Firewall Lab:

Task 1: Using Firewall

We have machine A as 10.0.2.10 and machine B as 10.0.2.7

Prevent A from doing telnet to Machine B.

```
[04/02/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Apr  2 07:04:55 EDT 2020 from 10.0.2.10 on pts/4
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

We can do the telnet from A to B originally, to prevent A from doing telnet to machine B:

We need to append rules on machine A by dropping the packets from machine A to B:

```
[04/02/20]seed@VM:~$ sudo iptables -A OUTPUT -s 10.0.2.10 -d 10.0.2.7 -p tcp --d
port 23 -j DROP
[04/02/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
telnet: Unable to connect to remote host: Connection timed out
```

Check the iptables:

```
[04/02/20]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.10           10.0.2.7             tcp dpt:telnet
```

Prevent B from doing telnet to Machine A.

```
[04/02/20]seed@VM:~$ telnet 10.0.2.10
Trying 10.0.2.10...
Connected to 10.0.2.10.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Apr  2 06:50:13 EDT 2020 from 192.168.60.1 on pts/17
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

Originally we could do the telnet from machine B to A:

We do the same:

```
[04/02/20]seed@VM:~$ sudo iptables -A OUTPUT -s 10.0.2.7 -d 10.0.2.10 -p tcp --d
port 23 -j DROP
[04/02/20]seed@VM:~$ telnet 10.0.2.10
Trying 10.0.2.10...
```

Prevent A from visiting an external web site.

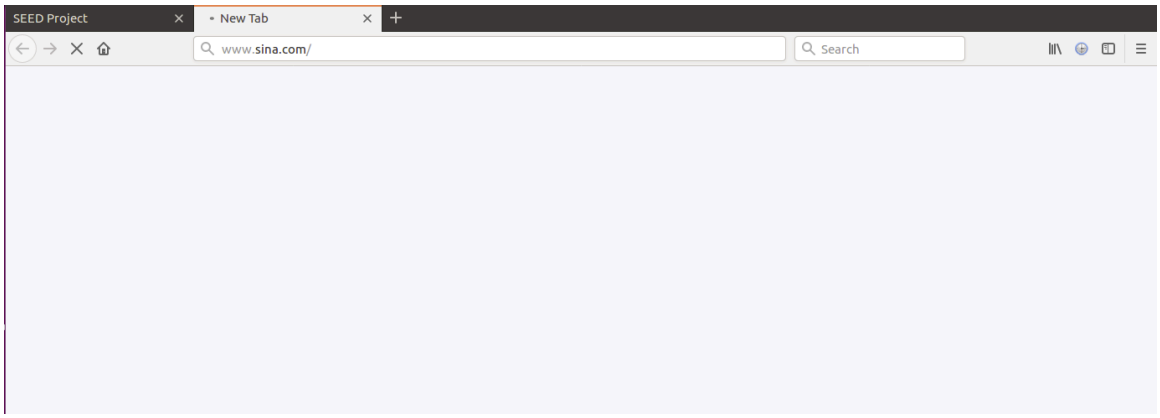
We could browse sina.com. To investigate, we want find the ip of the answer section

```
;; ANSWER SECTION:
www.sina.com.           60      IN      CNAME   us.sina.com.cn.
us.sina.com.cn.         60      IN      CNAME   wwwus.sina.com.
wwwus.sina.com.         60      IN      CNAME   ww1.sinaimg.cn.w.alikunlun.com.
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.228
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.230
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.225
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.227
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.229
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.232
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.226
ww1.sinaimg.cn.w.alikunlun.com. 1 IN    A       47.246.24.231
```

We do the same by drop the packet from local address to this specific address:

```
[04/02/20]seed@VM:~$ sudo iptables -A OUTPUT -s 10.0.2.7 -d 47.246.24.0/24 -p tc
p --dport 80 -j DROP
[04/02/20]seed@VM:~$
```

After adding this rule:



We refresh the page, it’ s unreachable.

Task 2: Implementing a Simple Firewall

Block telnet from A to B

```
{
  struct iphdr *iph;
  struct tcphdr *tcph;

  iph = ip_hdr(skb);
  tcph = (void *)iph+iph->ihl*4;

  if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) && iph->daddr == in_aton("10.0.2.7")) {
    printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
        ((unsigned char *)&iph->daddr)[0],
        ((unsigned char *)&iph->daddr)[1],
        ((unsigned char *)&iph->daddr)[2],
        ((unsigned char *)&iph->daddr)[3]);
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}


int setUpFilter(void) {
        printk(KERN_INFO "Registering a Telnet filter.\n");
        telnetFilterHook.hook = telnetFilter;
        telnetFilterHook.hooknum = NF_INET_LOCAL_OUT;
        telnetFilterHook.pf = PF_INET;
        telnetFilterHook.priority = NF_IP_PRI_FIRST;

        // Register the hook.
        nf_register_hook(&telnetFilterHook);
        return 0;
}
```

