Lab #6 Report: Network Management

CSC432 – Computer & Network Security

Franklin Nuth

27 February 2018

Abstract

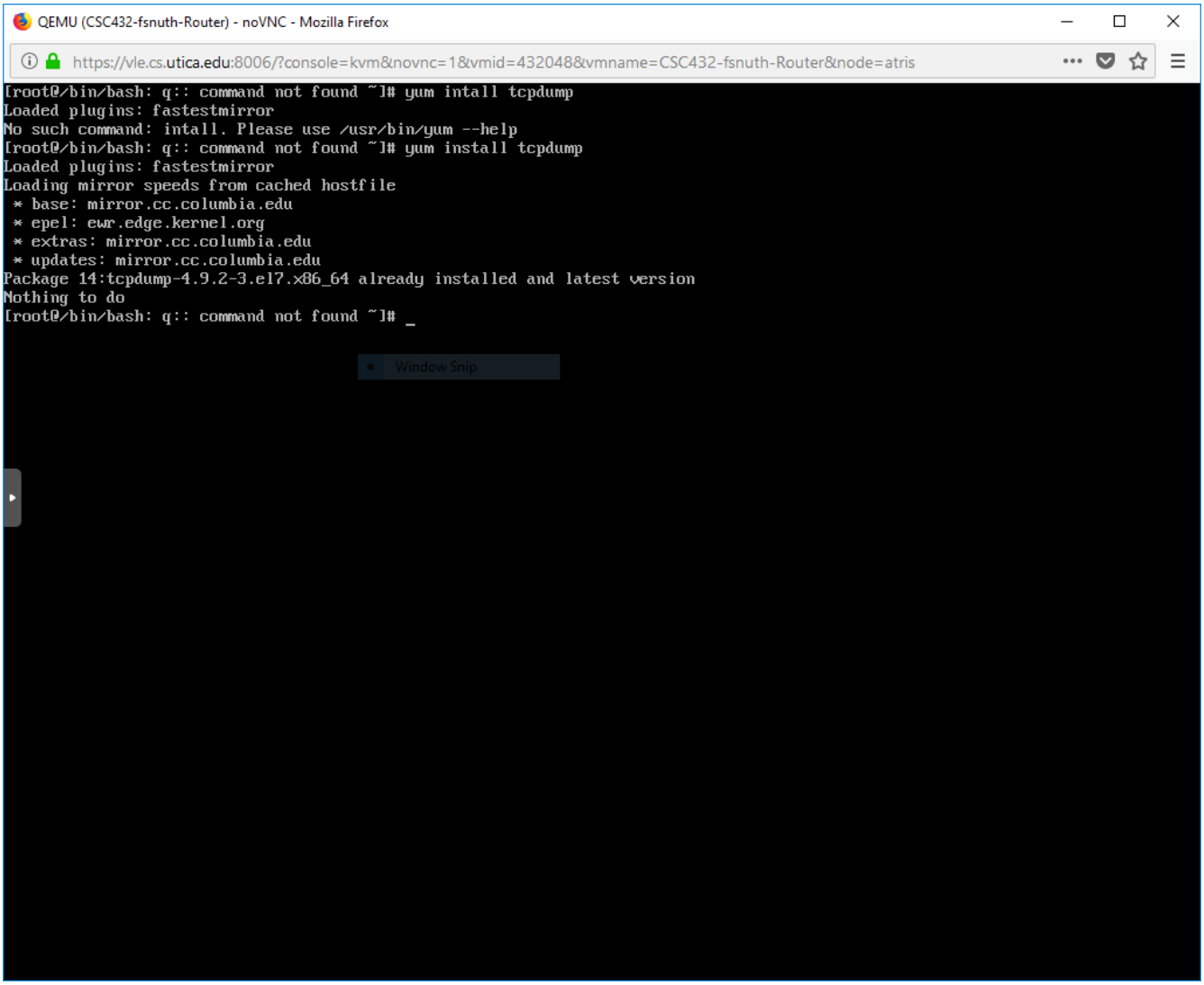
In this lab, I will install **tcpdump** and use it to monitor the network traffic on my internal and external interfaces. The type of traffic that I will monitor includes **SSH** and **HTTP** packets that interacted with the router. In order to study network flow in real time, I will also discuss how to download and use a tool called **argus**. The tool will be used to show how specific **IP** addresses in a network can be listened to for manual supervision of incoming and outgoing traffic. The lab will also include a small demo by one of argus's data clients called **ratop**, which can display data from a network flow.

Introduction

Wireshark is one of the most effective packet sniffers that is free to download for Windows and Linux users. It is used by both security and hackers in equal extremes for monitoring and analyzing traffic traveling through a network. Although not an intrusion detection system, its ability to read all types of packet formats and capture live traffic makes it a powerful analysis tool for individuals of all skill levels. Unfortunately, the its capabilities can't be utilized by the Cent OS 7 router. In order for the router to imitate the monitoring capabilities of Wireshark, a spectrum of programs must be installed before monitoring traffic from the Linux router can be possible.

Processes & Screenshots

In order to start sniffing traffic immediately, I first have to download **tcpdump** in the router. I typed "yum install tcpdump" into the command line so I can get the codes I need to use it.



(Downloading <u>tcpdump</u>. According the last couple of lines, I already installed it.)

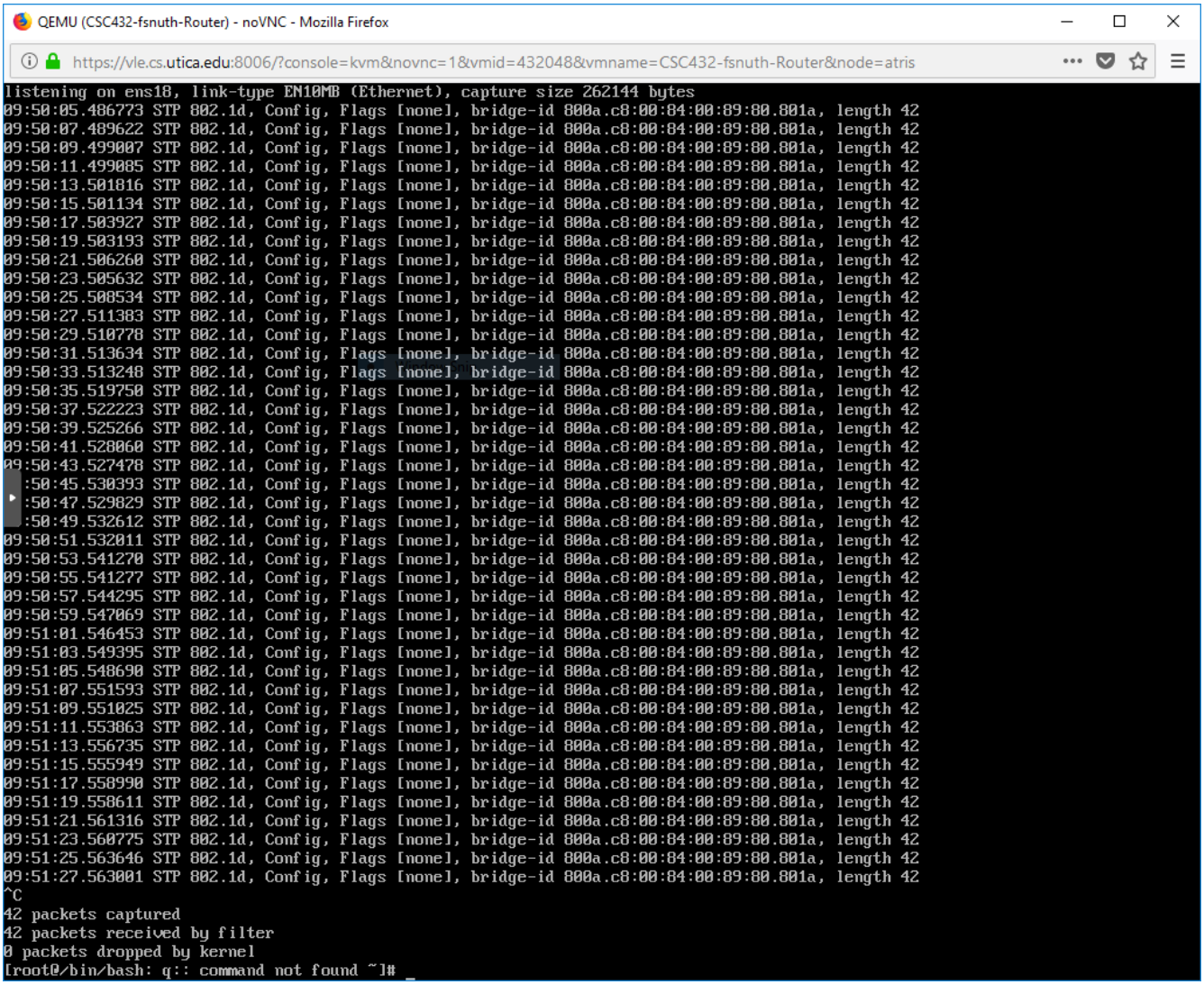
After installing tcpdump, my goals are to capture traffic going through my private interface and my public interface. For the publc interface, I typed “tcpdump -i ens19” and for the private interface I typed “tcpdump -i ens18”.

