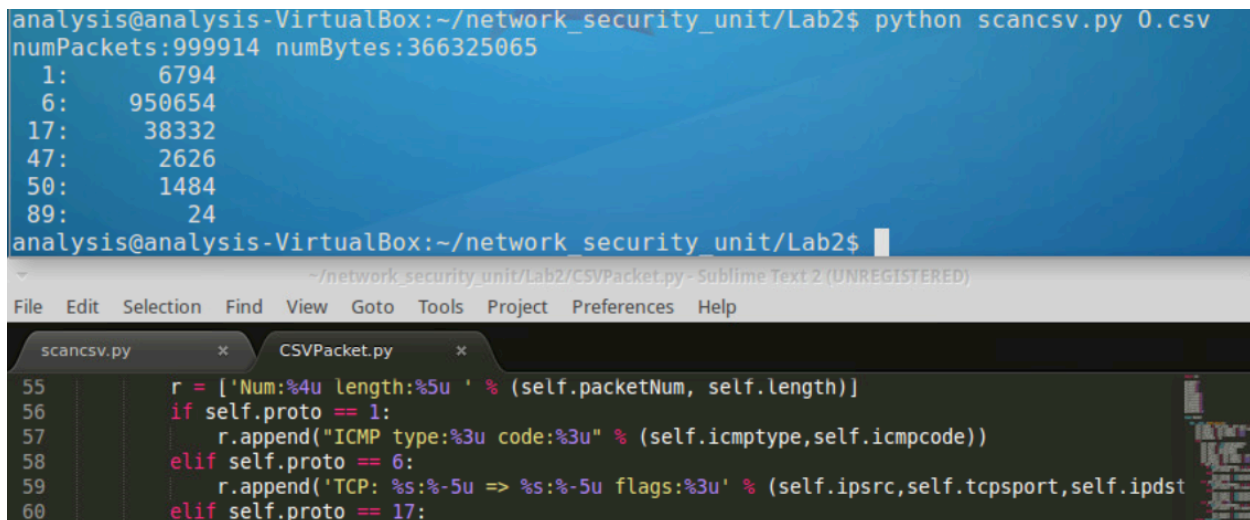


In doing some research and with the aid for the linked site (<https://technet.microsoft.com/en-us/library/cc959827.aspx> and <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>), I found that protocol 1 means ICMP (Internet Control Message Protocol), 2 equals IGMP (Internet Group Management), 6 equals TCP (Transmission Control Protocol) and 17 equals UDP (User Datagram Protocol).

Clearly from the data above the majority of the traffic captured is TCP and UDP, also the more commonly used network protocols.

I ran the same script against the O.csv file, since this file is much larger, and the following results were returned:

```
analysis@analysis-VirtualBox:~/network_security_unit/Lab2$ python scancsv.py 0.csv
numPackets:999914 numBytes:366325065
1: 6794
6: 950654
17: 38332
47: 2626
50: 1484
89: 24
analysis@analysis-VirtualBox:~/network_security_unit/Lab2$
```

