

q1:

```
iptables -I INPUT -j DROP
```

q2:

```
iptables -I INPUT -p icmp -j ACCEPT
```

q3:

```
iptables -I INPUT -p TCP --dport 5000 -j ACCEPT
```

```
iptables -I INPUT -p TCP --dport 900 -s 10.0.5.10 -j ACCEPT
```

q4:

The image displays four terminal windows. The top-left window shows the iperf2 help text, including options like --reportexclude, --reportstyle, and --help. The top-right window shows an nmap scan report for 10.0.2.10, indicating that ports 900/tcp and 5000/tcp are filtered. The bottom-left window shows an iperf server listening on TCP port 900, with a TCP window size of 128 KByte. The bottom-right window shows an nmap scan report for 10.0.2.10, indicating that ports 900/tcp and 5000/tcp are open.

```
Miscellaneous:
-X, --reportexclude [CDMSV]  exclude C(connection) D(data) M(multicast) S(set
ettings) V(server) reports
-y, --reportstyle C          report as a Comma-Separated Values
-h, --help                  print this message and quit
-v, --version               print version information and quit

[kmgKMG] Indicates options that support a k,m,g,K,M or G suffix
Lowercase format characters are 10^3 based and uppercase are 2^n based
(e.g. 1k = 1000, 1K = 1024, 1m = 1,000,000 and 1M = 1,048,576)

The TCP window size option can be set by the environment variable
TCP_WINDOW_SIZE. Most other options can be set by an environment variable
IPERF_<long option name>, such as IPERF_BANDWIDTH.

Source at <http://sourceforge.net/projects/iperf2/>
Report bugs to <iperf-users@lists.sourceforge.net>
root@n3:/tmp/pycore.38323/n3.conf# iperf -s -p 900,5000
-----
Server listening on TCP port 900
TCP window size: 128 KByte (default)
-----

iperf: ignoring extra argument -- 5000
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
^Croot@n3:/tmp/pycore.38323/n3.conf# ^C
root@n3:/tmp/pycore.38323/n3.conf# iperf -s -p 500
iperf: ignoring extra argument -- p
iperf: ignoring extra argument -- 500
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
^Croot@n3:/tmp/pycore.38323/n3.conf# ^C
root@n3:/tmp/pycore.38323/n3.conf# ^C
root@n3:/tmp/pycore.38323/n3.conf# ^C
root@n3:/tmp/pycore.38323/n3.conf# ^C
root@n3:/tmp/pycore.38323/n3.conf# iperf -s -p 5000
-----
Server listening on TCP port 5000
TCP window size: 128 KByte (default)
-----

Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.0.2.10
Host is up (0.00012s latency).

PORT      STATE SERVICE
900/tcp   filtered omginitialrefs
5000/tcp  closed  upnp

Nmap done: 1 IP address (1 host up) scanned in 1.40 seconds
root@n2:/tmp/pycore.38323/n2.conf# nmap 10.0.2.10 -p 900,5000
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-13 13:40 WET
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or sp
ecify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.0.2.10
Host is up (0.00021s latency).

PORT      STATE SERVICE
900/tcp   filtered omginitialrefs
5000/tcp  open    upnp

Nmap done: 1 IP address (1 host up) scanned in 1.39 seconds
root@n2:/tmp/pycore.38323/n2.conf#

Port scanning is often
ities on a host or in a
systems that flag por
ts without permission
network, as our netw
a nastygram. Feel fre
r machine.

p checks a variety of
specific ports to scan.
s of ports 1000 and 1
You can use man nmap.
server using nmap to co
showing the nmap output

address translation) is a
--system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.0.2.10
Host is up (0.00025s latency).

PORT      STATE SERVICE
900/tcp   open    omginitialrefs
5000/tcp  open    upnp

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
root@n7:/tmp/pycore.38323/n7.conf#
```

q5:

```
iptables -t NAT -A POSTROUTING -o eth3 -j MASQUERADE
```

q6:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
2	0.001156299	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
3	4.002219411	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
4	6.003011074	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
5	7.357751428	fe80::200:ff:feaa:3	ff02::5	OSPF	90	Hello Packet
6	8.004554225	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
7	10.005448913	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
8	12.006252106	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
9	12.055231405	10.0.1.20	10.0.5.10	ICMP	98	Echo (ping) request
10	12.055369021	10.0.5.10	10.0.1.20	ICMP	98	Echo (ping) reply
11	13.061280545	10.0.1.20	10.0.5.10	ICMP	98	Echo (ping) request
12	13.061392043	10.0.5.10	10.0.1.20	ICMP	98	Echo (ping) reply
13	14.006781904	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
14	14.085264295	10.0.1.20	10.0.5.10	ICMP	98	Echo (ping) request
15	14.085359605	10.0.5.10	10.0.1.20	ICMP	98	Echo (ping) reply
16	16.007392035	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet

  

