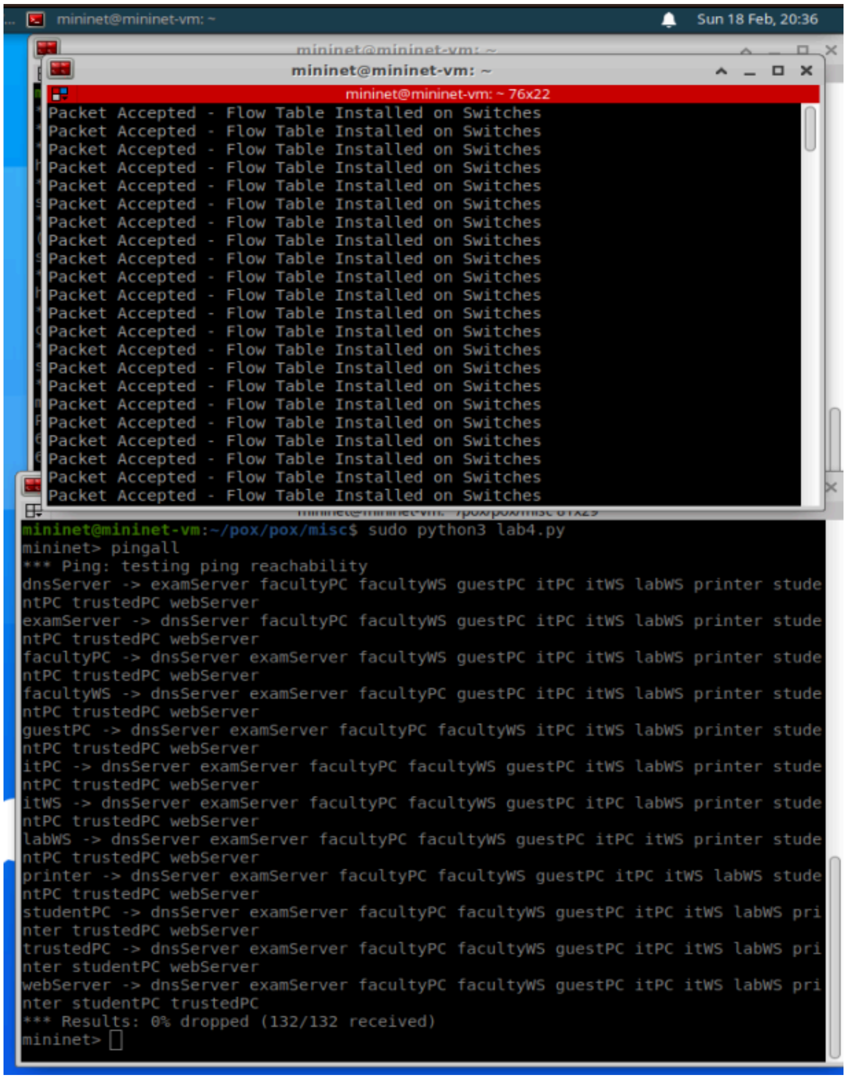
1. Pseudocode to implement firewall rules:

```
#Common access group ip list
access_list = [facultyWS, facultyPC, labWS, studentPC, iTWs, iTPC, trustedPC, guestPC]


do_firewall(src, dst, protocol):
  # Rule 1 - Allow all ARP traffic
  if protocol == "arp":
    accept()

  # Rule 2 - Allow ICMP traffic, but not to dnsServer
  elif protocol == "icmp":
    accept()

  # Rules 3 and 4 - Allow TCP traffic to and from webServer from access list
  elif protocol == "tcp":
    if (src in access_ws_list and dst == webServer) or (src == webServer and dst in access_ws_list):
      accept()

  # Rule 5 and 6 - Allow TCP traffic to and from examServer from facultyWS and facultyPC
  elif (src in [facultyWS, facultyPC] and dst == examServer) or (src == examServer and dst in [facultyWS, facultyPC]):
    accept()

  # Rule 7 - Allow TCP and UDP traffic between facultyWS, facultyPC, labWS, and studentPC
  elif ((src in [facultyWS, facultyPC, labWS, studentPC] and dst in [iTWs, iTPC]) or (src in [iTWs, iTPC] and dst in [facultyWS, facultyPC, labWS, studentPC])) and (protocol == "tcp" or protocol == "udp"):
    accept()

  # Rule 8 - Allow TCP and UDP traffic from facultyWS and facultyPC to iTWs and iTPC
  elif src in [facultyWS, facultyPC] and dst in [iTWs, iTPC] and(protocol == "tcp" or protocol == "udp"):
    accept()

  # Rule 9 - Allow UDP traffic to dnsServer from everyone
  elif protocol == "udp" and dst == dnsServer:
    accept()
```

```
# Rule 10 - Allow UDP traffic from dnsServer to everyone
elif protocol == "udp" and src == dnsServer:
    accept()

# Default rule - Drop any other traffic
else:
    drop()
```

