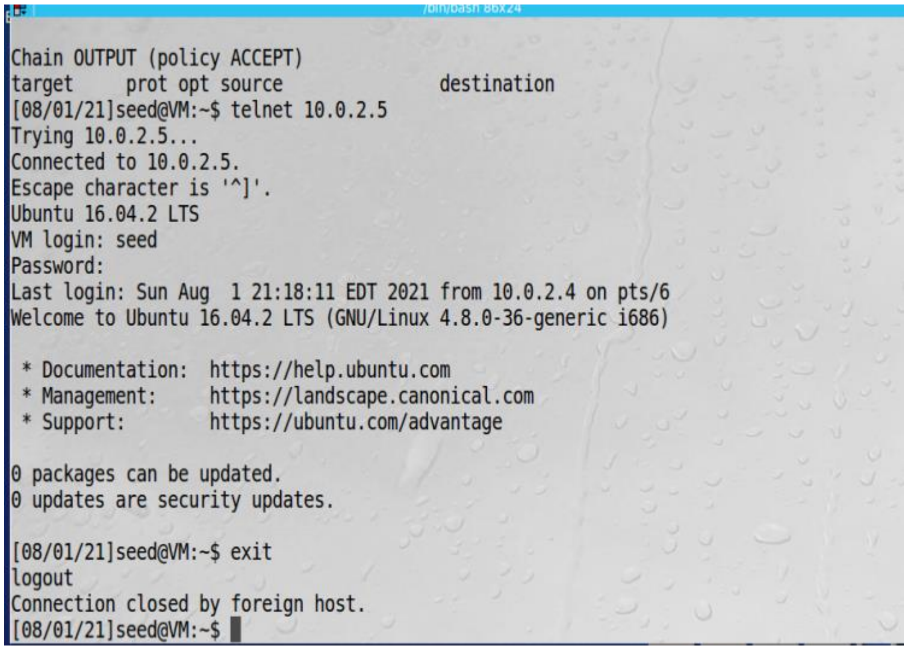
1. Using Firewall

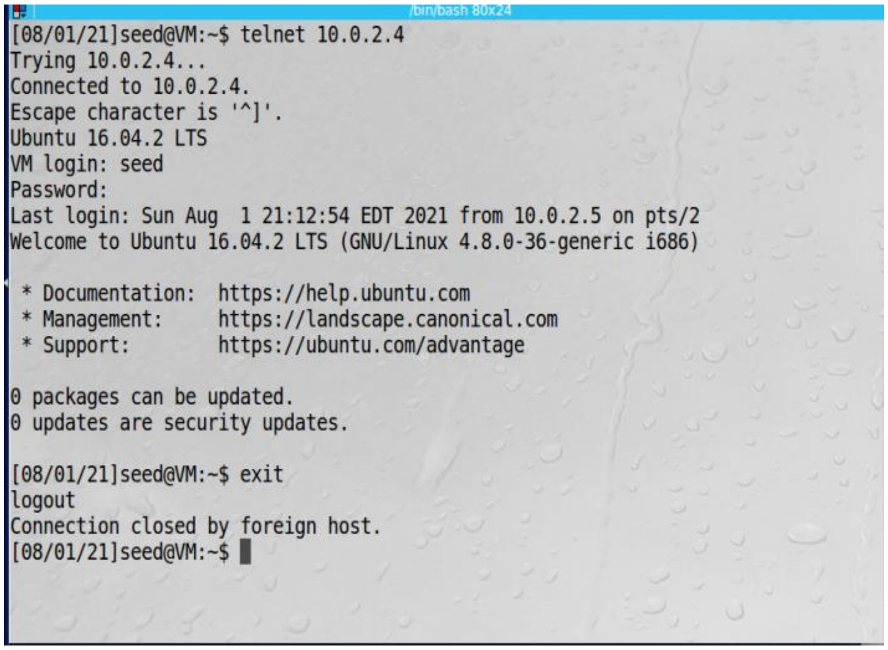
I have 2 VM’s **VM A with ip- 10.0.2.4** and **VM B with ip- 10.0.2.5**

From Machine A connecting to Machine B before the firewall rule was created/enabled:



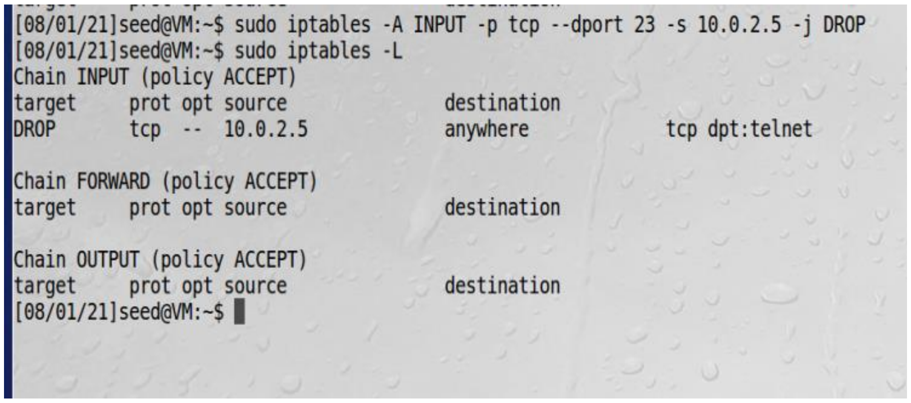
```
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[08/01/21]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Aug  1 21:18:11 EDT 2021 from 10.0.2.4 on pts/6
Welcome 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

0 packages can be updated.
0 updates are security updates.

[08/01/21]seed@VM:~$ exit
logout
Connection closed by foreign host.
[08/01/21]seed@VM:~$
```