QEMU (CSC432-fsnuth-Router) - noVNC - Mozilla Firefox

https://vle.cs.utica.edu:8006/?console=kvm&novnc=1&vmid=432048&vmname=CSC432-fsnuth-Router&node=atris

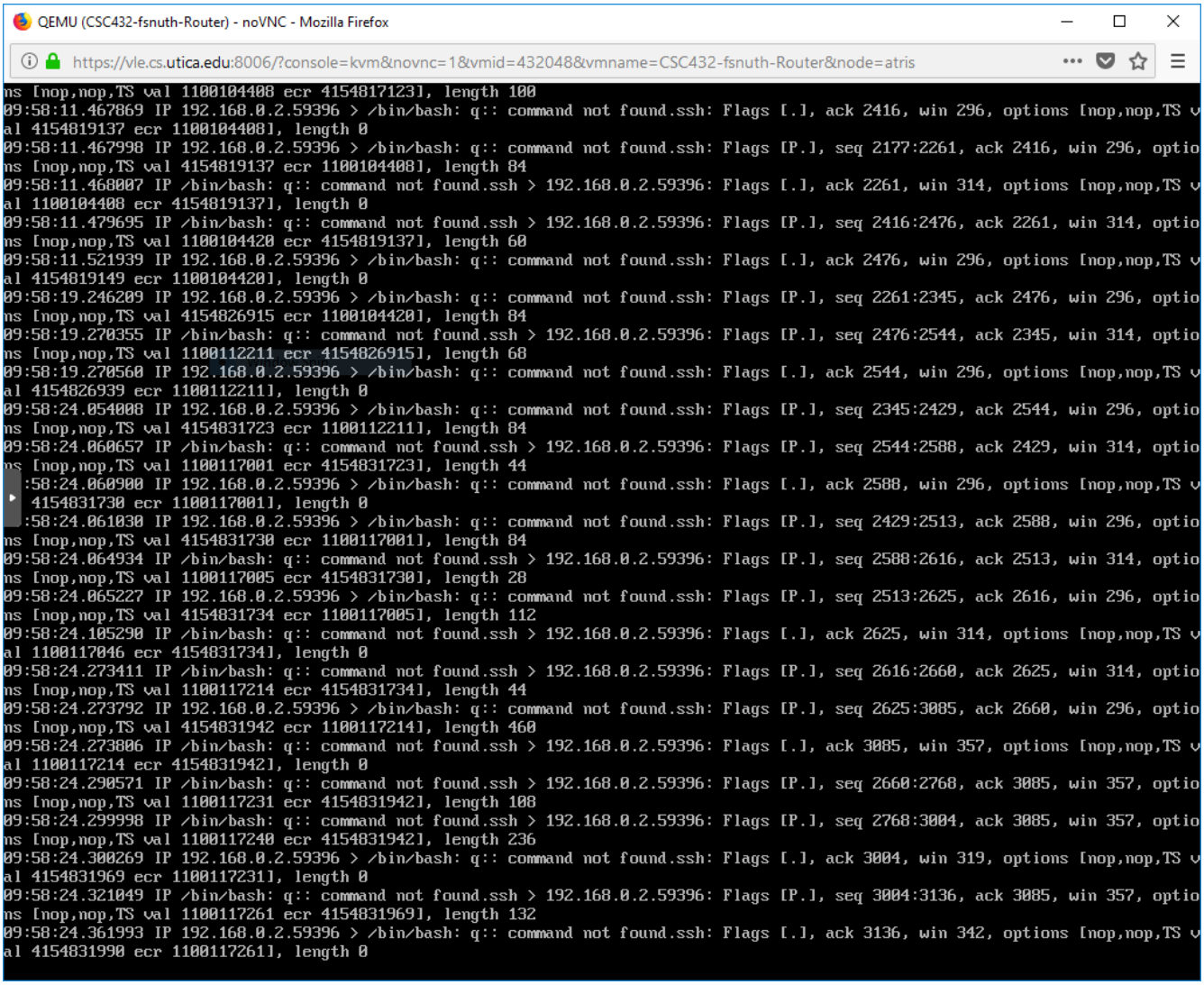
```
ns [nop,nop,TS val 669600104 ecr 1173143312], length 0
09:47:24.571979 IP 192.168.0.2.40118 > yyz12s05-in-f3.1e100.net.http: Flags [.], ack 1411, win 251, options [nop,nop,TS val 9650
11033 ecr 978271406], length 0
09:47:24.588527 IP yyz12s05-in-f3.1e100.net.http > 192.168.0.2.40118: Flags [.], ack 855, win 244, options [nop,nop,TS val 97828
1646 ecr 964929117], length 0
09:47:24.828130 IP 192.168.0.2.46502 > 72.21.91.29.http: Flags [.], ack 2364, win 266, options [nop,nop,TS val 3806920406 ecr 91
2048187], length 0
09:47:24.845134 IP 72.21.91.29.http > 192.168.0.2.46502: Flags [.], ack 1294, win 290, options [nop,nop,TS val 912058427 ecr 380
6869268], length 0
09:47:25.116649 IP 192.168.0.2.41174 > iad23s25-in-f14.1e100.net.https: Flags [P.], seq 16476:16522, ack 21040, win 606, options
 [nop,nop,TS val 3550922158 ecr 1376828106], length 46
09:47:25.116799 IP 192.168.0.2.34808 > server-52-85-105-53.jfk1.r.cloudfront.net.https: Flags [P.], seq 868:914, ack 4134, win 3
28, options [nop,nop,TS val 208758368 ecr 1656506896], length 46
09:47:25.131717 IP server-52-85-105-53.jfk1.r.cloudfront.net.https > 192.168.0.2.34808: Flags [P.], seq 4134:4180, ack 914, win
126, options [nop,nop,TS val 1656512796 ecr 208758368], length 46
09:47:25.131919 IP 192.168.0.2.34808 > server-52-85-105-53.jfk1.r.cloudfront.net.https: Flags [.], ack 4180, win 328, options [n
op,nop,TS val 208758384 ecr 1656512796], length 0
09:47:25.133011 IP iad23s25-in-f14.1e100.net.https > 192.168.0.2.41174: Flags [P.], seq 21040:21086, ack 16522, win 376, options
 [nop,nop,TS val 1376887107 ecr 3550922158], length 46
09:47:25.133186 IP 192.168.0.2.41174 > iad23s25-in-f14.1e100.net.https: Flags [.], ack 21086, win 606, options [nop,nop,TS val 3
550922175 ecr 1376887107], length 0
:47:25.596014 IP 192.168.0.2.40116 > yyz12s05-in-f3.1e100.net.http: Flags [.], ack 1411, win 251, options [nop,nop,TS val 9650
058 ecr 2180862112], length 0
:47:25.612253 IP yyz12s05-in-f3.1e100.net.http > 192.168.0.2.40116: Flags [.], ack 855, win 244, options [nop,nop,TS val 21808
72352 ecr 964930324], length 0
09:47:26.620049 IP 192.168.0.2.34894 > server-52-85-105-41.jfk1.r.cloudfront.net.https: Flags [.], ack 362929, win 4316, options
 [nop,nop,TS val 1345561898 ecr 442655537], length 0
09:47:26.635206 IP server-52-85-105-41.jfk1.r.cloudfront.net.https > 192.168.0.2.34894: Flags [.], ack 1170, win 126, options [n
op,nop,TS val 442656561 ecr 1345459762], length 0
09:47:28.117730 IP 192.168.0.2.53388 > server-52-85-105-138.jfk1.r.cloudfront.net.https: Flags [P.], seq 816:862, ack 5200, win
334, options [nop,nop,TS val 998415321 ecr 442649502], length 46
09:47:28.132376 IP server-52-85-105-138.jfk1.r.cloudfront.net.https > 192.168.0.2.53388: Flags [P.], seq 5200:5246, ack 862, win
 126, options [nop,nop,TS val 442655403 ecr 998415321], length 46
09:47:28.132571 IP 192.168.0.2.53388 > server-52-85-105-138.jfk1.r.cloudfront.net.https: Flags [.], ack 5246, win 334, options [
nop,nop,TS val 998415336 ecr 442655403], length 0
09:47:29.364452 IP 192.168.0.2.34894 > server-52-85-105-41.jfk1.r.cloudfront.net.https: Flags [P.], seq 1170:1201, ack 362929, w
in 4316, options [nop,nop,TS val 1345574867 ecr 442656561], length 31
09:47:29.364575 IP 192.168.0.2.34894 > server-52-85-105-41.jfk1.r.cloudfront.net.https: Flags [F.], seq 1201, ack 362929, win 43
16, options [nop,nop,TS val 1345574867 ecr 442656561], length 0
09:47:29.380554 IP server-52-85-105-41.jfk1.r.cloudfront.net.https > 192.168.0.2.34894: Flags [F.], seq 362929, ack 1202, win 12
6, options [nop,nop,TS val 442656836 ecr 1345574867], length 0
09:47:29.380772 IP 192.168.0.2.34894 > server-52-85-105-41.jfk1.r.cloudfront.net.https: Flags [.], ack 362930, win 4316, options
 [nop,nop,TS val 1345574883 ecr 442656836], length 0
^C
3605 packets captured
3713 packets received by filter
108 packets dropped by kernel
[root@/bin/bash: q:: command not found ~]# _
```