Set the filter where the destination address is 10.0.2.7, port is 23 and protocol is tcp

```
[04/02/20]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Desktop/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Desktop/packet_filter/telnetFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Desktop/packet_filter/telnetFilter.mod.o
  LD [M]  /home/seed/Desktop/packet_filter/telnetFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
```

```
[04/02/20]seed@VM:~/.../packet_filter$ ls
Makefile       Module.symvers  telnetFilter.ko     telnetFilter.mod.o
modules.order  telnetFilter.c  telnetFilter.mod.c  telnetFilter.o
[04/02/20]seed@VM:~/.../packet_filter$ sudo insmod telnetFilter.ko
[04/02/20]seed@VM:~/.../packet_filter$ lsmod | grep telnetFilter
telnetFilter           16384  0
```

Insert module to the kernel, the we find we cannot telnet 10.0.2.7 anymore

```
[04/02/20]seed@VM:~/.../packet_filter$ telnet 10.0.2.7
Trying 10.0.2.7...
telnet: Unable to connect to remote host: Connection timed out
```

We cannot telnet 10.0.2.7 anymore

After we remove this:

```
[04/02/20]seed@VM:~/.../packet_filter$ sudo rmmod telnetFilter
[04/02/20]seed@VM:~/.../packet_filter$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login:
```

We can do the telnet

Block telnet from B to A

Similar to last, we do:

```
if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) && iph->saddr == in_aton("10.0.2.7")) {
  printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
      ((unsigned char *)&iph->daddr)[0],
      ((unsigned char *)&iph->daddr)[1],
      ((unsigned char *)&iph->daddr)[2],
      ((unsigned char *)&iph->daddr)[3]);
  return NF_DROP;
} else {
  return NF_ACCEPT;
}


nt setUpFilter(void) {
      printk(KERN_INFO "Registering a Telnet filter.\n");
      telnetFilterHook.hook = telnetFilter;
      telnetFilterHook.hooknum = NF_INET_LOCAL_IN;
      telnetFilterHook.pf = PF_INET;
      telnetFilterHook.priority = NF_IP_PRI_FIRST;
```

on machine A we drop the packet sent by B that is incoming, we change to saddr is B.

then we make all and insert the module:

```
[04/02/20]seed@VM:~/.../packet_filter$ make all
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Desktop/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Desktop/packet_filter/telnetFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Desktop/packet_filter/telnetFilter.mod.o
  LD [M]  /home/seed/Desktop/packet_filter/telnetFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[04/02/20]seed@VM:~/.../packet_filter$ sudo insmod telnetFilter.ko
```

Try telnet from B to A

```
[04/02/20]seed@VM:~$ telnet 10.0.2.10
Trying 10.0.2.10...
telnet: Unable to connect to remote host: Connection timed out
[04/02/20]seed@VM:~$
```

After we remove this:

```
[04/02/20]seed@VM:~/.../packet_filter$ sudo rmmod telnetFilter
[04/02/20]seed@VM:~/.../packet_filter$
```

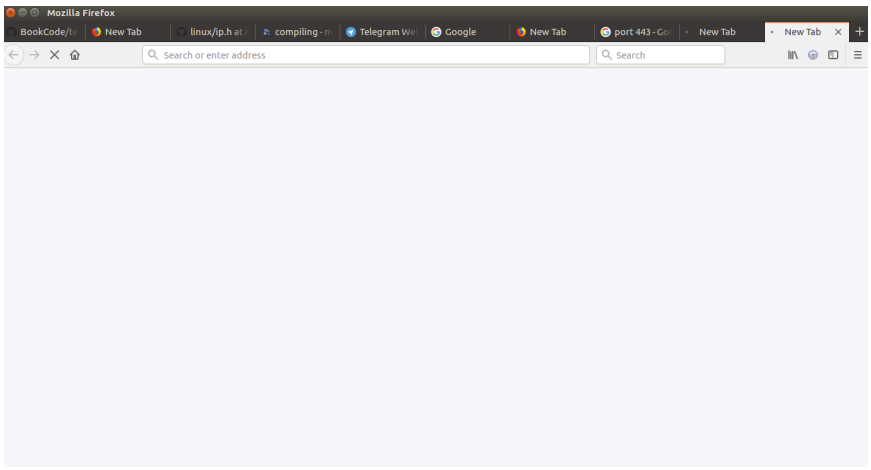
```
[04/02/20]seed@VM:~$ telnet 10.0.2.10
Trying 10.0.2.10...
Connected to 10.0.2.10.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login:
```

Block external website access from A

```
  if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(443) && iph->saddr == in_aton("10.0.2.10")) 
    printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
        ((unsigned char *)&iph->daddr)[0],
        ((unsigned char *)&iph->daddr)[1],
        ((unsigned char *)&iph->daddr)[2],
        ((unsigned char *)&iph->daddr)[3]);
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}


int setUpFilter(void) {
        printk(KERN_INFO "Registering a Telnet filter.\n");
        telnetFilterHook.hook = telnetFilter;
        telnetFilterHook.hooknum = NF_INET_LOCAL_OUT;
        telnetFilterHook.pf = PF_INET;
        telnetFilterHook.priority = NF_IP_PRI_FIRST;
```

We do the same make and insert:



As all of the traffic to port 443 has been dropped, no page would be viewed anymore:

Port 443 corresponds to https.