From Machine B connecting to Machine A before the firewall rule was created/enabled:



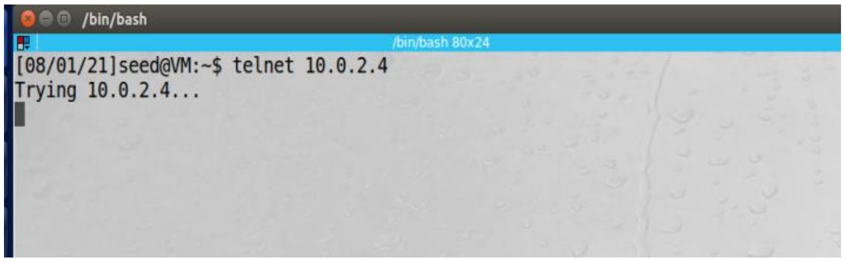
```
[08/01/21]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Aug  1 21:12:54 EDT 2021 from 10.0.2.5 on pts/2
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

0 packages can be updated.
0 updates are security updates.

[08/01/21]seed@VM:~$ exit
logout
Connection closed by foreign host.
[08/01/21]seed@VM:~$
```

Firewall rule created on Machine A blocking telnet from Machine B



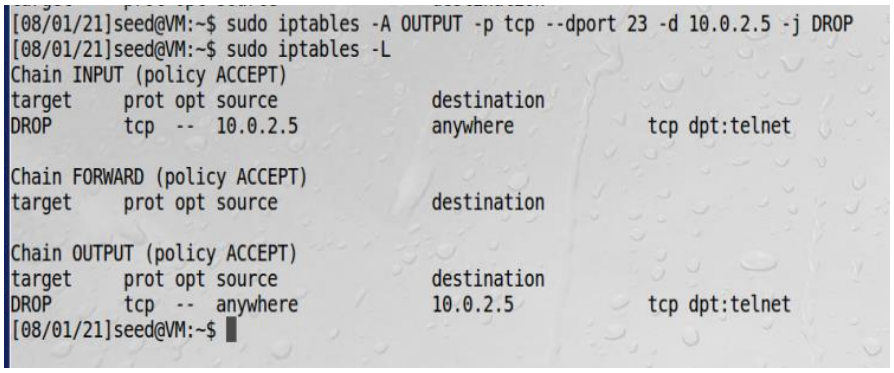
```
[08/01/21]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 23 -s 10.0.2.5 -j DROP
[08/01/21]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.5             anywhere             tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[08/01/21]seed@VM:~$
```

The firewall rule created and enabled on Machine A blocking telnet traffic from Machine B and showing that it is not able to connect:



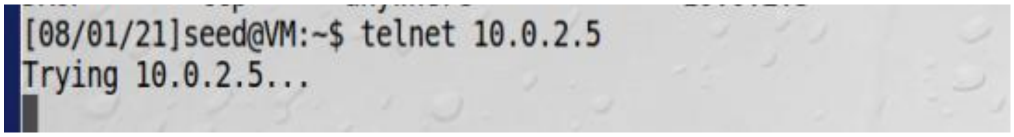
```
[08/01/21]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...
```