(Listening to all network traffic being capture by my private interface, ens19)

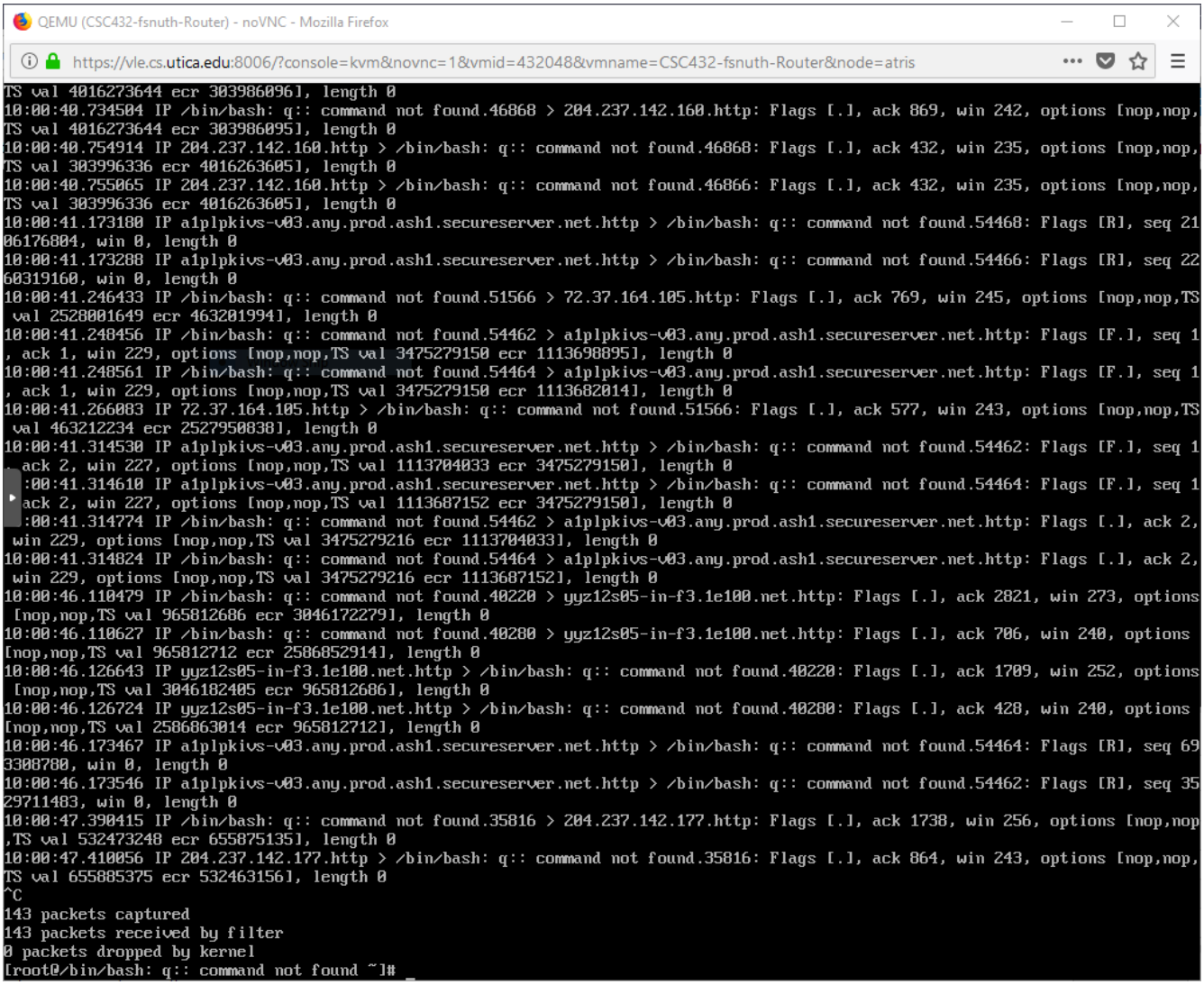


(Listening to all network traffic going to my public interface, ens18)

After that, I then went on to typing the commands that will let me listen in on only SSH traffic and HTTP traffic on my private interface. To listen in on SSH traffic, I typed “tcpdump -i ens19 port 22”. For the HTTP traffic, I typed “tcpdump -i ens19 port 80”. Because I know that SSH traffic goes through port 22 and HTTP traffic goes through port 80, I filtered the traffic by port forwarding.

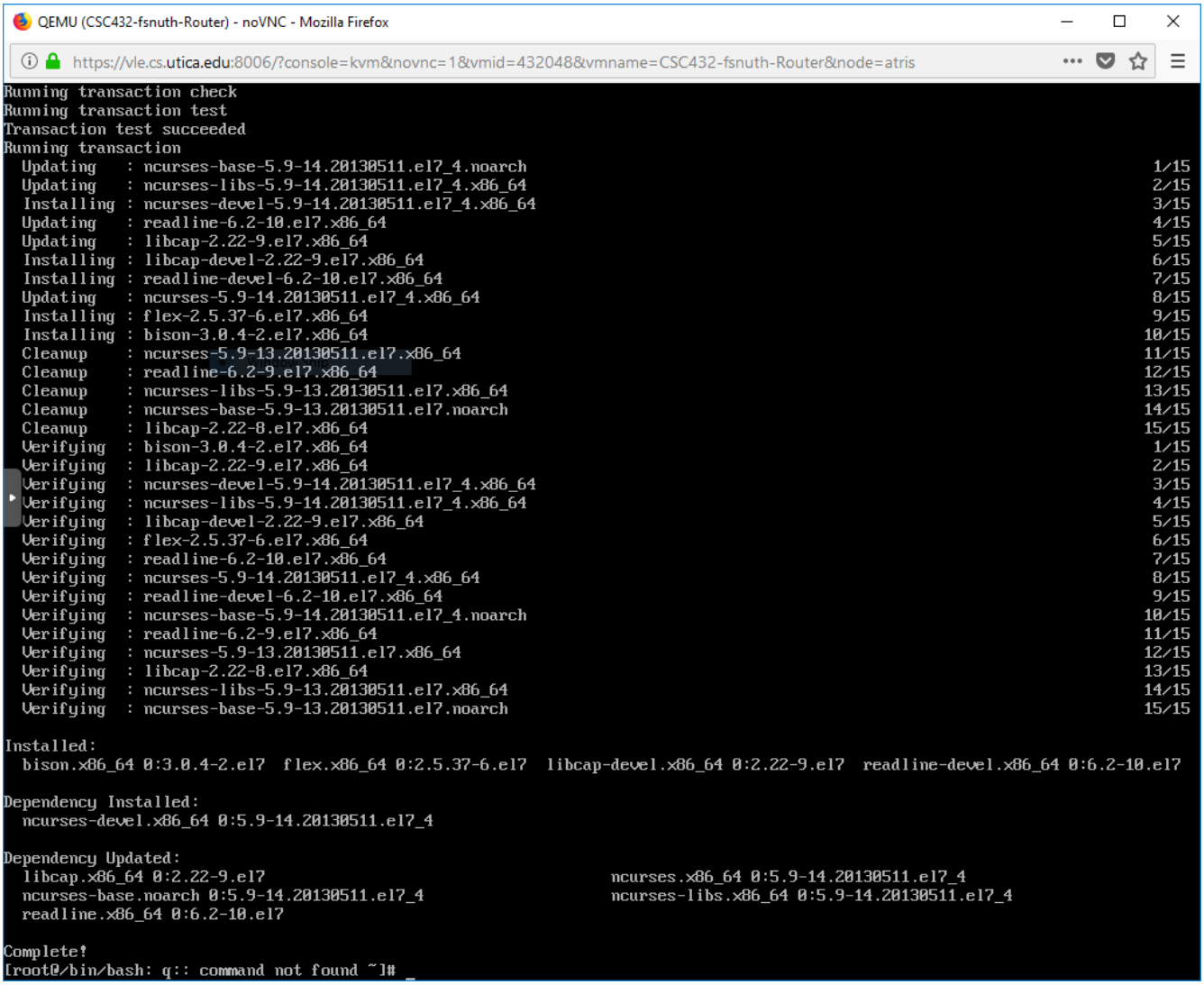


(tcpdump sniffing for SSH traffic.)