Block ssh from A to B

By default, ssh runs on port 22, so we change the port to 22 and other is the block A to B telnet:

```
if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(22) && iph->daddr == in_aton("10.0.2.7")) {
  printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
      ((unsigned char *)&iph->daddr)[0],
      ((unsigned char *)&iph->daddr)[1],
      ((unsigned char *)&iph->daddr)[2],
      ((unsigned char *)&iph->daddr)[3]);
  return NF_DROP;
} else {
  return NF_ACCEPT;
}


nt setUpFilter(void) {
      printk(KERN_INFO "Registering a Telnet filter.\n");
      telnetFilterHook.hook = telnetFilter;
      telnetFilterHook.hooknum = NF_INET_LOCAL_OUT;
      telnetFilterHook.pf = PF_INET;
      telnetFilterHook.priority = NF_IP_PRI_FIRST;
```

```
[04/02/20]seed@VM:~/.../packet_filter$ ssh 10.0.2.7
```

Block ssh from B to A

And vice versa we have

```
  if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(22) && iph->saddr == in_aton("10.0.2.7")) {
    printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
        ((unsigned char *)&iph->daddr)[0],
        ((unsigned char *)&iph->daddr)[1],
        ((unsigned char *)&iph->daddr)[2],
        ((unsigned char *)&iph->daddr)[3]);
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}


int setUpFilter(void) {
        printk(KERN_INFO "Registering a Telnet filter.\n");
        telnetFilterHook.hook = telnetFilter;
        telnetFilterHook.hooknum = NF_INET_LOCAL_IN;
        telnetFilterHook.pf = PF_INET;
        telnetFilterHook.priority = NF_IP_PRI_FIRST;
```

```
[04/02/20]seed@VM:~$ ssh 10.0.2.10
```

Task 3: Evading Egress Filtering

To Block all the outgoing traffic to external telnet servers on machine A

```
[04/02/20]seed@VM:~/.../packet_filter$ sudo iptables -A OUTPUT -p tcp --dport 23 -j DROP
```

```
[04/02/20]seed@VM:~/.../packet_filter$ telnet 10.0.2.7
Trying 10.0.2.7...
^C
```

To Block all the outgoing traffic to www.facebook.com

We first try dig www.facebook.com

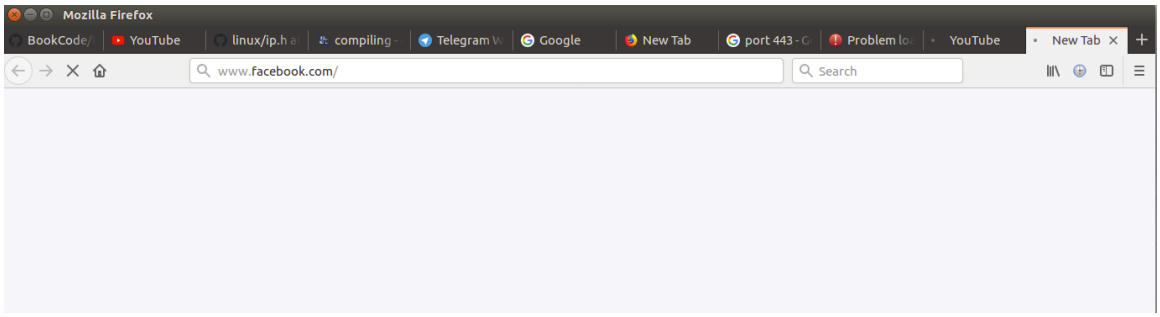
```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54880
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.facebook.com.              IN      A

;; ANSWER SECTION:
www.facebook.com.       3446    IN      CNAME   star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com. 21 IN      A       157.240.18.35

;; Query time: 9 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Thu Apr 02 20:18:45 EDT 2020
;; MSG SIZE  rcvd: 90
```

```
[04/02/20]seed@VM:~/.../packet_filter$ sudo iptables -A OUTPUT -d 157.240.18.35 -p tcp --dport 443 -j DROP
```



Then we cannot browse faceboook.com anymore:

```
[04/02/20]seed@VM:~/.../packet_filter$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             edge-star-mini-shv-02-ort2.facebook.com  tcp dpt:https
[04/02/20]seed@VM:~/.../packet_filter$
```

Telnet to Machine B through the firewall

We have machine C as 10.0.2.15:

```
[04/02/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
```