Firewall rule created on Machine A blocking telnet to Machine B



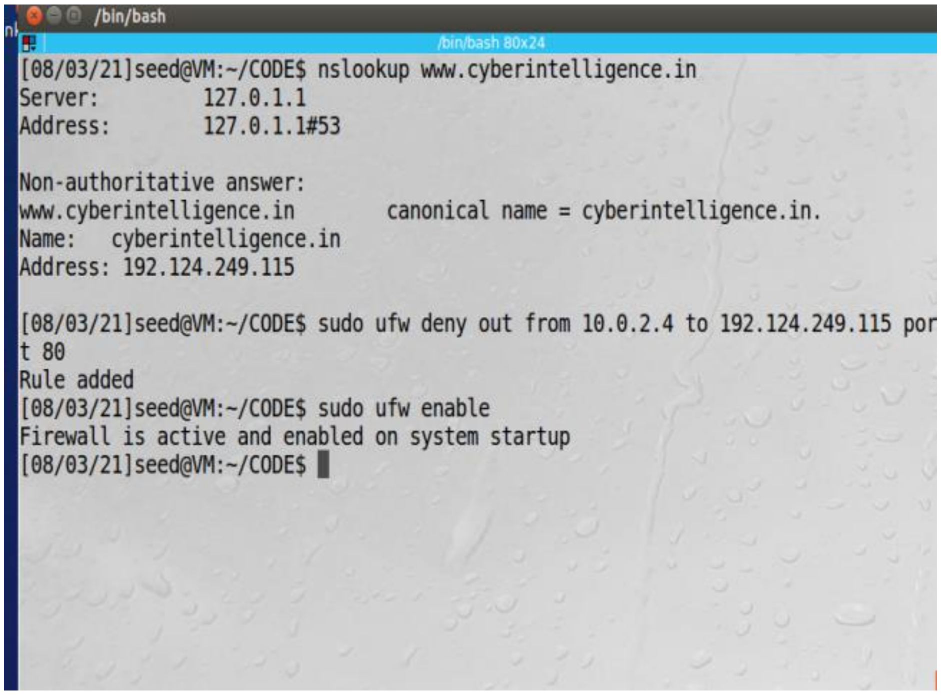
```
[08/01/21]seed@VM:~$ sudo iptables -A OUTPUT -p tcp --dport 23 -d 10.0.2.5 -j DROP
[08/01/21]seed@VM:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.5             anywhere             tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             10.0.2.5             tcp dpt:telnet
[08/01/21]seed@VM:~$
```

Showing that Machine A can not telnet into Machine B:



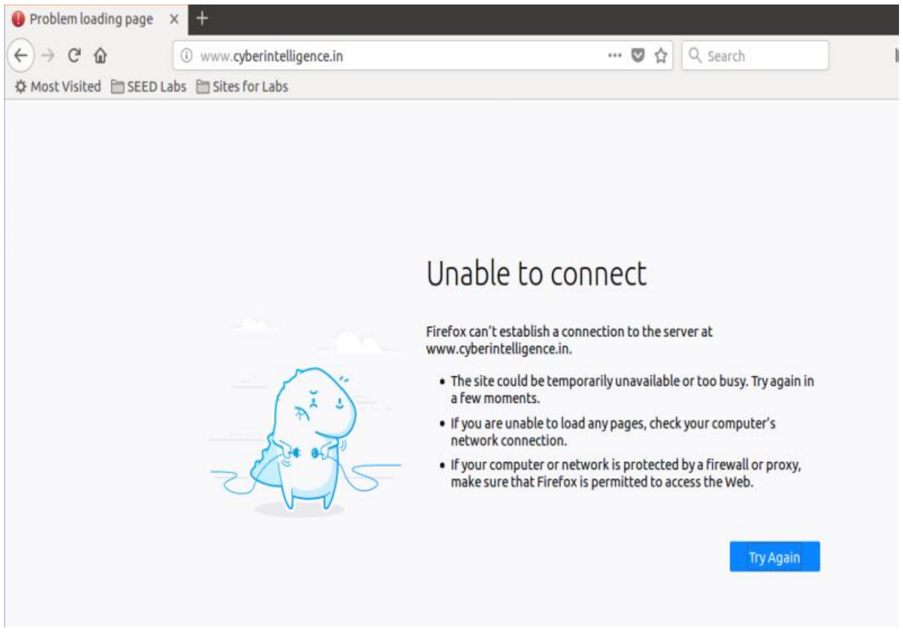
```
[08/01/21]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
```

Creating and enabling a rule to block all web traffic to the website, www.cyberintelligence.in



```
[08/03/21]seed@VM:~/CODE$ nslookup www.cyberintelligence.in
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
www.cyberintelligence.in        canonical name = cyberintelligence.in.
Name:   cyberintelligence.in
Address: 192.124.249.115

[08/03/21]seed@VM:~/CODE$ sudo ufw deny out from 10.0.2.4 to 192.124.249.115 por
t 80
Rule added
[08/03/21]seed@VM:~/CODE$ sudo ufw enable
Firewall is active and enabled on system startup
[08/03/21]seed@VM:~/CODE$ 
```

Showing that the site is unreachable:



Unable to connect

Firefox can't establish a connection to the server at www.cyberintelligence.in.