No.	Time	Source	Destination	Protocol	Length	Info
3	4.002372449	10.0.5.1	224.0.0.5	OSPF	78	Hello Packet
4	4.287099291	fe80::200:ff:feaa:a	ff02::5	OSPF	90	Hello Packet
5	6.002838879	10.0.5.1	224.0.0.5	OSPF	78	Hello Packet
6	6.003301927	10.0.5.1	224.0.0.5	OSPF	78	Hello Packet
7	7.052669041	10.0.3.1	10.0.5.10	ICMP	98	Echo (ping) request
8	8.052690180	10.0.5.10	10.0.3.1	ICMP	98	Echo (ping) reply
9	9.058699526	10.0.3.1	10.0.5.10	ICMP	98	Echo (ping) request
10	9.058724382	10.0.5.10	10.0.3.1	ICMP	98	Echo (ping) reply
11	10.003959453	10.0.5.1	224.0.0.5	OSPF	78	Hello Packet
12	10.082667706	10.0.3.1	10.0.5.10	ICMP	98	Echo (ping) request
13	10.082686582	10.0.5.10	10.0.3.1	ICMP	98	Echo (ping) reply
14	12.004590897	10.0.5.1	224.0.0.5	OSPF	78	Hello Packet
15	13.282540597	00:00:00:aa:00:0a	00:00:00:aa:00:0a	ARP	42	Who has 10.0.5.1? Tell me
16	13.282651053	00:00:00:aa:00:0a	00:00:00:aa:00:0a	ARP	42	Who has 10.0.5.10? Tell me
17	13.282783941	00:00:00:aa:00:0a	00:00:00:aa:00:0a	ARP	42	10.0.5.1 is at 00:00:00:00:00:00
18	13.282692059	00:00:00:aa:00:0a	00:00:00:aa:00:0a	ARP	42	10.0.5.10 is at 00:00:00:00:00:00
19	14.005598550	10.0.5.1	224.0.0.5	OSPF	78	Hello Packet
20	14.247032063	fe80::200:ff:feaa:a	ff02::5	OSPF	90	Hello Packet

Using ping, n7 thinks it is talking to n4, because there's a mask connection to n7 (right Wireshark window in screenshot). On the other side, n2 thinks it is talking directly to n7 (left Wireshark window in screenshot).

q7:

Temos de configurar as chains de PREROUTING e FORWARDING para o n4 saber de que forma tratar o packet que neste caso iria ser redirecionado da port 80 do n4 para a port 80 do n3.

q8:

iptables -A PREROUTING -t nat -i eth3 -p TCP --dport 80 -j DNAT --to 10.0.2.10:80  
iptables -A FORWARD -p TCP -d 10.0.2.10 --dport 80 -j ACCEPT

q9:

No, port-forwarding is transparent, so n3 doesn't realize it is happening. It just sees the packets coming from n7.

No.	Time	Source	Destination	Protocol	Length	Info
5839	10.142349147	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142349808	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142350490	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142351092	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142351844	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142352636	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142353318	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142367458	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142368129	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142368871	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142369563	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142370336	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142371027	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142371699	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142372231	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142372822	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142373414	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0
5839	10.142374076	10.0.5.10	10.0.2.10	TCP	1514	41588 → 80 [ACK] Seq=1323857462, Win=0, Len=0

  

No.	Time	Source	Destination	Protocol	Length	Info
8458	10.006945270	10.0.3.1	10.0.5.10	TCP	66	80 → 41588 [FIN, ACK] Seq=1795623882, Win=0, Len=0
8458	10.007105486	10.0.3.1	10.0.5.10	TCP	66	80 → 41588 [PSH, ACK] Seq=1795623882, Win=0, Len=0
8459	10.007133624	10.0.5.10	10.0.3.1	TCP	66	41586 → 80 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.007173716	10.0.5.10	10.0.3.1	TCP	70	41586 → 80 [PSH, ACK] Seq=1795623882, Win=0, Len=0
8459	10.007183182	10.0.3.1	10.0.5.10	TCP	66	80 → 41586 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.007191926	10.0.5.10	10.0.3.1	TCP	357	41586 → 80 [PSH, ACK] Seq=1795623882, Win=0, Len=0
8459	10.007199547	10.0.3.1	10.0.5.10	TCP	66	80 → 41586 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.007249968	10.0.3.1	10.0.5.10	TCP	70	80 → 41586 [PSH, ACK] Seq=1795623882, Win=0, Len=0
8459	10.007265130	10.0.5.10	10.0.3.1	TCP	66	41586 → 80 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.007277795	10.0.3.1	10.0.5.10	TCP	358	80 → 41586 [PSH, ACK] Seq=1795623882, Win=0, Len=0
8459	10.007280794	10.0.5.10	10.0.3.1	TCP	66	41586 → 80 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.007315931	10.0.5.10	10.0.3.1	TCP	66	41588 → 80 [FIN, ACK] Seq=1795623882, Win=0, Len=0
8459	10.007340339	10.0.3.1	10.0.5.10	TCP	66	80 → 41588 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.011502940	10.0.5.10	10.0.3.1	TCP	67	41586 → 80 [PSH, ACK] Seq=1795623882, Win=0, Len=0
8459	10.011634670	10.0.3.1	10.0.5.10	TCP	66	80 → 41586 [ACK] Seq=1795623882, Win=0, Len=0
8459	10.011659810	10.0.3.1	10.0.5.10	TCP	66	80 → 41586 [FIN, ACK] Seq=1795623882, Win=0, Len=0
8459	10.012066813	10.0.5.10	10.0.3.1	TCP	66	41586 → 80 [FIN, ACK] Seq=1795623882, Win=0, Len=0
8459	10.012137650	10.0.3.1	10.0.5.10	TCP	66	80 → 41586 [ACK] Seq=1795623882, Win=0, Len=0