With the firewall, we cannot telnet 10.0.2.15, so we try ssh to machine B:
establishes an SSH tunnel between the localhost (port 8000) and machine B (using the default port 22)

```
[04/02/20]seed@VM:~$ ssh -L 8000:10.0.2.15:23  seed@10.0.2.7
The authenticity of host '10.0.2.7 (10.0.2.7)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Thu Apr  2 20:32:48 2020 from 10.0.2.10
[04/02/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:12:9b:fe
          inet addr:10.0.2.7  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6dc6:8637:7e4a:d4fd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

As now we are now on machine B already, we should be able to telnet 10.0.2.15 directly:

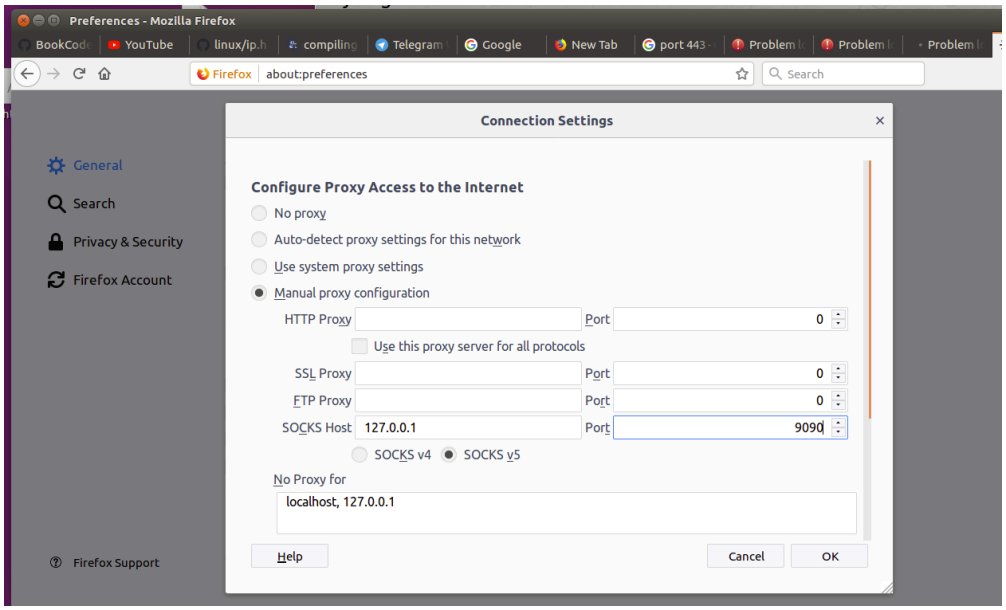
```
[04/02/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
```

On the middle machine, we find that machine keep sending the packet 10.0.2.15 as a station on the tcp protocol. We use ssh to bypass the firewall and telnet to the machine that we want to reach.

```
    70 2020-04-02 21:13:52.7926102… 10.0.2.10      10.0.2.7       SSH      104 Client: Enc
    74 2020-04-02 21:13:52.7935844… 10.0.2.7       10.0.2.10      SSH      112 Server: Enc
    76 2020-04-02 21:13:52.9386266… 10.0.2.10      10.0.2.7       SSH      104 Client: Enc
    80 2020-04-02 21:13:52.9395378… 10.0.2.7       10.0.2.10      SSH      112 Server: Enc
    82 2020-04-02 21:13:53.1243310… 10.0.2.10      10.0.2.7       SSH      104 Client: Enc
    86 2020-04-02 21:13:53.1252302… 10.0.2.7       10.0.2.10      SSH      112 Server: Enc
    88 2020-04-02 21:13:53.6144862… 10.0.2.10      10.0.2.7       SSH      104 Client: Enc
    92 2020-04-02 21:13:53.6180528… 10.0.2.7       10.0.2.10      SSH      104 Server: Enc
    96 2020-04-02 21:13:53.6255750… 10.0.2.7       10.0.2.10      SSH      200 Server: Enc
    99 2020-04-02 21:13:53.6257841… 10.0.2.7       10.0.2.10      SSH      552 Server: Enc
   103 2020-04-02 21:13:53.6260076… 10.0.2.7       10.0.2.10      SSH      608 Server: Enc
   107 2020-04-02 21:13:53.6262208… 10.0.2.7       10.0.2.10      SSH      496 Server: Enc
   112 2020-04-02 21:13:53.6288723… 10.0.2.7       10.0.2.10      SSH      128 Server: Enc
     7 2020-04-02 21:13:49.7993404… 10.0.2.7       10.0.2.15      TCP       68 48258 → 23
     9 2020-04-02 21:13:49.8000275… 10.0.2.10      10.0.2.7       TCP       68 58002 → 22
    13 2020-04-02 21:13:49.9568876… 10.0.2.7       10.0.2.15      TCP       68 48258 → 23
    15 2020-04-02 21:13:49.9575660… 10.0.2.10      10.0.2.7       TCP       68 58002 → 22
    19 2020-04-02 21:13:50.1717283… 10.0.2.7       10.0.2.15      TCP       68 48258 → 23
    21 2020-04-02 21:13:50.1722716… 10.0.2.10      10.0.2.7       TCP       68 58002 → 22
    25 2020-04-02 21:13:50.2328600… 10.0.2.7       10.0.2.15      TCP       68 48258 → 23
    27 2020-04-02 21:13:50.2333714… 10.0.2.10      10.0.2.7       TCP       68 58002 → 22
```

Connect to Facebook using SSH Tunnel.

Set proxy



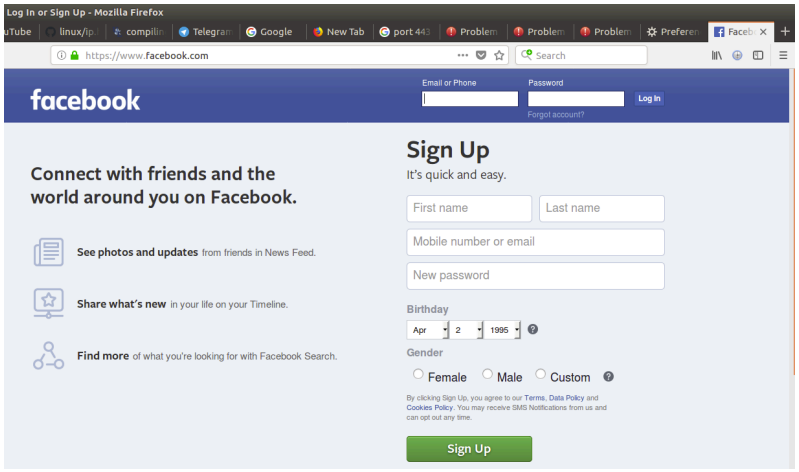
Then run ssh for dynamic port forwarding, we should be able to browse fb:

```
[04/02/20]seed@VM:~$ ssh -D 9090 -C 10.0.2.15
seed@10.0.2.15's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Thu Apr  2 21:32:19 2020 from 10.0.2.10
[04/02/20]seed@VM:~$
```



Through wireshark:

The browser would first go to the local proxy, then ssh package is sent through the tunnel to .15, then .15 machine C would connect to the fb server

282	2020-04-02 21:45:11.1894643…	10.0.2.10	10.0.2.15	SSH	136	Client: E…
283	2020-04-02 21:45:11.1897372…	10.0.2.15	10.0.2.10	TCP	68	22 → 5309…
284	2020-04-02 21:45:11.4023992…	127.0.0.1	127.0.0.1	TCP	408	41248 → 9…
285	2020-04-02 21:45:11.4024097…	127.0.0.1	127.0.0.1	TCP	68	9090 → 41…
286	2020-04-02 21:45:11.4025309…	10.0.2.10	10.0.2.15	SSH	448	Client: E…
287	2020-04-02 21:45:11.4029866…	10.0.2.15	10.0.2.10	TCP	68	22 → 5309…
288	2020-04-02 21:45:11.4033972…	127.0.0.1	127.0.0.1	TCP	110	41248 → 9…
289	2020-04-02 21:45:11.4034012…	127.0.0.1	127.0.0.1	TCP	68	9090 → 41…
290	2020-04-02 21:45:11.4034518…	10.0.2.10	10.0.2.15	SSH	136	Client: E…
291	2020-04-02 21:45:11.4037815…	10.0.2.15	10.0.2.10	TCP	68	22 → 5309…
292	2020-04-02 21:45:11.7549005…	127.0.0.1	127.0.0.1	TCP	1087	41248 → 9…
293	2020-04-02 21:45:11.7549073…	127.0.0.1	127.0.0.1	TCP	68	9090 → 41…
294	2020-04-02 21:45:11.7552019…	10.0.2.10	10.0.2.15	SSH	1128	Client: E…
295	2020-04-02 21:45:11.7555288…	10.0.2.15	10.0.2.10	TCP	68	22 → 5309…
296	2020-04-02 21:45:11.8038642…	10.0.2.15	10.0.2.10	SSH	1568	Server: E…
297	2020-04-02 21:45:11.8039402…	10.0.2.10	10.0.2.15	TCP	68	53098 → 2…
298	2020-04-02 21:45:11.8040637…	127.0.0.1	127.0.0.1	TCP	1528	9090 → 41…
299	2020-04-02 21:45:11.8458977…	10.0.2.15	10.0.2.10	SSH	488	Server: E…
300	2020-04-02 21:45:11.8458856…	127.0.0.1	127.0.0.1	TCP	68	41248 → 9…
301	2020-04-02 21:45:11.8459842…	127.0.0.1	127.0.0.1	TCP	447	9090 → 41…

On machine C, we found:

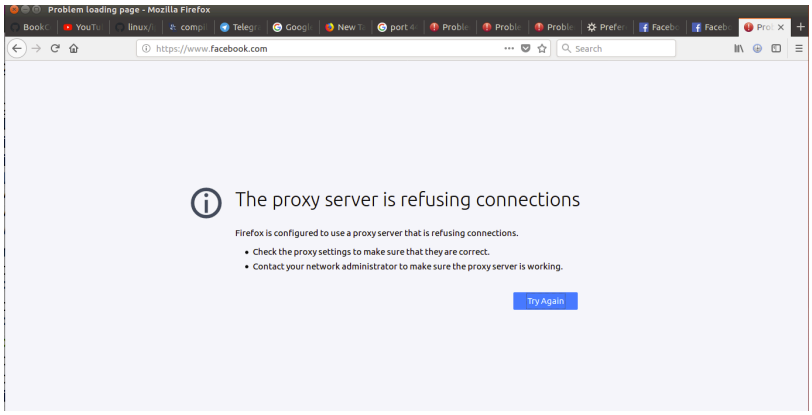
1	2020-04-02 21:49:28.2016565…	::1	::1	UDP	64	54678 → 5…
2	2020-04-02 21:49:34.8344868…	10.0.2.7	224.0.0.251	MDNS	89	Standard …
3	2020-04-02 21:49:34.9337374…	10.0.2.10	224.0.0.251	MDNS	162	Standard …
4	2020-04-02 21:49:35.6598351…	fe80::6dc6:8637:7e4…	ff02::fb	MDNS	109	Standard …
5	2020-04-02 21:49:36.3765686…	fe80::3872:9574:e77…	ff02::fb	MDNS	205	Standard …
6	2020-04-02 21:49:37.0950423…	10.0.2.10	10.0.2.15	SSH	136	Client: E…
7	2020-04-02 21:49:37.0951772…	10.0.2.15	172.217.8.162	TLSv1.2	102	Applicati…
8	2020-04-02 21:49:37.0953493…	10.0.2.10	10.0.2.15	SSH	120	Client: E…
9	2020-04-02 21:49:37.0954408…	10.0.2.15	10.0.2.10	TCP	68	22 → 5309…
10	2020-04-02 21:49:37.0955642…	10.0.2.10	10.0.2.15	SSH	104	Client: E…
11	2020-04-02 21:49:37.0956153…	10.0.2.15	172.217.8.162	TLSv1.2	87	Encrypted…
12	2020-04-02 21:49:37.0957738…	10.0.2.15	172.217.8.162	TCP	56	43148 → 4…
13	2020-04-02 21:49:37.0959232…	172.217.8.162	10.0.2.15	TCP	62	443 → 431…
14	2020-04-02 21:49:37.0959259…	172.217.8.162	10.0.2.15	TCP	62	443 → 431…
15	2020-04-02 21:49:37.1268869…	172.217.8.162	10.0.2.15	TCP	62	443 → 431…
16	2020-04-02 21:49:37.1269052…	10.0.2.15	172.217.8.162	TCP	56	43148 → 4…
17	2020-04-02 21:49:37.1270319…	10.0.2.15	10.0.2.10	SSH	140	Server: E…
18	2020-04-02 21:49:37.1275144…	10.0.2.10	10.0.2.15	TCP	68	53098 → 2…
19	2020-04-02 21:49:37.1275230…	10.0.2.10	10.0.2.15	SSH	104	Client: E…
20	2020-04-02 21:49:37.1707899…	10.0.2.15	10.0.2.10	TCP	68	22 → 5309…
21	2020-04-02 21:49:37.5866851…	10.0.2.10	10.0.2.15	SSH	120	Client: E…