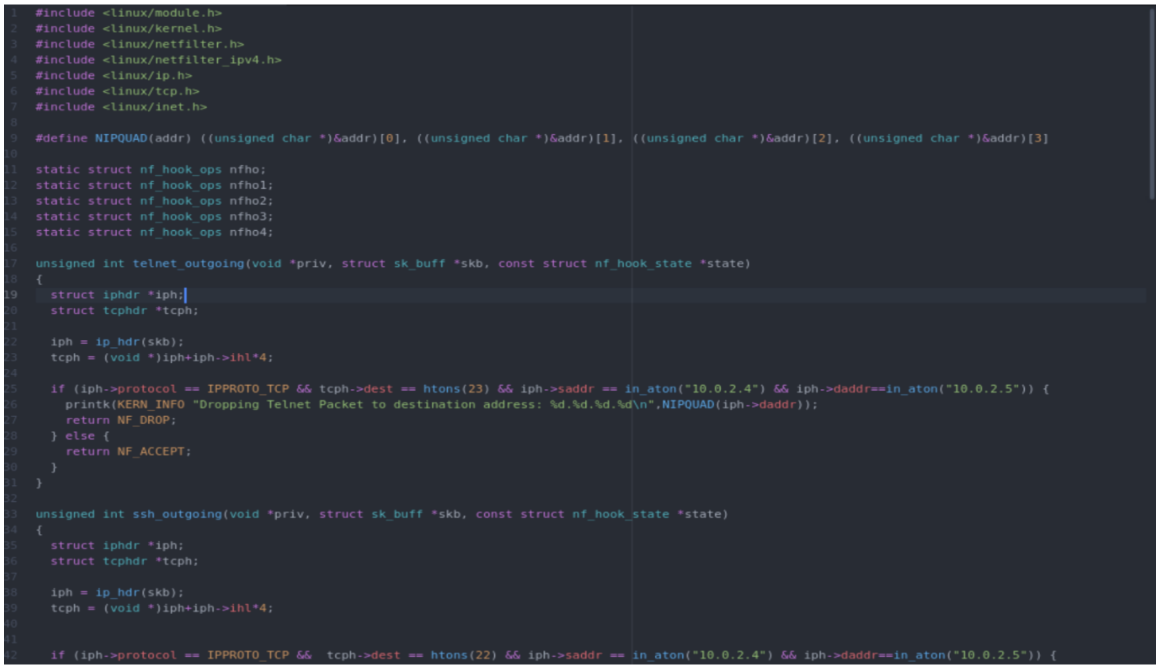
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Observation: The screenshots above demonstrate the use of the iptables and UFW program to create simple packet filtering firewall rules. We first blocked telnet traffic from Machine A to Machine B, then we blocked telnet traffic from Machine B to Machine A and finally we blocked port 80 (http) traffic from Machine A to the website www.cyberintelligence.in

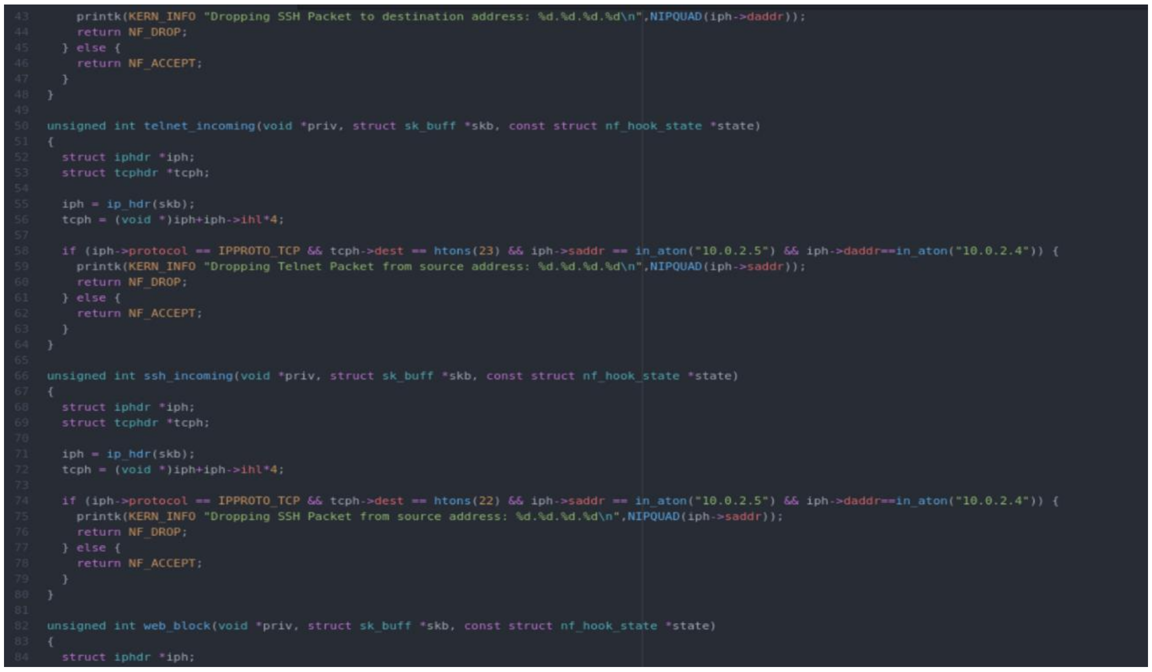
Explanation: Implementing simple packet filtering rules for iptables. Traffic can be blocked between specific hosts, networks or protocols. Root privileges are required to configure rules, enable the firewall or disable the firewall.