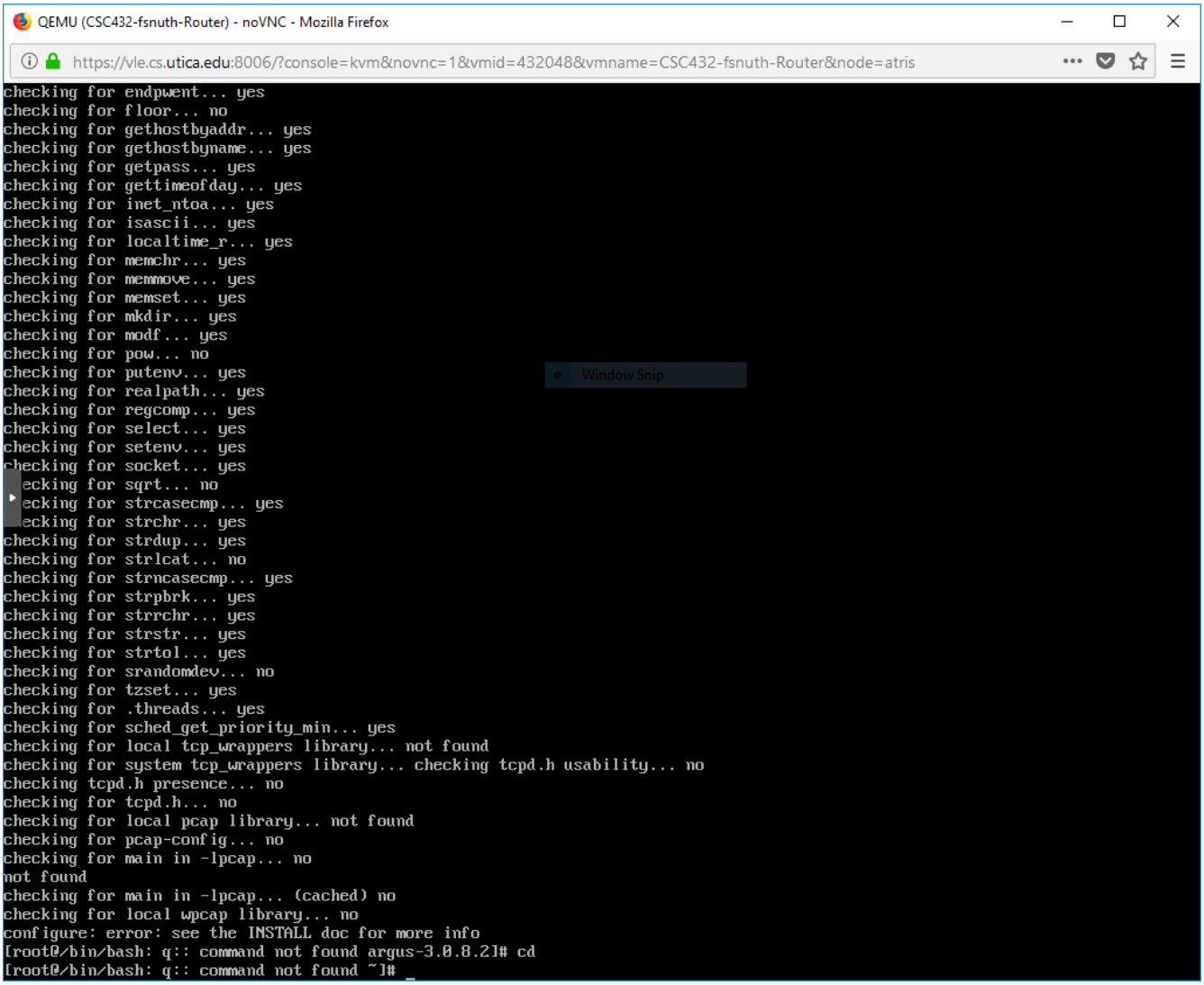
(tcpdump sniffing for HTTP traffic.)

After using **tcpdump**, I moved on to download and use a tool called **argus**. Argus is a network monitoring tool that can provide information about active network flows. Because the **argus** package is not in the **yum** repository, I have to download other components to build it back up. I did this by typing "yum install gcc make bison libpcap libpcap-devel readline-devel flex".

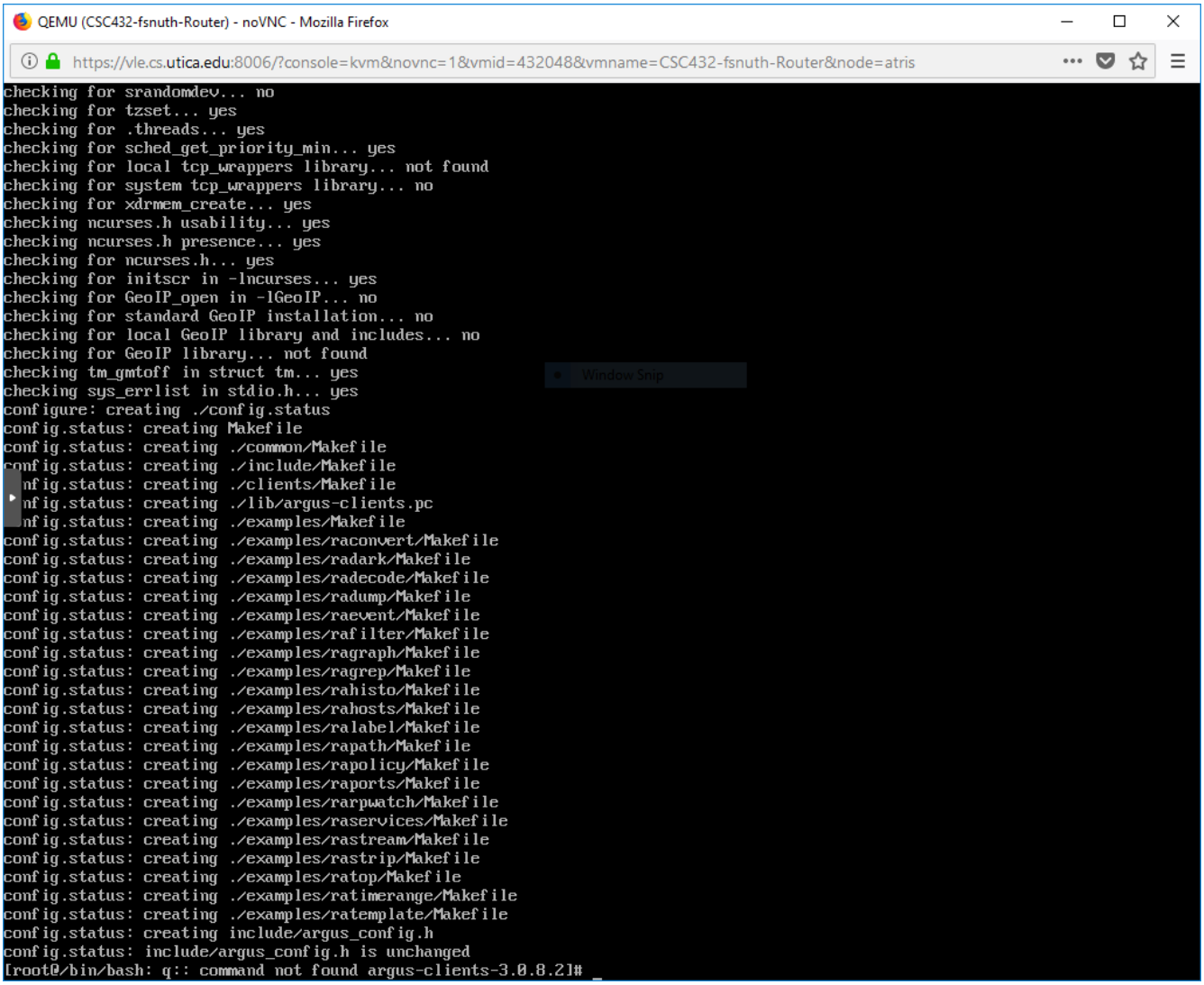


(Installing all the packages needed for Argus.)