```
▸ Frame 11: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 172.217.8.162
▸ Transmission Control Protocol, Src Port: 43148, Dst Port: 443, Seq: 3057703229,
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Encrypted Alert
```

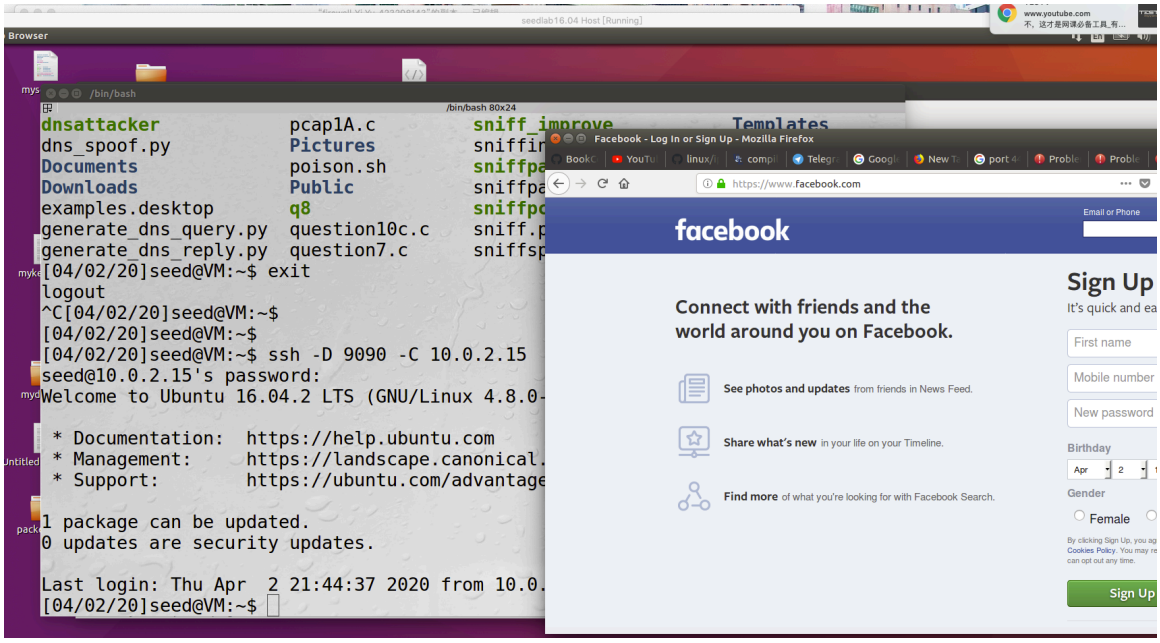
Machine C uses tls to send the application data, all the outbound and inbound data goes to

machine C first.

After we exit the ssh conn,



Reestablish conn, we can log fb again.