2. a. The output of pingall was completed successfully with all the packets transmitted which was as defined in the Firewall rules. In the Firewall Rule #2, it is set so that ICMP packets from any source can be transmitted through the network.



```
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
```

```
mininet@mininet-vm:~/pox/pox/misc$ sudo python3 lab4.py
mininet> pingall
*** Ping: testing ping reachability
dnsServer -> examServer facultyPC facultyWS guestPC itPC itWS labWS printer stude
ntPC trustedPC webServer
examServer -> dnsServer facultyPC facultyWS guestPC itPC itWS labWS printer stude
ntPC trustedPC webServer
facultyPC -> dnsServer examServer facultyWS guestPC itPC itWS labWS printer stude
ntPC trustedPC webServer
facultyWS -> dnsServer examServer facultyPC guestPC itPC itWS labWS printer stude
ntPC trustedPC webServer
guestPC -> dnsServer examServer facultyPC facultyWS itPC itWS labWS printer stude
ntPC trustedPC webServer
itPC -> dnsServer examServer facultyPC facultyWS guestPC itWS labWS printer stude
ntPC trustedPC webServer
itWS -> dnsServer examServer facultyPC facultyWS guestPC itPC labWS printer stude
ntPC trustedPC webServer
labWS -> dnsServer examServer facultyPC facultyWS guestPC itPC itWS printer stude
ntPC trustedPC webServer
printer -> dnsServer examServer facultyPC facultyWS guestPC itPC itWS labWS stude
ntPC trustedPC webServer
studentPC -> dnsServer examServer facultyPC facultyWS guestPC itPC itWS labWS pri
nter trustedPC webServer
trustedPC -> dnsServer examServer facultyPC facultyWS guestPC itPC itWS labWS pri
nter studentPC webServer
webServer -> dnsServer examServer facultyPC facultyWS guestPC itPC itWS labWS pri
nter studentPC trustedPC
*** Results: 0% dropped (132/132 received)
mininet> 
```

#	Link	iperf command	Pass or Fail?
1	*IT Workstation - Lab Workstation*	`iperf itWS labWS`	pass
2	*Faculty PC - DNS Server*	`iperf facultyPC dnsServer`	fail (if none-responsive use “Ctrl + C”)
3	*Student PC - Exam Server*	iperf studentPC examServer	fail
4	*Printer - IT Workstation*	iperf printer itWS	fail
5	*Faculty Workstation - Web Server*	iperf facultyWS webServer	pass

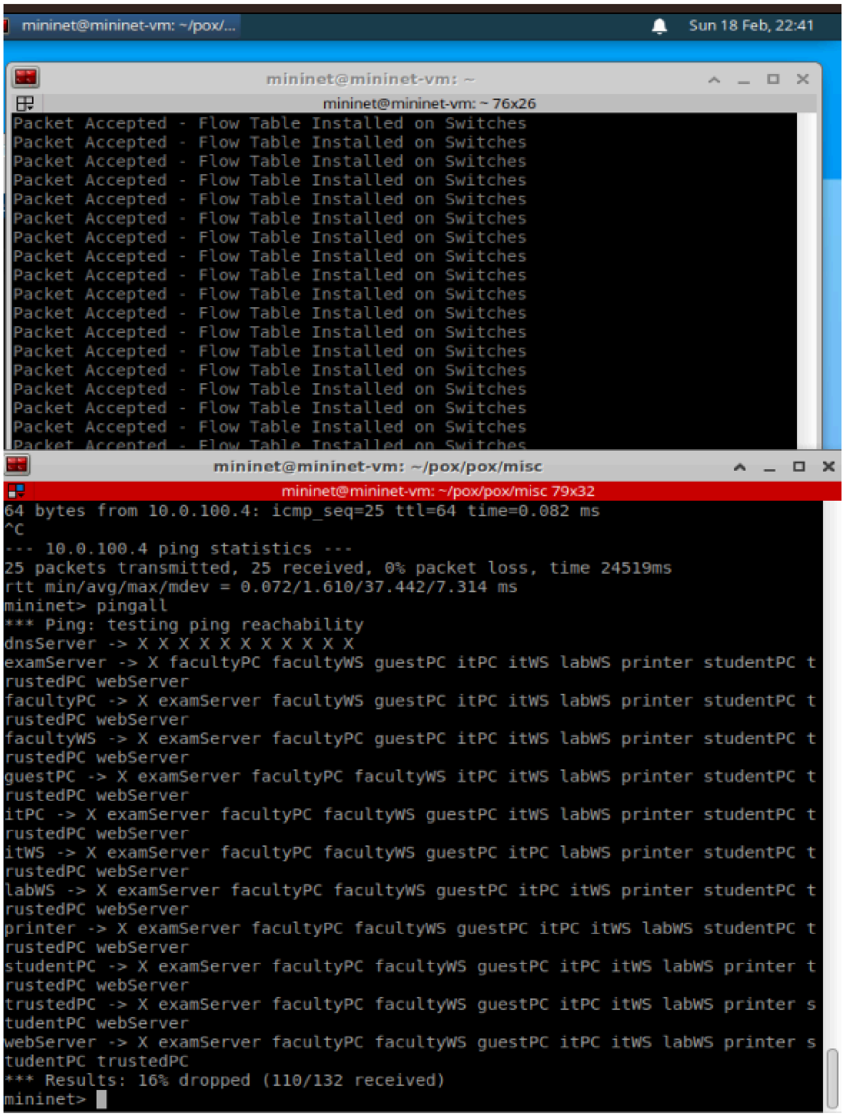
b.