1. **Implementing a Simple Firewall**

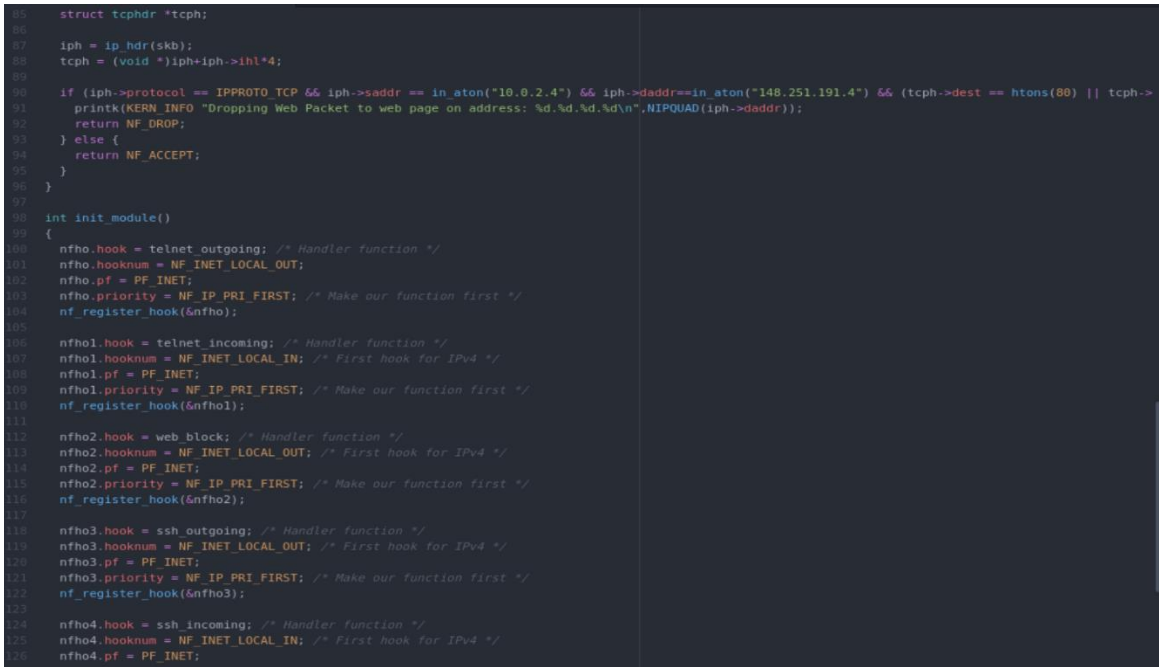
Code for the packet filter, based on the code in the textbook and the sample:



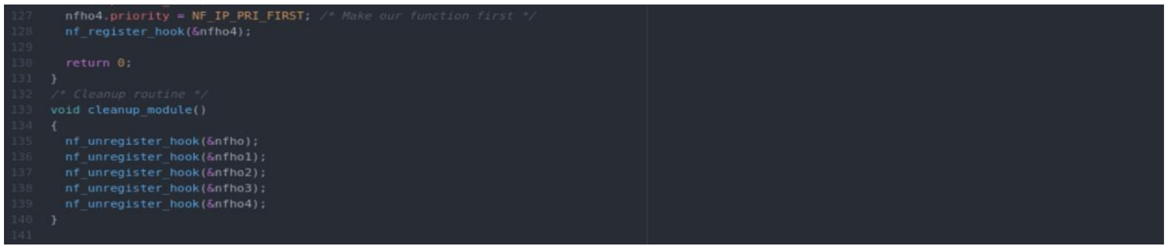
```c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/inet.h>

#define NIPQUAD(addr) ((unsigned char *)&addr)[0], ((unsigned char *)&addr)[1], ((unsigned char *)&addr)[2], ((unsigned char *)&addr)[3]

static struct nf_hook_ops nfho;
static struct nf_hook_ops nfho1;
static struct nf_hook_ops nfho2;
static struct nf_hook_ops nfho3;
static struct nf_hook_ops nfho4;

unsigned int telnet_outgoing(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
  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->saddr == in_aton("10.0.2.4") && iph->daddr==in_aton("10.0.2.5")) {
    printk(KERN_INFO "Dropping Telnet Packet to destination address: %d.%d.%d.%d\n",NIPQUAD(iph->daddr));
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}

unsigned int ssh_outgoing(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
  struct iphdr *iph;
  struct tcphdr *tcph;

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


  if (iph->protocol == IPPROTO_TCP &&  tcph->dest == htons(22) && iph->saddr == in_aton("10.0.2.4") && iph->daddr==in_aton("10.0.2.5")) {
```



```c
    printk(KERN_INFO "Dropping SSH Packet to destination address: %d.%d.%d.%d\n",NIPQUAD(iph->daddr));
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}

unsigned int telnet_incoming(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
  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->saddr == in_aton("10.0.2.5") && iph->daddr==in_aton("10.0.2.4")) {
    printk(KERN_INFO "Dropping Telnet Packet from source address: %d.%d.%d.%d\n",NIPQUAD(iph->saddr));
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}

unsigned int ssh_incoming(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
  struct iphdr *iph;
  struct tcphdr *tcph;

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

  if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(22) && iph->saddr == in_aton("10.0.2.5") && iph->daddr==in_aton("10.0.2.4")) {
    printk(KERN_INFO "Dropping SSH Packet from source address: %d.%d.%d.%d\n",NIPQUAD(iph->saddr));
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}

unsigned int web_block(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
  struct iphdr *iph;
```