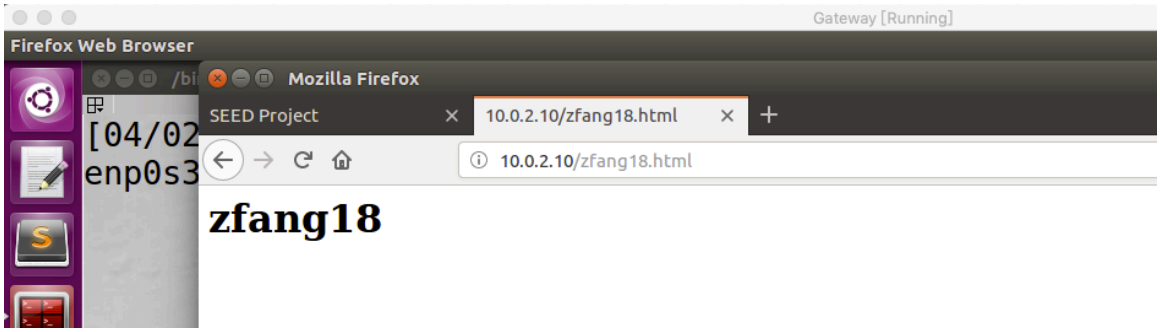


Task 4: Evading Ingress Filtering

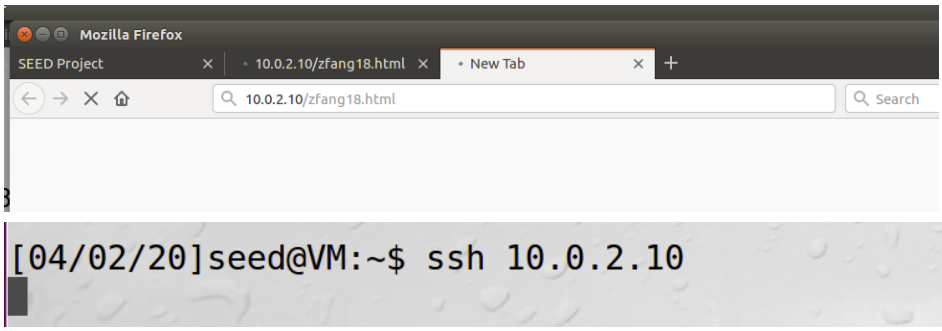
On the machine A, we create an html document:

The path is /var/www/html

On machine B we can browse this document:



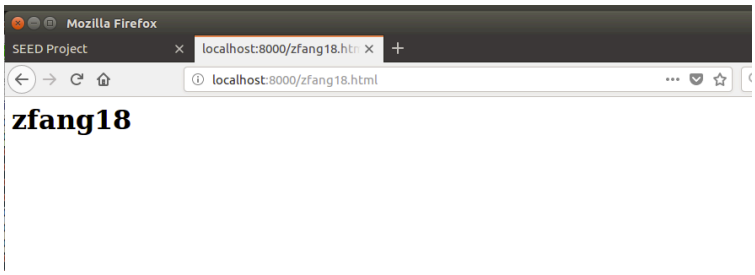
Then we set the rules on A to block ssh port 22 and port 80 for web browsing:



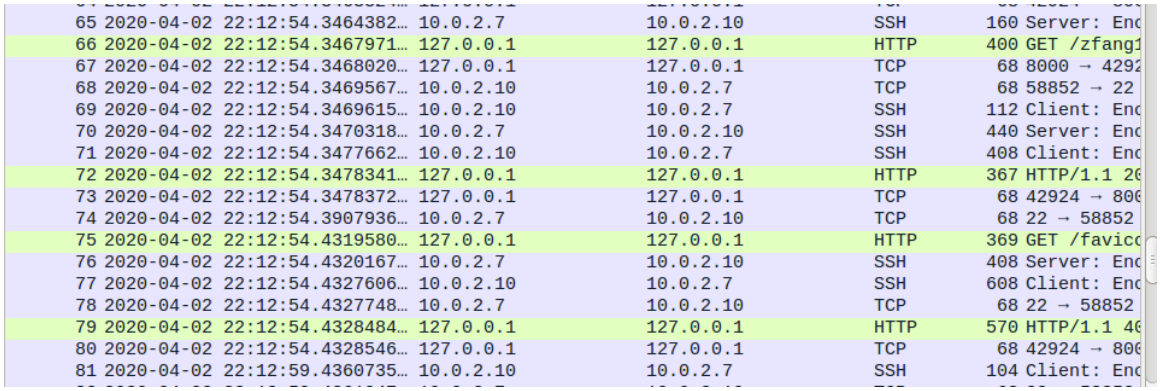
Then on machine A, we should set the reverse SSH tunnel, which means machine A runs the SSH server.