6	*Lab Workstation - Student PC*	iperf labWS studentPC	fail

c.

Protocol	Link	Traceroute Command	Pass or Fail?
ICMP	*Student PC - Exam Server*	studentPC traceroute -I examServer	pass
UDP	*IT Workstation - Lab Workstation*	itWS traceroute -U labWS	fail
UDP	*IT PC - Web Server*	itPC traceroute -U webServer	fail
UDP	*DNS Server - Faculty PC*	dnsServer traceroute -U facultyPC	pass

3. a. The command used to test the Firewall is pingall. The rules are working because dnsServer cannot be pinged and therefore is not receiving ICMP requests and is safe from the DoS attack.

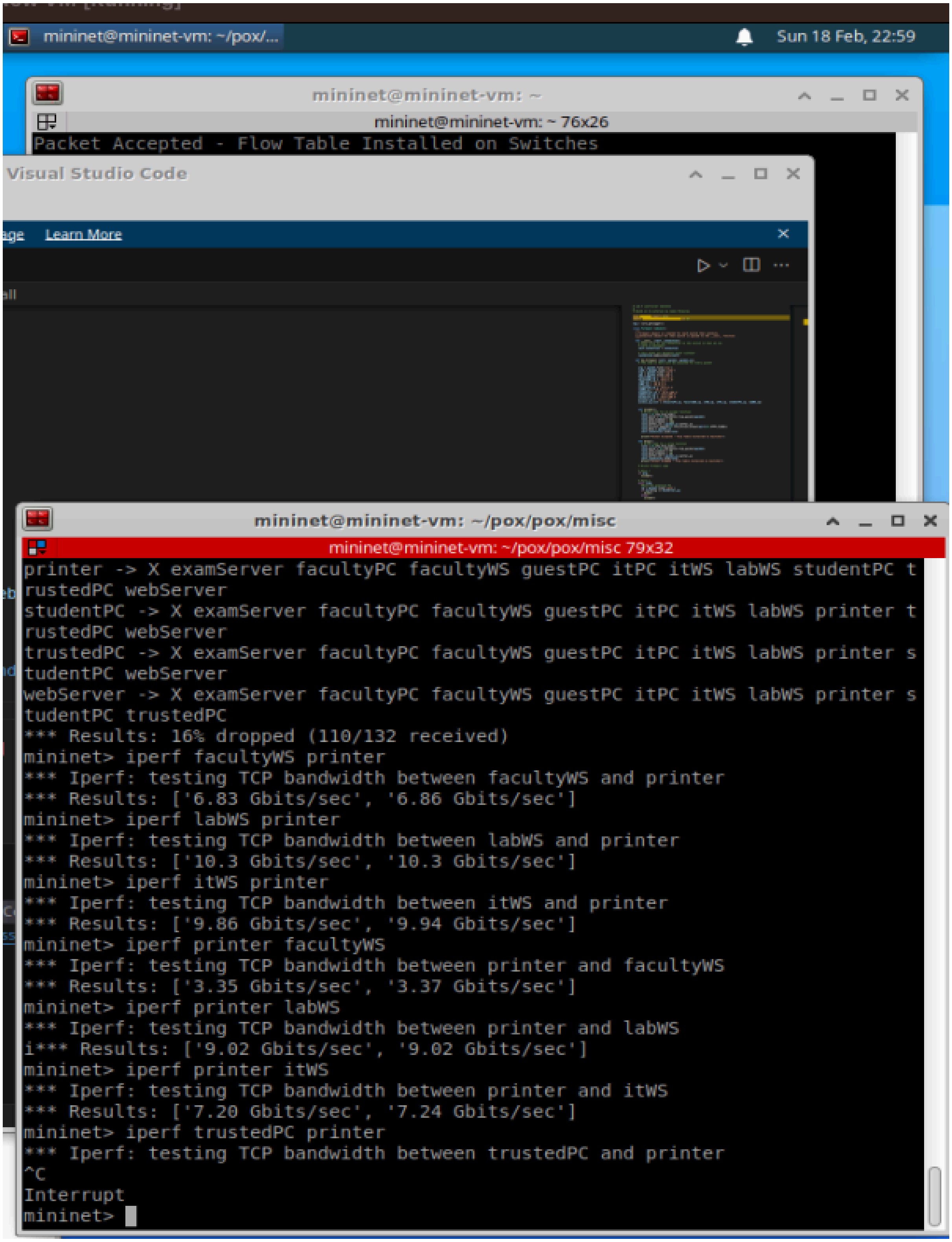


```
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
Packet Accepted - Flow Table Installed on Switches
```