After that, I downloaded all the source code needed to run Argus. There are two types of files I neeeded to download. I needed to get the latest version of Argus as well as the client component, then compile them afterward.



(Installing Argus with "cd /usr/src", "yum install wget", "wget http://qosient.com/argus/src/argus-3.0.8.2.tar.gz", and "tar -xaf argus-3.0.8.2.tar.gz"),



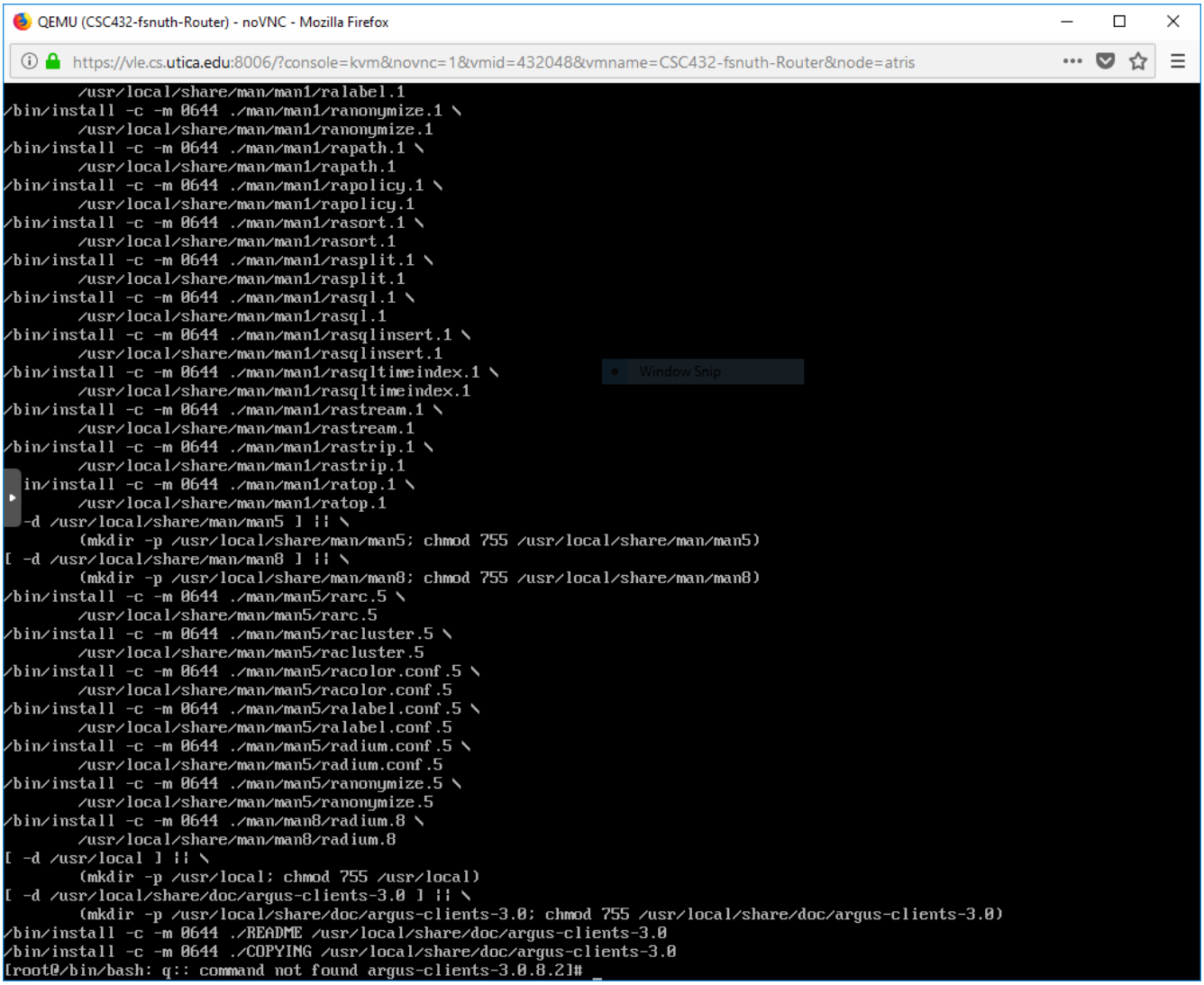
(Installing Argus with "cd /usr/src", "yum install wget", "wget http://qosient.com/argus/src/argus-clients-3.0.8.2.tar.gz", and "tar -xaf argus-clients-3.0.8.2.tar.gz")

QEMU (CSC432-fsnuth-Router) - noVNC - Mozilla Firefox

https://vle.cs.utica.edu:8006/?console=kvm&novnc=1&vmid=432048&vmname=CSC432-fsnuth-Router&node=atris