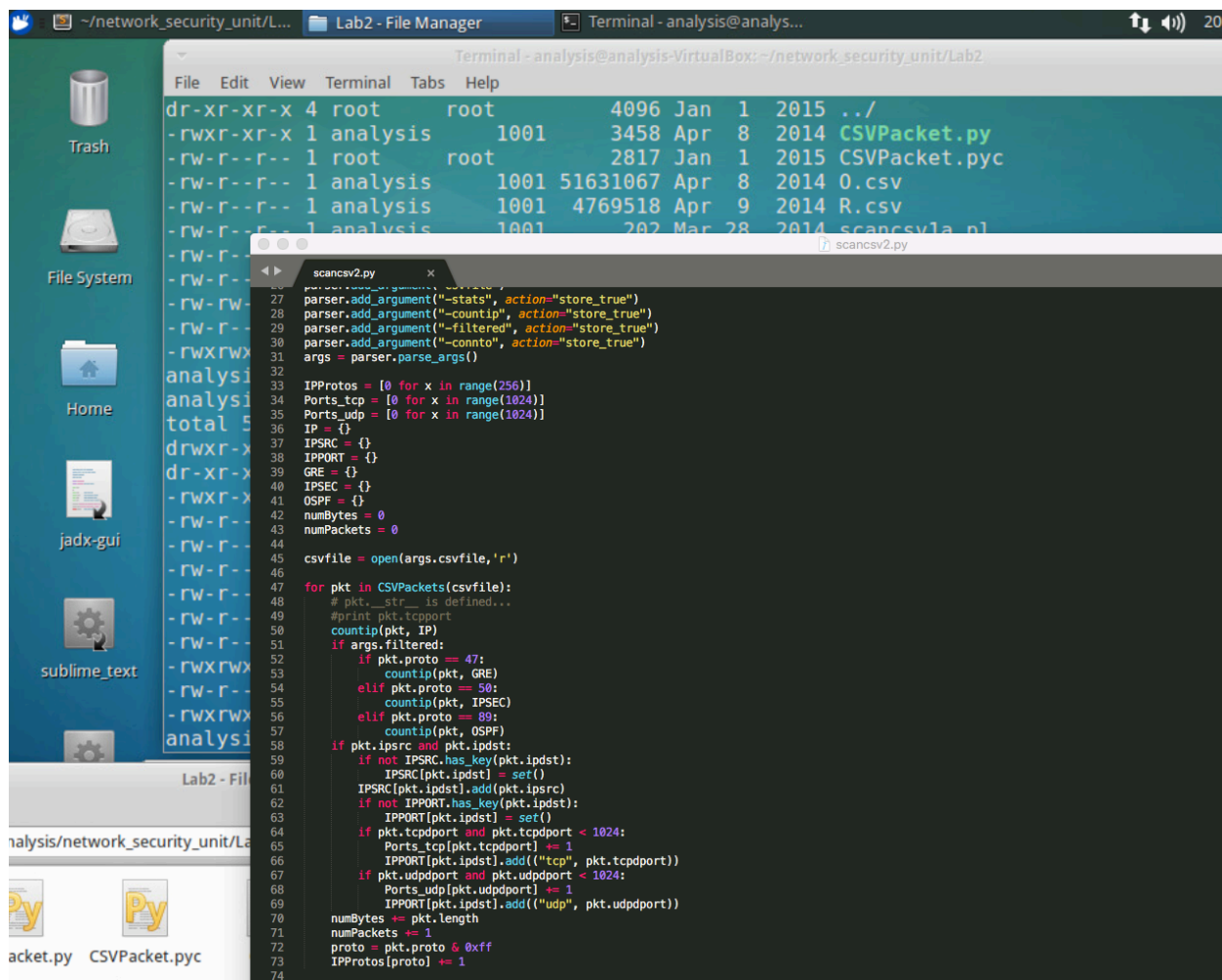


The image shows a terminal window and a Sublime Text editor. The terminal window displays the output of a Python script named 'scancsv.py' which processes a CSV file '0.csv'. The output shows the total number of packets (999914) and bytes (366325065), followed by a list of protocols and their counts: 1: 6794, 6: 950654, 17: 38332, 47: 2626, 50: 1484, and 89: 24. The Sublime Text editor shows the code for 'CSVPacket.py', which is a class that processes network packet data and appends it to a list 'r' based on the protocol number.

The meta data in the O.csv included protocols now found in the R.csv file. The protocols in particular were:

- 47 which equals GRE (Generic Routing Encapsulation)
- 50 which equals ESP (Encap Security Payload)
- 89 which means OSPFIGP which stands for Open Shortest Path First Interior Gateway Protocol which is a type of protocol used for exchanging routing information between routers within an autonomous system.

Afterwards I started to write my script with instructions as stated in the lab for commands such as -stats, -countip, -filtered and more



```
dr-xr-xr-x 4 root root 4096 Jan 1 2015 ../
-rwxr-xr-x 1 analysis 1001 3458 Apr 8 2014 CSVPacket.pyc
-rw-r--r-- 1 root 2817 Jan 1 2015 CSVPacket.pyc
-rw-r--r-- 1 analysis 1001 51631067 Apr 8 2014 O.csv
-rw-r--r-- 1 analysis 1001 4769518 Apr 9 2014 R.csv
-rw-r--r-- 1 analysis 1001 202 Mar 28 2014 scancsv1a.pl
-rw-r--r-- 1 analysis 1001 202 Mar 28 2014 scancsv2.py
```

```
scancsv2.py
27 parser.add_argument("-stats", action="store_true")
28 parser.add_argument("-countip", action="store_true")
29 parser.add_argument("-filtered", action="store_true")
30 parser.add_argument("-connto", action="store_true")
31 args = parser.parse_args()
32
33 IPProtos = [0 for x in range(256)]
34 Ports_tcp = [0 for x in range(1024)]
35 Ports_udp = [0 for x in range(1024)]
36 IP = {}
37 IPSRC = {}
38 IPPORT = {}
39 GRE = {}
40 IPSEC = {}
41 OSPF = {}
42 numBytes = 0
43 numPackets = 0
44
45 csvfile = open(args.csvfile, 'r')
46
47 for pkt in CSVPackets(csvfile):
48     # pkt.__str__ is defined...
49     #print pkt.tcport
50     countip(pkt, IP)
51     if args.filtered:
52         if pkt.proto == 47:
53             countip(pkt, GRE)
54         elif pkt.proto == 50:
55             countip(pkt, IPSEC)
56         elif pkt.proto == 89:
57             countip(pkt, OSPF)
58     if pkt.ipsrc and pkt.ipdst:
59         if not IPSRC.has_key(pkt.ipdst):
60             IPSRC[pkt.ipdst] = set()
61         IPSRC[pkt.ipdst].add(pkt.ipsrc)
62         if not IPPORT.has_key(pkt.ipdst):
63             IPPORT[pkt.ipdst] = set()
64         if pkt.tcport and pkt.tcport < 1024:
65             Ports_tcp[pkt.tcport] += 1
66             IPPORT[pkt.ipdst].add(("tcp", pkt.tcport))
67         if pkt.udport and pkt.udport < 1024:
68             Ports_udp[pkt.udport] += 1
69             IPPORT[pkt.ipdst].add(("udp", pkt.udport))
70     numBytes += pkt.length
71     numPackets += 1
72     proto = pkt.proto & 0xff
73     IPProtos[proto] += 1
74
```

In my tests I found that the top 5 ports were Netbios sessions, HTTP, POP3, SSH, and DNS for the R.csv file. In the O.csv file I found that SMTP, HTTP, and DNS.

Also, in the R.csv there several private IP addresses as both the source and destination (addresses starting with 10.*). This suggests this is logs for a large network of internal machines exchanging information.

Also in the O.csv file, the largest set of connections were made to addresses starting at IP address 192.245 with TCP connections at ports 25, 135, and 22.