mininet@mininet-vm: ~/pox/pox/misc
mininet@mininet-vm: ~/pox/pox/misc 79x32

```
64 bytes from 10.0.100.4: icmp_seq=25 ttl=64 time=0.082 ms
^C
--- 10.0.100.4 ping statistics ---
25 packets transmitted, 25 received, 0% packet loss, time 24519ms
rtt min/avg/max/mdev = 0.072/1.610/37.442/7.314 ms
mininet> pingall
*** Ping: testing ping reachability
dnsServer -> X X X X X X X X X X X
examServer -> X facultyPC facultyWS guestPC itPC itWS labWS printer studentPC t
rustedPC webServer
facultyPC -> X examServer facultyWS guestPC itPC itWS labWS printer studentPC t
rustedPC webServer
facultyWS -> X examServer facultyPC guestPC itPC itWS labWS printer studentPC t
rustedPC webServer
guestPC -> X examServer facultyPC facultyWS itPC itWS labWS printer studentPC t
rustedPC webServer
itPC -> X examServer facultyPC facultyWS guestPC itWS labWS printer studentPC t
rustedPC webServer
itWS -> X examServer facultyPC facultyWS guestPC itPC labWS printer studentPC t
rustedPC webServer
labWS -> X examServer facultyPC facultyWS guestPC itPC itWS printer studentPC t
rustedPC webServer
printer -> X examServer facultyPC facultyWS guestPC itPC itWS labWS studentPC t
rustedPC webServer
studentPC -> X examServer facultyPC facultyWS guestPC itPC itWS labWS printer t
rustedPC webServer
trustedPC -> X examServer facultyPC facultyWS guestPC itPC itWS labWS printer s
tudentPC webServer
webServer -> X examServer facultyPC facultyWS guestPC itPC itWS labWS printer s
tudentPC trustedPC
*** Results: 16% dropped (110/132 received)
mininet> 
```

b. The commands used to test the Firewall are iperf facultyWS printer, iperf labWS printer, iperf itWS printer, iperf printer facultyWS, iperf printer labWS, iperf printer itWS, iperf trustedPC printer (to check for iperf failure), etc. The rules support the answer because all of the iperf commands from the University workstations to the printer and vice-versa worked, but the iperf command for any other device outside the University failed.