```
[04/02/20]seed@VM:.../html$ ssh -R 8000:10.0.2.10:80 10.0.2.7
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
```



See wireshark:



Task 5 (SNAT): Use iptables to set up a SNAT. Use Wireshark to prove that your SNAT is working.

On server: we

```
[04/02/20]seed@VM:~$ sudo iptables -t nat -A POSTROUTING -o enp0s3 -j SNAT --to-
source 10.0.2.7
[04/02/20]seed@VM:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[04/02/20]seed@VM:~$
```

Then on the internal network machine

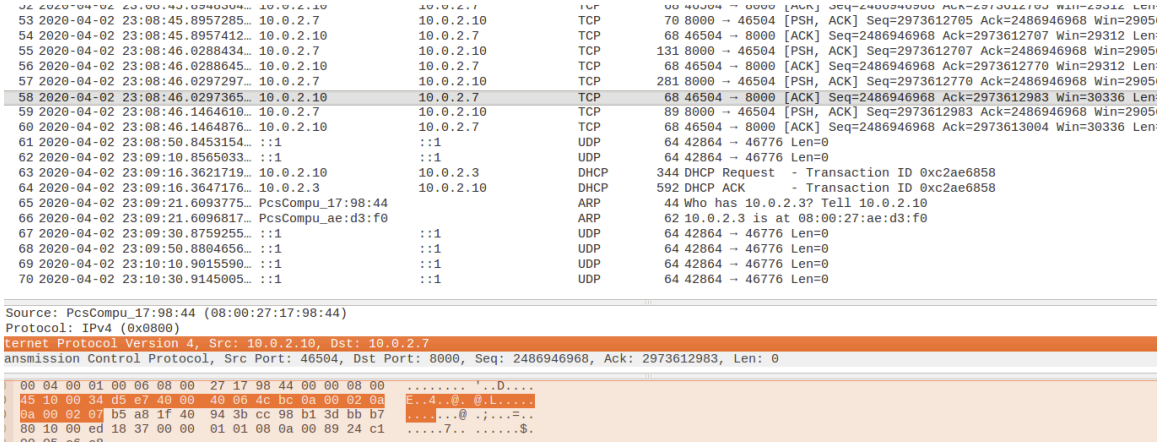
```
[04/02/20]seed@VM:~$ ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=63 time=1.12 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=63 time=1.01 ms
^C
--- 10.0.2.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.018/1.073/1.128/0.055 ms
```

Check the wireshark

1	2020-04-02 23:18:34.1905111…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
2	2020-04-02 23:18:34.1914769…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
3	2020-04-02 23:18:34.1915466…	::1	::1	UDP	64	53761 → 3…
4	2020-04-02 23:18:35.1918765…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
5	2020-04-02 23:18:35.1935353…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
6	2020-04-02 23:18:36.1940656…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
7	2020-04-02 23:18:36.1960579…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
8	2020-04-02 23:18:36.7297744…	192.168.60.1	224.0.0.251	MDNS	185	Standard …
9	2020-04-02 23:18:37.1956355…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
10	2020-04-02 23:18:37.1968293…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
11	2020-04-02 23:18:38.1759404…	fe80::2804:b070:162…	ff02::fb	MDNS	205	Standard …
12	2020-04-02 23:18:38.1973532…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
13	2020-04-02 23:18:38.1983066…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
14	2020-04-02 23:18:39.1982999…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
15	2020-04-02 23:18:39.1993255…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
16	2020-04-02 23:18:40.1992919…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
17	2020-04-02 23:18:40.2003019…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
18	2020-04-02 23:18:41.2008503…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
19	2020-04-02 23:18:41.2018547…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…
20	2020-04-02 23:18:42.2027548…	192.168.60.5	10.0.2.10	ICMP	100	Echo (pin…
21	2020-04-02 23:18:42.2037830…	10.0.2.10	192.168.60.5	ICMP	100	Echo (pin…

Task 6 (DNAT): Use iptables to set up a DNAT for port forwarding. Use Wireshark to prove that your DNAT is working.

```
[04/02/20]seed@VM:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 8000 -j D
NAT --to-destination 192.168.60.5:23
```



in the wireshark, the destination port is still 8000.

Task 7 (DNAT): Use iptables to set up a DNAT for load balancing and demonstrate how it works. In my lecture, I didn't get a perfect load balancing. Can you improve my result?

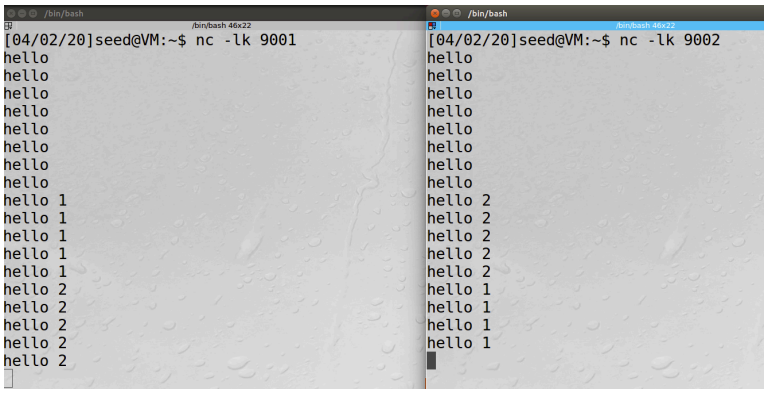
Type this following on machine 10.0.2.7:

```
[04/02/20]seed@VM:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 8000 -m s
tatistic --mode nth --every 2 --packet 0 -j DNAT --to-destination 192.168.60.5:9
001
[04/02/20]seed@VM:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 8000 -m s
tatistic --mode nth --every 2 --packet 1 -j DNAT --to-destination 192.168.60.5:9
002
[04/02/20]seed@VM:~$
```

On machine 10.0.2.10:

```
[04/02/20]seed@VM:~$ echo "hello" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9001
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 2" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$ echo "hello 1" | nc 10.0.2.7 9002
[04/02/20]seed@VM:~$
```

Then:



Task 8 (Connection Track): Set up a firewall rule based on connections.

```
[04/02/20]seed@VM:~$ telnet 10.0.2.10
Trying 10.0.2.10...
Connected to 10.0.2.10.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login:
```

```
[04/02/20]seed@VM:~$ sudo iptables -A OUTPUT -p tcp -m conntrack --ctstate ESTAB
LISHED,RELATED -j ACCEPT
[04/02/20]seed@VM:~$ sudo iptables -A OUTPUT -p tcp -j REJECT
[04/02/20]seed@VM:~$
```

Then:

A cannot conn to B, because there is no established or related record, so it will be rejected.

However b can still conn A

```
[04/02/20]seed@VM:~$ telnet 10.0.2.10
Trying 10.0.2.10...
telnet: Unable to connect to remote host: Connection refused
[04/02/20]seed@VM:~$
```

```
[04/02/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login:
```