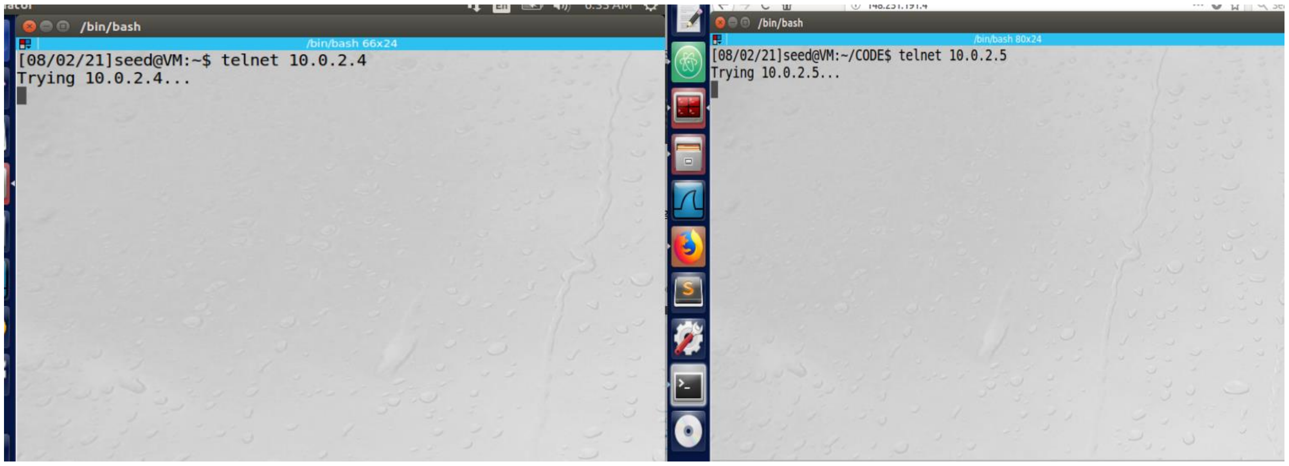


```c
  struct tcphdr *tcph;

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

  if (iph->protocol == IPPROTO_TCP && iph->saddr == in_aton("10.0.2.4") && iph->daddr==in_aton("148.251.191.4") && (tcph->dest == htons(80) || tcph->
    printk(KERN_INFO "Dropping Web Packet to web page on address: %d.%d.%d.%d\n",NIPQUAD(iph->daddr));
    return NF_DROP;
  } else {
    return NF_ACCEPT;
  }
}

int init_module()
{
  nfho.hook = telnet_outgoing; /* Handler function */
  nfho.hooknum = NF_INET_LOCAL_OUT;
  nfho.pf = PF_INET;
  nfho.priority = NF_IP_PRI_FIRST; /* Make our function first */
  nf_register_hook(&nfho);

  nfho1.hook = telnet_incoming; /* Handler function */
  nfho1.hooknum = NF_INET_LOCAL_IN; /* First hook for IPv4 */
  nfho1.pf = PF_INET;
  nfho1.priority = NF_IP_PRI_FIRST; /* Make our function first */
  nf_register_hook(&nfho1);

  nfho2.hook = web_block; /* Handler function */
  nfho2.hooknum = NF_INET_LOCAL_OUT; /* First hook for IPv4 */
  nfho2.pf = PF_INET;
  nfho2.priority = NF_IP_PRI_FIRST; /* Make our function first */
  nf_register_hook(&nfho2);

  nfho3.hook = ssh_outgoing; /* Handler function */
  nfho3.hooknum = NF_INET_LOCAL_OUT; /* First hook for IPv4 */
  nfho3.pf = PF_INET;
  nfho3.priority = NF_IP_PRI_FIRST; /* Make our function first */
  nf_register_hook(&nfho3);

  nfho4.hook = ssh_incoming; /* Handler function */
  nfho4.hooknum = NF_INET_LOCAL_IN; /* First hook for IPv4 */
  nfho4.pf = PF_INET;
```



```c
  nfho4.priority = NF_IP_PRI_FIRST; /* Make our function first */
  nf_register_hook(&nfho4);

  return 0;
}
/* Cleanup routine */
void cleanup_module()
{
  nf_unregister_hook(&nfho);
  nf_unregister_hook(&nfho1);
  nf_unregister_hook(&nfho2);
  nf_unregister_hook(&nfho3);
  nf_unregister_hook(&nfho4);
}
```