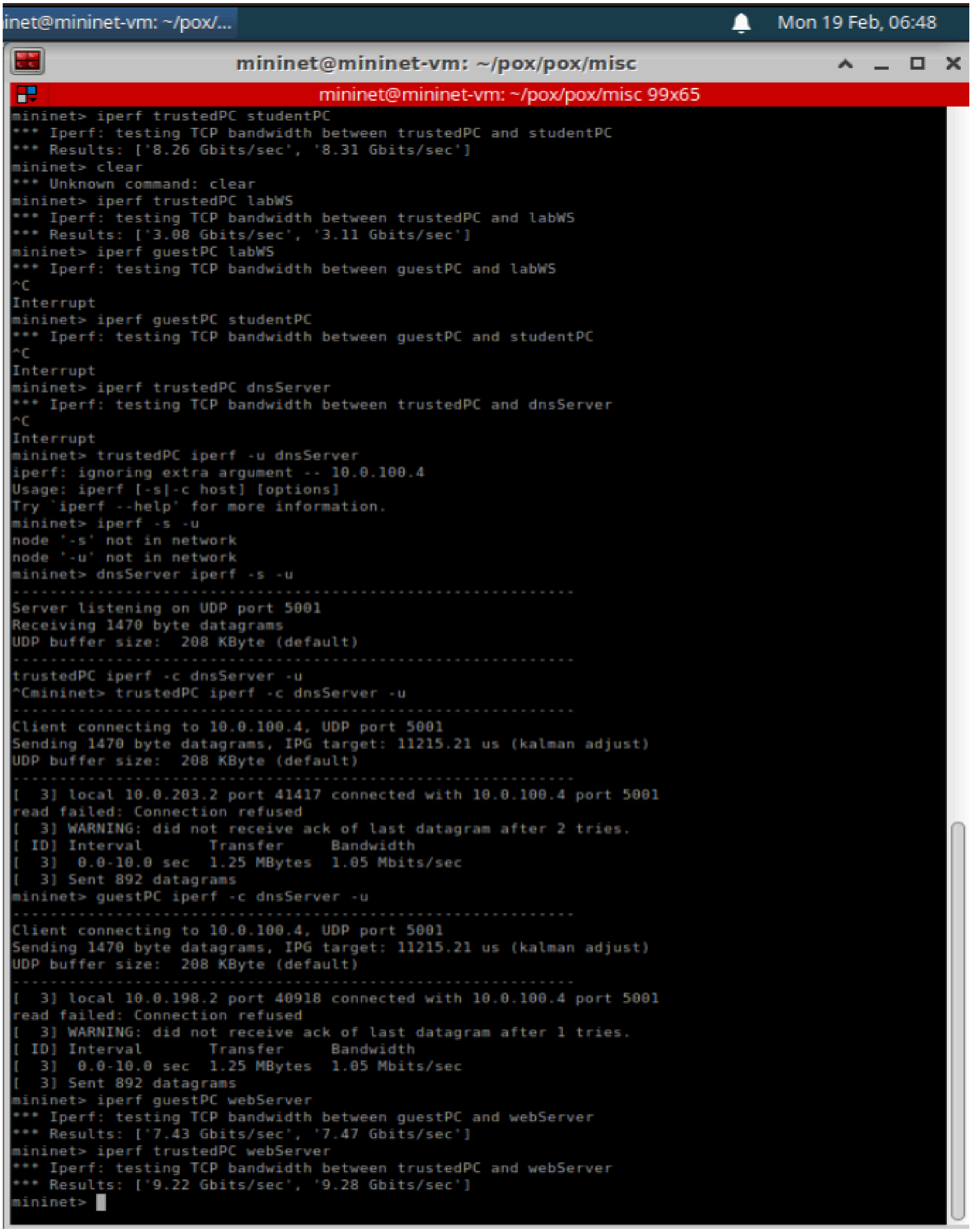

mininet@mininet-vm: ~/pox/...
Sun 18 Feb, 22:59
mininet@mininet-vm: ~
mininet@mininet-vm: ~ 76x26
Visual Studio Code
Learn More
mininet@mininet-vm: ~/pox/pox/misc
mininet@mininet-vm: ~/pox/pox/misc 79x32

```
Packet Accepted - Flow Table Installed on Switches
```

```
printer -> X examServer facultyPC facultyWS guestPC itPC itWS labWS studentPC t
rustedPC webServer
studentPC -> X examServer facultyPC facultyWS guestPC itPC itWS labWS printer t
rustedPC webServer
trustedPC -> X examServer facultyPC facultyWS guestPC itPC itWS labWS printer s
tudentPC webServer
webServer -> X examServer facultyPC facultyWS guestPC itPC itWS labWS printer s
tudentPC trustedPC
*** Results: 16% dropped (110/132 received)
mininet> iperf facultyWS printer
*** Iperf: testing TCP bandwidth between facultyWS and printer
*** Results: ['6.83 Gbits/sec', '6.86 Gbits/sec']
mininet> iperf labWS printer
*** Iperf: testing TCP bandwidth between labWS and printer
*** Results: ['10.3 Gbits/sec', '10.3 Gbits/sec']
mininet> iperf itWS printer
*** Iperf: testing TCP bandwidth between itWS and printer
*** Results: ['9.86 Gbits/sec', '9.94 Gbits/sec']
mininet> iperf printer facultyWS
*** Iperf: testing TCP bandwidth between printer and facultyWS
*** Results: ['3.35 Gbits/sec', '3.37 Gbits/sec']
mininet> iperf printer labWS
*** Iperf: testing TCP bandwidth between printer and labWS
i*** Results: ['9.02 Gbits/sec', '9.02 Gbits/sec']
mininet> iperf printer itWS
*** Iperf: testing TCP bandwidth between printer and itWS
*** Results: ['7.20 Gbits/sec', '7.24 Gbits/sec']
mininet> iperf trustedPC printer
*** Iperf: testing TCP bandwidth between trustedPC and printer
^C
Interrupt
mininet>
```

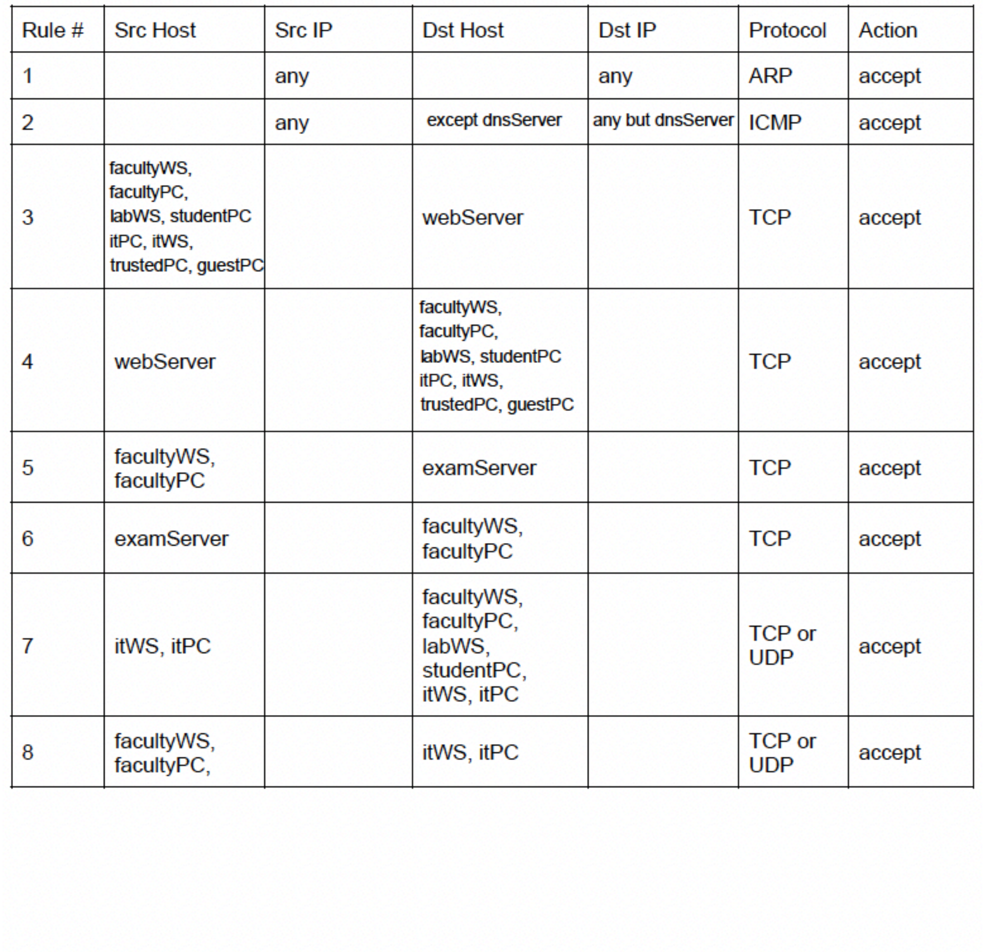
c. The commands used to test the Firewall are iperf trustedPC labWS, iperf trustedPC studentPC, iperf guestPC labWS, iperf guestPC studentPC, iperf -c dnsServer -u and trustedPC iperf -c dnsServer -u, guestPC iperf -c dnsServer -u (to test iperf with UDP can also use iperfudp src dst), iperf trustedPC webServer, iperf guestPC webServer. (*The same commands with the src and dst reversed would also be good to check but since my implementation is the same for both ways and there are already plenty of commands to check, I am only testing one-way.)