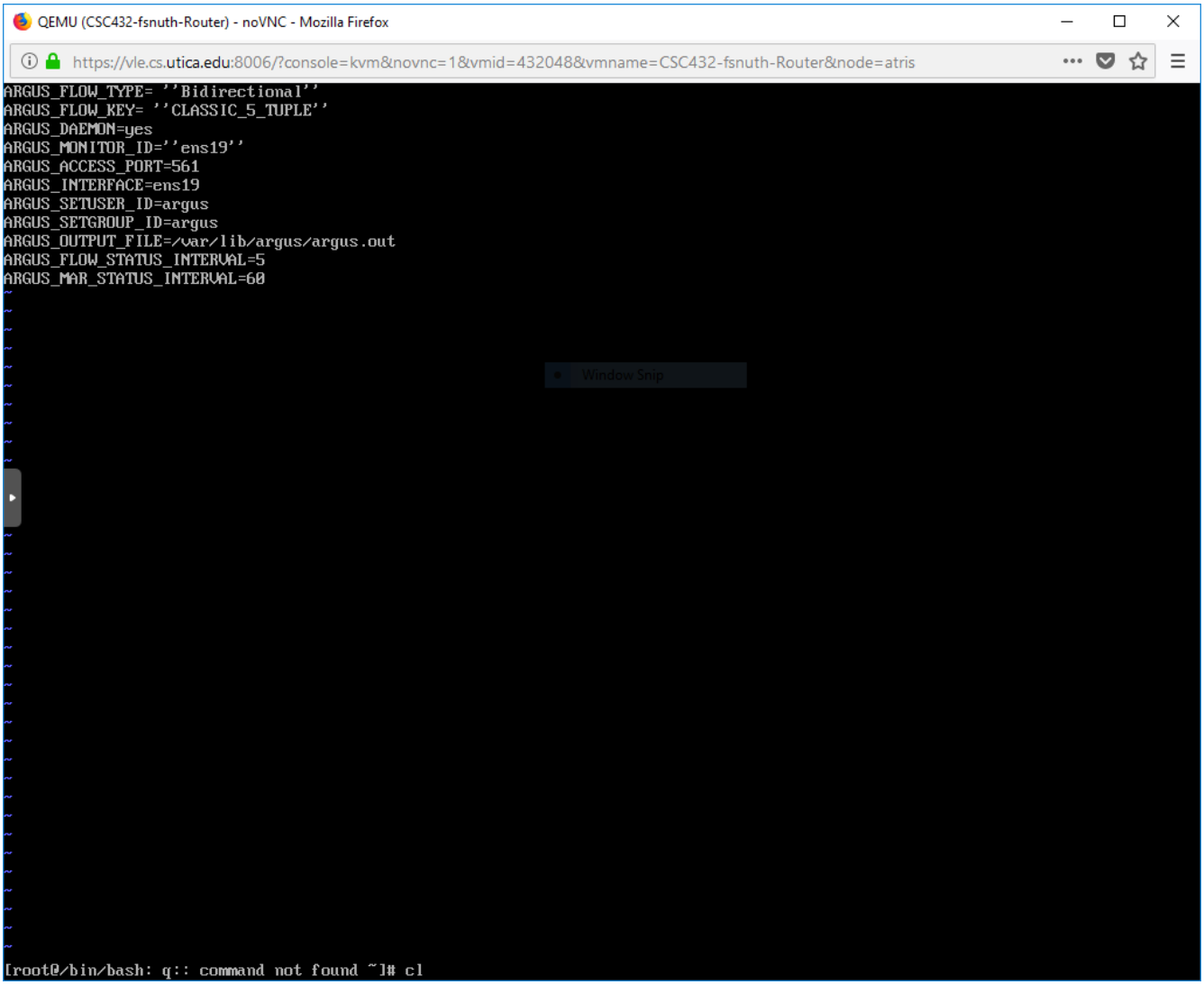
```
make[1]: Entering directory `/usr/src/argus-3.0.8.2'
./config/mkinstalldirs  \
        /usr/local/share/doc/argus-3.0 /usr/local/argus/archive
mkdir /usr/local/share/doc/argus-3.0
mkdir /usr/local/argus/archive
make[1]: Leaving directory `/usr/src/argus-3.0.8.2'
[ -d /usr/local ] || \
        (mkdir -p /usr/local; chmod 755 /usr/local)
[ -d /usr/local/sbin ] || \
        (mkdir -p /usr/local/sbin; chmod 755 /usr/local/sbin)
[ -d /usr/local/bin ] || \
        (mkdir -p /usr/local/bin; chmod 755 /usr/local/bin)
[ -d /usr/local/argus ] || \
        (mkdir -p /usr/local/argus; chmod 755 /usr/local/argus)
[ -d /usr/local/argus/archive ] || \
        (mkdir -p /usr/local/argus/archive; chmod 755 /usr/local/argus/archive)
### Make install in /usr/src/argus-3.0.8.2/common
make[1]: Entering directory `/usr/src/argus-3.0.8.2/common'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/usr/src/argus-3.0.8.2/common'
### Done with /usr/src/argus-3.0.8.2/common
### Make install in /usr/src/argus-3.0.8.2/argus
make[1]: Entering directory `/usr/src/argus-3.0.8.2/argus'
[ -d /usr/local/sbin ] || \
        (mkdir -p /usr/local/sbin; chmod 755 /usr/local/sbin)
/bin/install -c ./../bin/argus /usr/local/sbin/argus
make[1]: Leaving directory `/usr/src/argus-3.0.8.2/argus'
### Done with /usr/src/argus-3.0.8.2/argus
### Make install in /usr/src/argus-3.0.8.2/events
make[1]: Entering directory `/usr/src/argus-3.0.8.2/events'
[ -d /usr/local/sbin ] || \
        (mkdir -p /usr/local/sbin; chmod 755 /usr/local/sbin)
/bin/install -c ./../bin/argus-extip  /usr/local/bin/argus-extip
/bin/install -c ./../bin/argus-lsof   /usr/local/bin/argus-lsof
/bin/install -c ./../bin/argus-snmp   /usr/local/bin/argus-snmp
/bin/install -c ./../bin/argus-vmstat /usr/local/bin/argus-vmstat
make[1]: Leaving directory `/usr/src/argus-3.0.8.2/events'
### Done with /usr/src/argus-3.0.8.2/events
/bin/install -c -m 0755 ./bin/argusbug /usr/local/bin/argusbug
[ -d /usr/local/share/man ] || \
        (mkdir -p /usr/local/share/man; chmod 755 /usr/local/share/man)
[ -d /usr/local/share/man/man5 ] || \
        (mkdir -p /usr/local/share/man/man5; chmod 755 /usr/local/share/man/man5)
[ -d /usr/local/share/man/man8 ] || \
        (mkdir -p /usr/local/share/man/man8; chmod 755 /usr/local/share/man/man8)
/bin/install -c -m 0644 ./man/man5/argus.conf.5 /usr/local/share/man/man5/argus.conf.5
/bin/install -c -m 0644 ./man/man8/argus.8 /usr/local/share/man/man8/argus.8
[root@/bin/bash: q:: command not found argus-3.0.8.2]# _
```

(Compiling Argus with "cd /usr/src/argus-3.0.8.2", "./configure", "make && make install".)



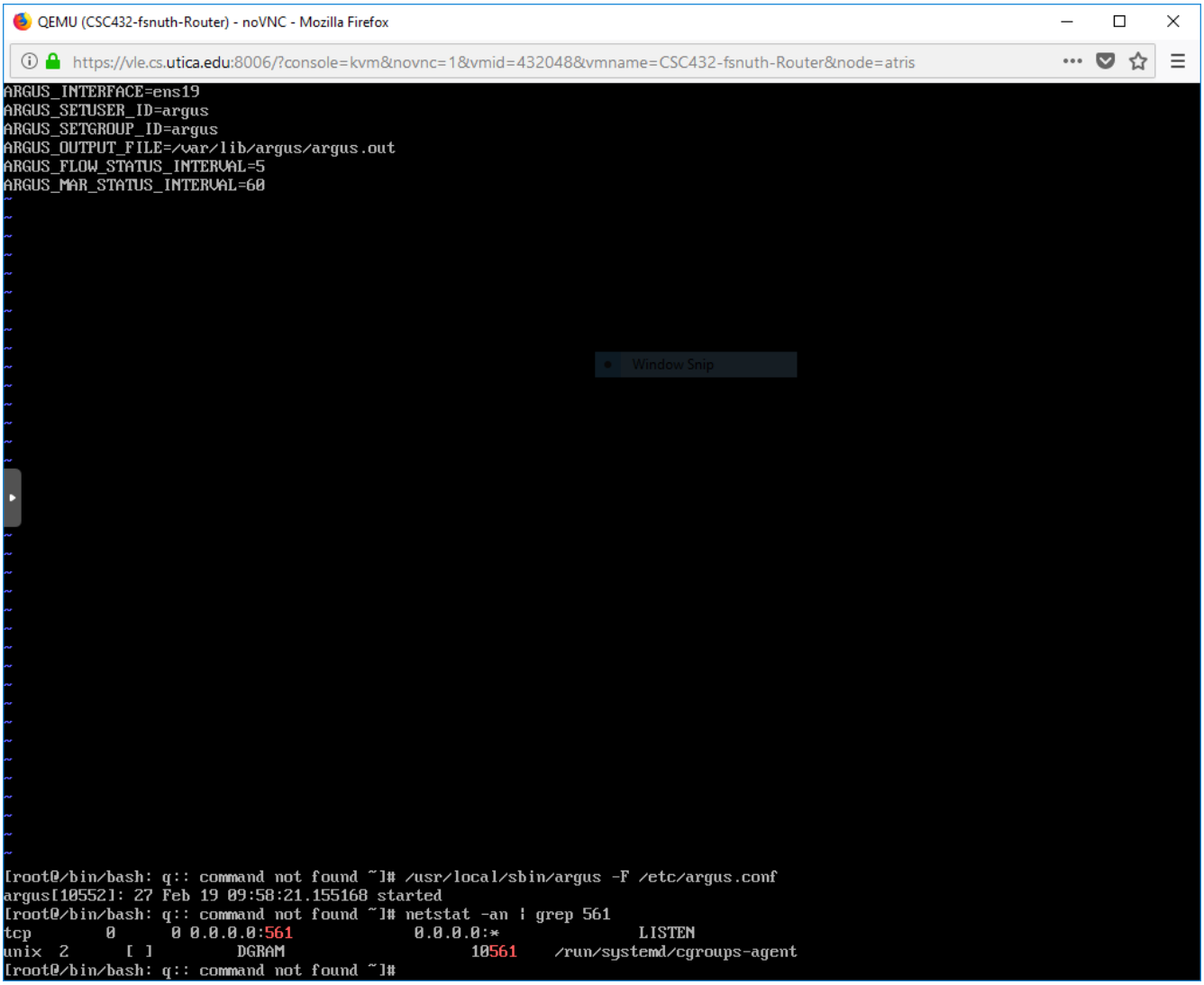
(Compiling Argus with "cd /usr/src/argus-3.0.8.2", "./configure", "make && make install".)

I also worked on creating an **argus.conf** file in the **/etc/argus.conf**. This will alllow us to actually run Argus properly.