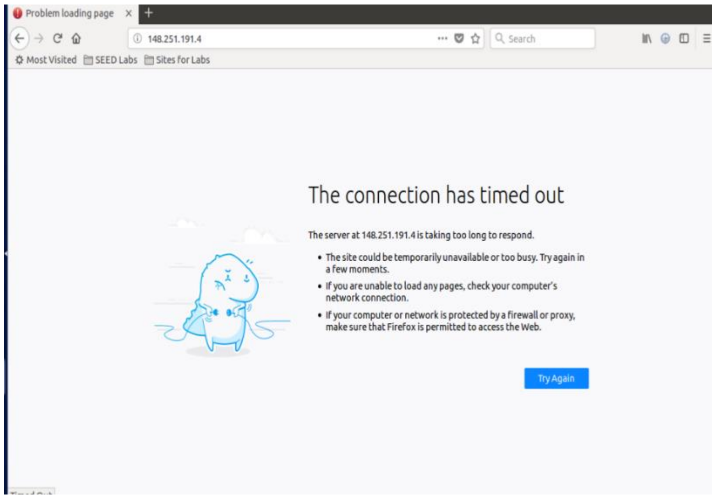
after implementing the filter, and trying to telnet again:



```
[08/02/21]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...
```

```
[08/02/21]seed@VM:~/CODE$ telnet 10.0.2.5
Trying 10.0.2.5...
```

Also, website no longer load:

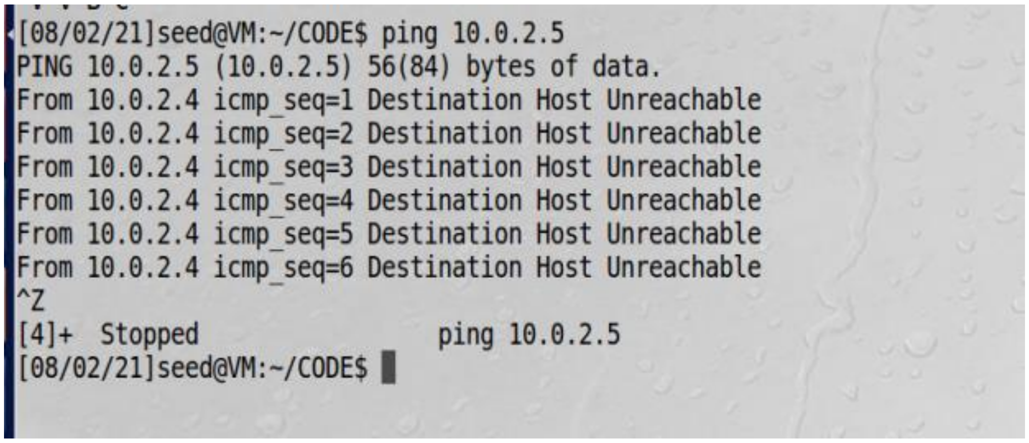


The connection has timed out

The server at 148.251.191.4 is taking too long to respond.

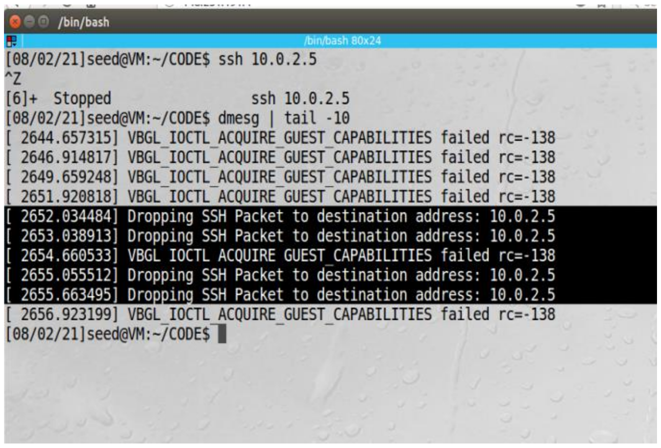
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Trying to ping from another machine:



```
[08/02/21]seed@VM:~/CODE$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
From 10.0.2.4 icmp_seq=1 Destination Host Unreachable
From 10.0.2.4 icmp_seq=2 Destination Host Unreachable
From 10.0.2.4 icmp_seq=3 Destination Host Unreachable
From 10.0.2.4 icmp_seq=4 Destination Host Unreachable
From 10.0.2.4 icmp_seq=5 Destination Host Unreachable
From 10.0.2.4 icmp_seq=6 Destination Host Unreachable
^Z
[4]+  Stopped                 ping 10.0.2.5
[08/02/21]seed@VM:~/CODE$
```