These rules are working by testing the access of guestPC and trustedPC to web browsing, dnsServers, and student LAN. The rules work as intended since all are true except for guestPC to student LAN devices because it has limited access compared to trustedPC's broader access.

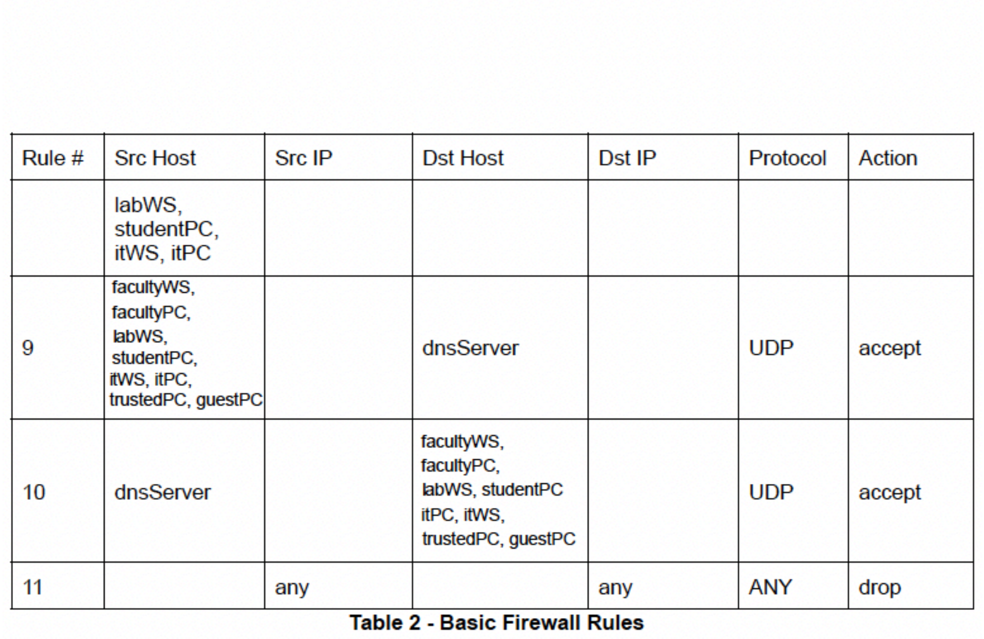

inet@mininet-vm: ~/pox/...
Mon 19 Feb, 06:48
mininet@mininet-vm: ~/pox/pox/misc
mininet@mininet-vm: ~/pox/pox/misc 99x65

```
mininet> iperf trustedPC studentPC
*** Iperf: testing TCP bandwidth between trustedPC and studentPC
*** Results: ['8.26 Gbits/sec', '8.31 Gbits/sec']
mininet> clear
*** Unknown command: clear
mininet> iperf trustedPC labWS
*** Iperf: testing TCP bandwidth between trustedPC and labWS
*** Results: ['3.08 Gbits/sec', '3.11 Gbits/sec']
mininet> iperf guestPC labWS
*** Iperf: testing TCP bandwidth between guestPC and labWS
^C
Interrupt
mininet> iperf guestPC studentPC
*** Iperf: testing TCP bandwidth between guestPC and studentPC
^C
Interrupt
mininet> iperf trustedPC dnsServer
*** Iperf: testing TCP bandwidth between trustedPC and dnsServer
^C
Interrupt
mininet> trustedPC iperf -u dnsServer
iperf: ignoring extra argument -- 10.0.100.4
Usage: iperf [-s|-c host] [options]
Try `iperf --help' for more information.
mininet> iperf -s -u
node '-s' not in network
node '-u' not in network
mininet> dnsServer iperf -s -u
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
trustedPC iperf -c dnsServer -u
^Cmininet> trustedPC iperf -c dnsServer -u
------------------------------------------------------------
Client connecting to 10.0.100.4, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.203.2 port 41417 connected with 10.0.100.4 port 5001
read failed: Connection refused
[  3] WARNING: did not receive ack of last datagram after 2 tries.
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Sent 892 datagrams
mininet> guestPC iperf -c dnsServer -u
------------------------------------------------------------
Client connecting to 10.0.100.4, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.198.2 port 40918 connected with 10.0.100.4 port 5001
read failed: Connection refused
[  3] WARNING: did not receive ack of last datagram after 1 tries.
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Sent 892 datagrams
mininet> iperf guestPC webServer
*** Iperf: testing TCP bandwidth between guestPC and webServer
*** Results: ['7.43 Gbits/sec', '7.47 Gbits/sec']
mininet> iperf trustedPC webServer
*** Iperf: testing TCP bandwidth between trustedPC and webServer
*** Results: ['9.22 Gbits/sec', '9.28 Gbits/sec']
mininet> 
```

d. Modified table 2 (rule 2, 3, 4, 9, 10 - 10c):