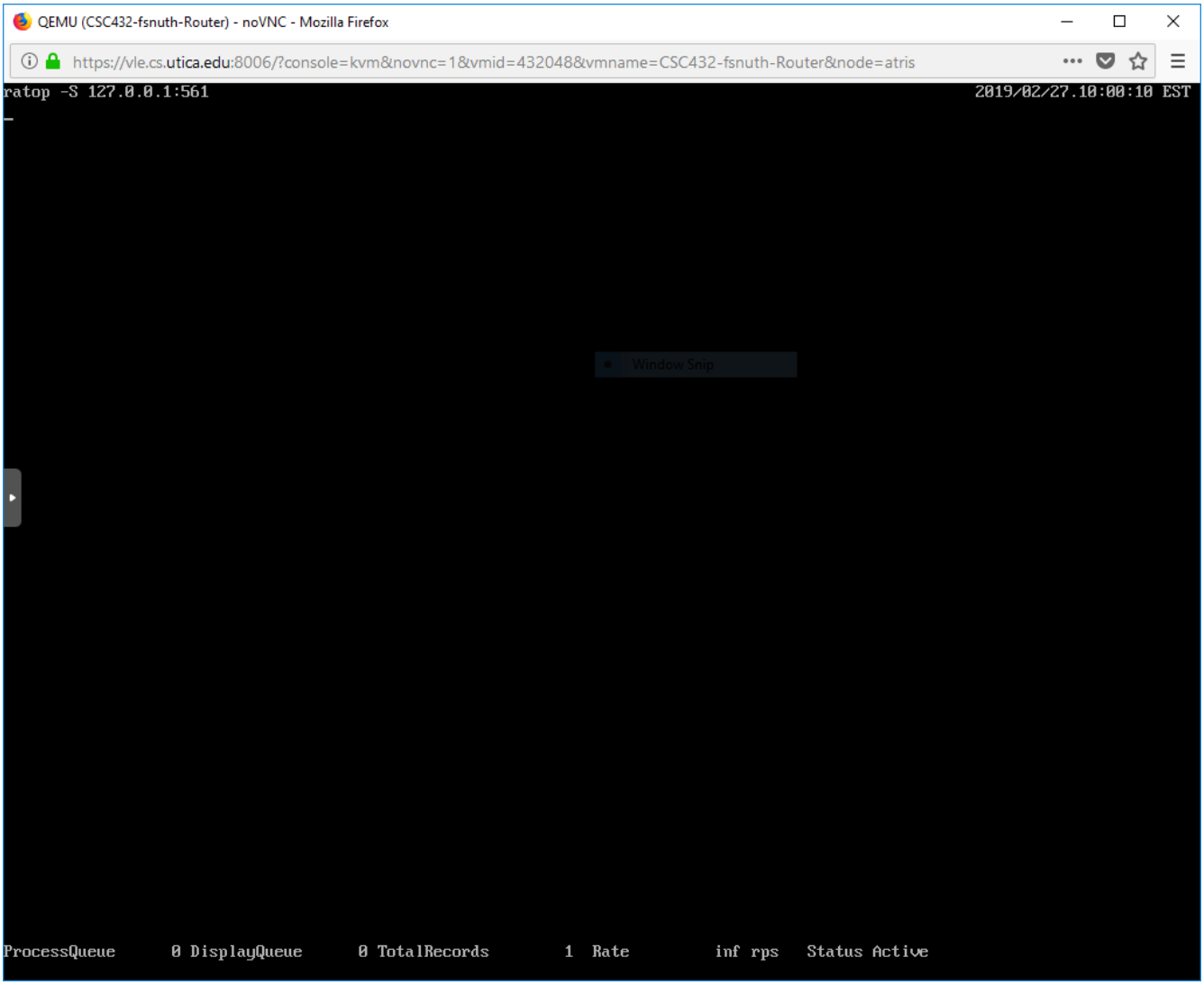
(Creating and customing the <u>argus.conf</u> file.)

We also created an argus user and group with “echo “argus:x:6000:6000:Argus:/home/argus:/sbin/nologin” >> /etc/passwd” and “argus:x:6000 >> /etc/group”. To set the correct permissions with the output directory, we use “mkdir /var/lib/argus”, “touch /var/lib/argus/argus.out”, and “chown -R argus:argus /var/lib/argus”. Then we run “/usr/local/sbin/argus -F /etc/argus.conf”.

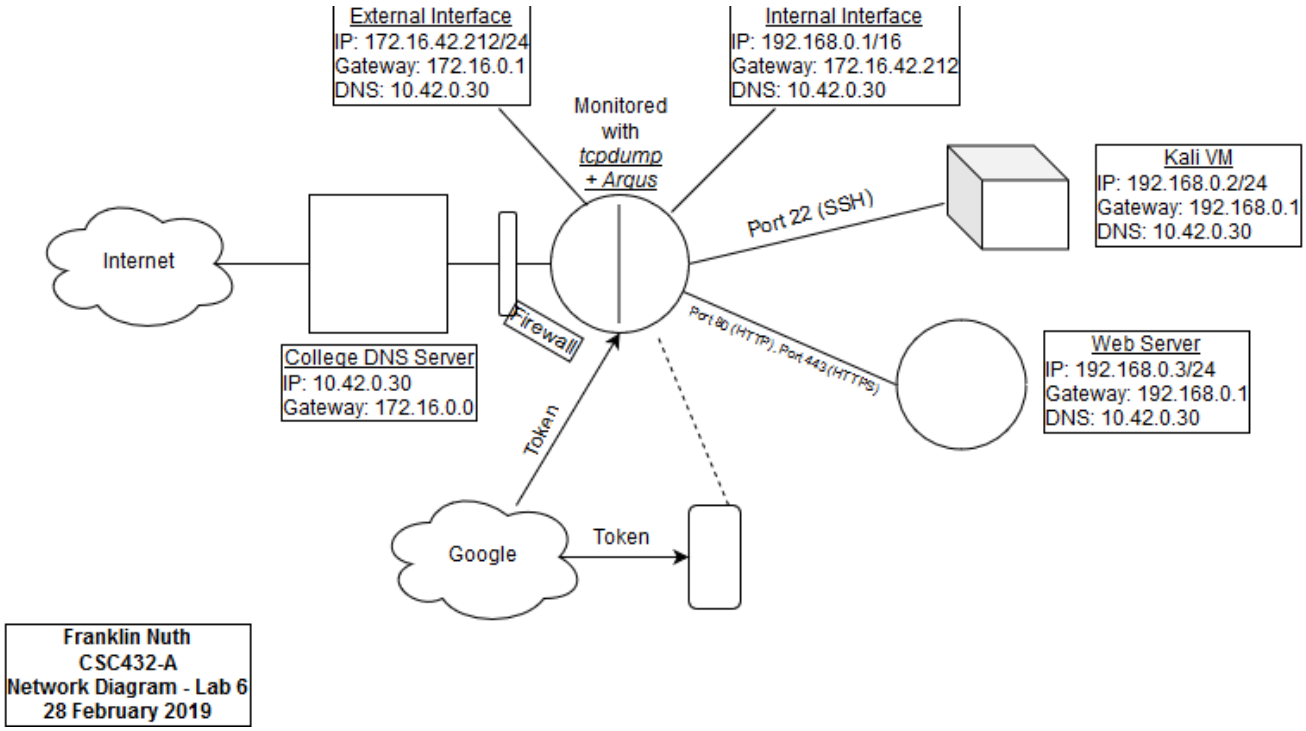


(Argus running its network flow monitoring services.)

<u>ratop</u> is one of the data cliens that come with Argus. **<u>Ratop</u>** is useful for displaying network flow data for easire use. One way to use **<u>ratop</u>** is “ratop -S 127.0.0.1:561”. Port deals with TCP and UDP information, so this means that this listens in on IP 127.0.0.1 for any protocol from the Transport Layer.