Rule #	Src Host	Src IP	Dst Host	Dst IP	Protocol	Action
1		any		any	ARP	accept
2		any	except dnsServer	any but dnsServer	ICMP	accept
3	facultyWS, facultyPC, labWS, studentPC itPC, itWS, trustedPC, guestPC		webServer		TCP	accept
4	webServer		facultyWS, facultyPC, labWS, studentPC itPC, itWS, trustedPC, guestPC		TCP	accept
5	facultyWS, facultyPC		examServer		TCP	accept
6	examServer		facultyWS, facultyPC		TCP	accept
7	itWS, itPC		facultyWS, facultyPC, labWS, studentPC, itWS, itPC		TCP or UDP	accept
8	facultyWS, facultyPC,		itWS, itPC		TCP or UDP	accept



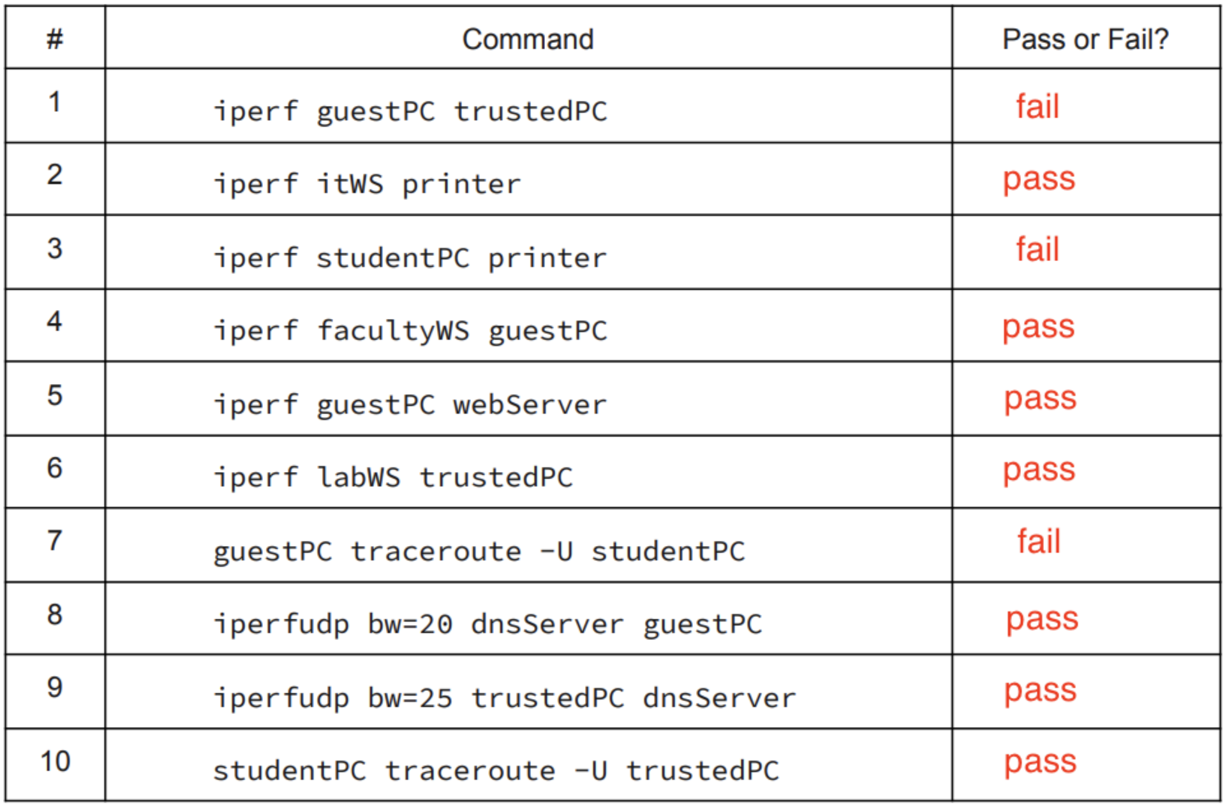
Rule #	Src Host	Src IP	Dst Host	Dst IP	Protocol	Action
	labWS, studentPC, itWS, itPC					
9	facultyWS, facultyPC, labWS, studentPC, itWS, itPC, trustedPC, guestPC		dnsServer		UDP	accept
10	dnsServer		facultyWS, facultyPC, labWS, studentPC itPC, itWS, trustedPC, guestPC		UDP	accept
11		any		any	ANY	drop

Table 2 - Basic Firewall Rules

New tables added before Rule 11 ->

#10.1 (10b)	facultyWS, labWS, itWS		printer		TCP	accept
#10.2 (10b)	printer		facultyWS, labWS, itWS		TCP	accept
#10.3 (10c)						
#1.1 (after rule 1 before rule 2) - (10c)	any		dnsServer		ICMP	drop