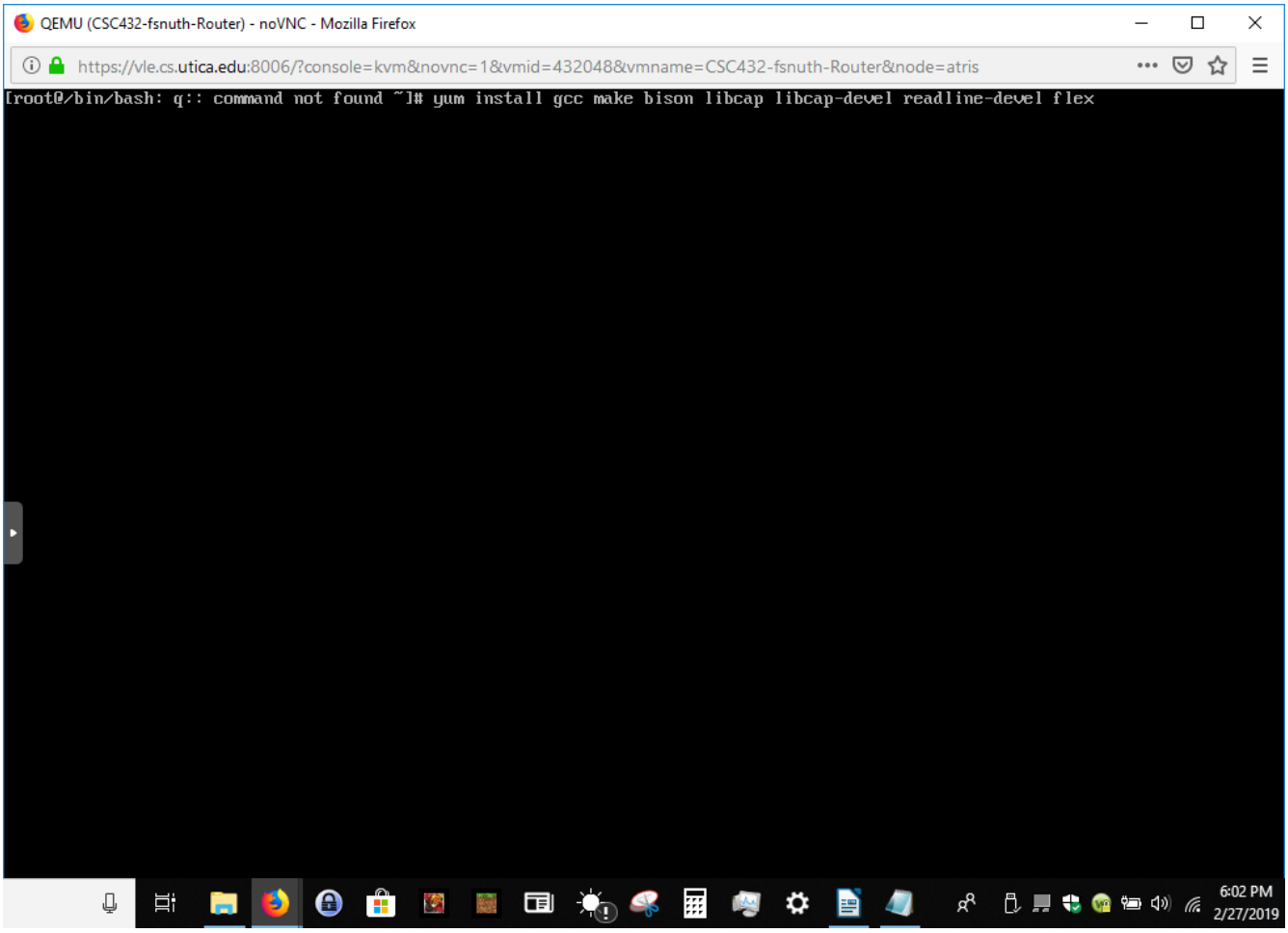
(Using ratop, which is part of the Argus software.)



(The sixth revision of my network diagram after installing tcpdump and Argus)

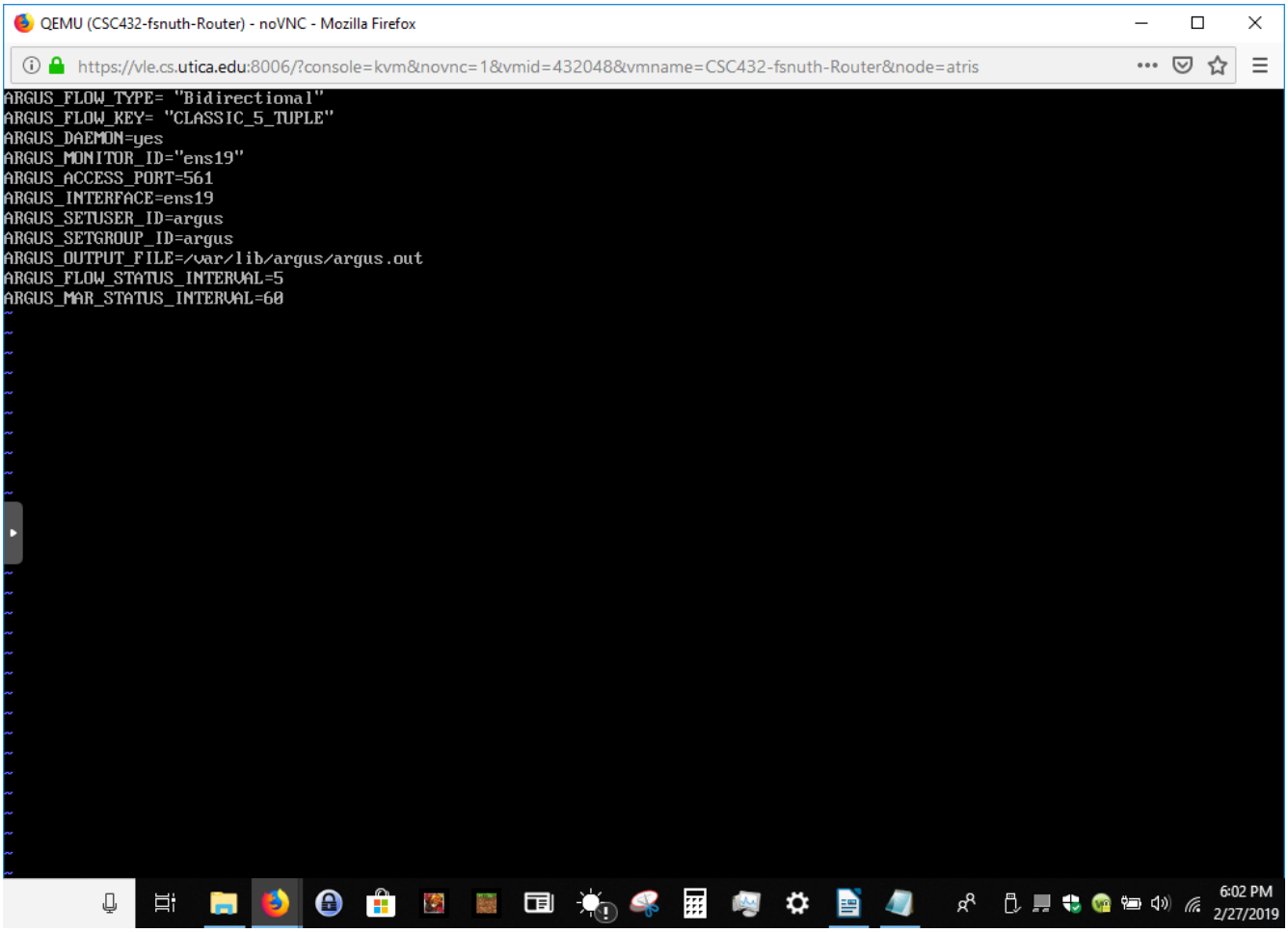
Issues & Resolutions

I encountered two minor issues in this lab. The first issue that I ran into is that I mistyped one of the packages. In the syntax "yum install gcc make bison libpcap libcap-devel readline-devel flex", I accidently typed "lipcap" instead of "libpcap". This small mistake set me back a couple of days, until my professor and classmates helped pointed it out. I will take measure to start reading the lab slowly to avoid this issue.



(I will start to read more slowly from this point on.)

Another issue I've dealt with is incorrect information in my argus.conf file. While trying to set correct permissions in the output directory, it does not work for me because I had an ID problem. The ID lines in my configuration file was actually two single quotatoin marks, not the double quotations. My classmate pointed this out to me and we immediately changed the single quotations to doubles. I had no problem making users and groups after that small fix.



(Removed the single quotations and inserted the double quotations.)

Conclusion

In this lab, I have shown how to install **tcpdump** and use it for monitoring SSH or HTTP traffic. I also shown how to install Argus as well as a few of its data clients for network flow monitoring. A network administarator can use both of these tools as ways to manually monitor all IP and network traffic on the network. It is intriguing to watch how packets and IP address work with each other through programs like Wireshark and tcpdump. Although it is more technical than standing in front of a door, the techniques and concepts I picked up from this lab are just as good for my network.

References

Bullard, Carter. 2012 August 3. *RAFILTERADDR(1).* Retrieved from: https://qosient.com/argus/man/man1/rafilteraddr.1.pdf

QoSient.com. 2019 January 14. *Argus Archive*. Retrieved from: https://qosient.com/argus/

Shrestha, Narad. 2012 August 20. *12 Tcpdump Commands – A Network Sniffer Tool*. Retrireved from: https://www.tecmint.com/12-tcpdump-commands-a-network-sniffer-tool/

Wireshark.org. 2019 February 28. *Chapter 1 Introduction.* Retrieved from: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html