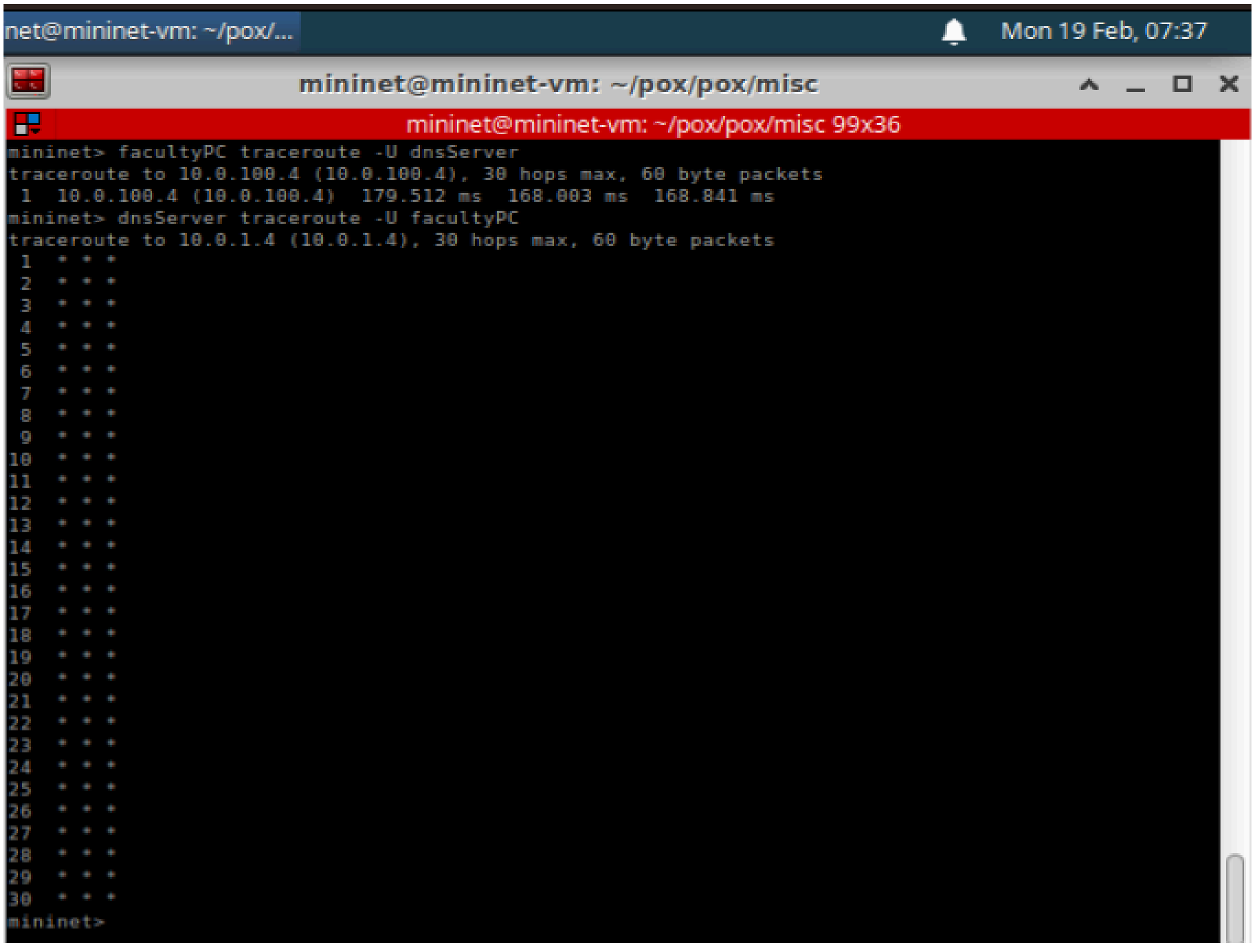
4.



#	Command	Pass or Fail?
1	`iperf guestPC trustedPC`	fail
2	`iperf itWS printer`	pass
3	`iperf studentPC printer`	fail
4	`iperf facultyWS guestPC`	pass
5	`iperf guestPC webServer`	pass
6	`iperf labWS trustedPC`	pass
7	`guestPC traceroute -U studentPC`	fail
8	`iperfudp bw=20 dnsServer guestPC`	pass
9	`iperfudp bw=25 trustedPC dnsServer`	pass
10	`studentPC traceroute -U trustedPC`	pass

5. The traceroute forcing UDP probes between dnsServer and facultyPC doesn’t work now because question 10a’s DoS attack forced me to block ICMP packet leaving from the dnsServer. This means that when dnsServer tries to send an ICMP reply to acknowledge

receipt of the UDP probe, it is dropped due to the new rule I added #1.1.


net@mininet-vm: ~/pox/...
Mon 19 Feb, 07:37
mininet@mininet-vm: ~/pox/pox/misc
mininet@mininet-vm: ~/pox/pox/misc 99x36

```
mininet> facultyPC traceroute -U dnsServer
traceroute to 10.0.100.4 (10.0.100.4), 30 hops max, 60 byte packets
 1  10.0.100.4 (10.0.100.4)  179.512 ms  168.003 ms  168.841 ms
mininet> dnsServer traceroute -U facultyPC
traceroute to 10.0.1.4 (10.0.1.4), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
mininet>
```