Trying ssh from VM A to VM B



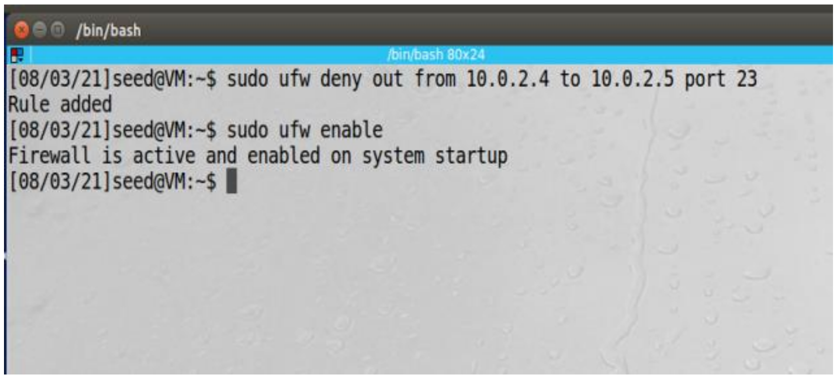
```
[08/02/21]seed@VM:~/CODE$ ssh 10.0.2.5
^Z
[6]+  Stopped                 ssh 10.0.2.5
[08/02/21]seed@VM:~/CODE$ dmesg | tail -10
[ 2644.657315] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 2646.914817] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 2649.659248] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 2651.920818] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 2652.034484] Dropping SSH Packet to destination address: 10.0.2.5
[ 2653.038913] Dropping SSH Packet to destination address: 10.0.2.5
[ 2654.660533] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 2655.055512] Dropping SSH Packet to destination address: 10.0.2.5
[ 2655.663495] Dropping SSH Packet to destination address: 10.0.2.5
[ 2656.923199] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[08/02/21]seed@VM:~/CODE$
```

Observation: The screenshots above demonstrate the use of the C program to create simple packet filtering firewall. Inside the telnet filter function, we inspect the packet, it if meets one of our blocking criteria we block it, if not we accept it. There are two functions that that will print out the IP address of the packet that was blocked.

Explanation: The NETFILTER application provides hooks for installing a packet-filter firewall. We created a firewall and implemented. We were able to inspect, and block packets based on different attributes, like destination IP, port number, or protocol used.

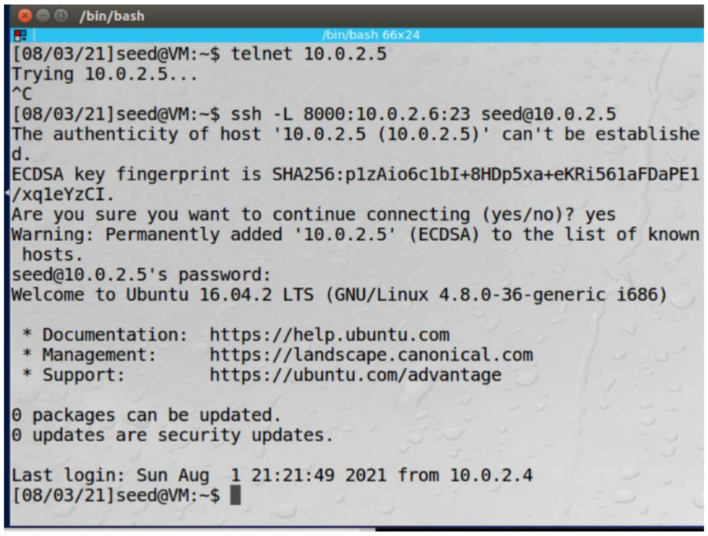
Evading Egress Filtering

3.a Telnet to Machine B through Firewall
firewall rules blocking telnet from VM A to VM B:



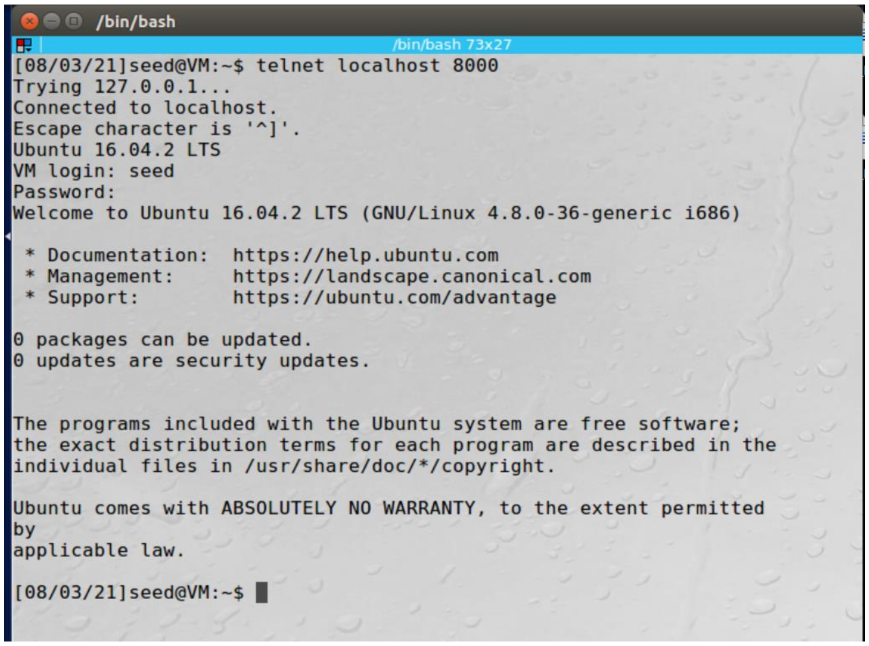
```
[08/03/21]seed@VM:~$ sudo ufw deny out from 10.0.2.4 to 10.0.2.5 port 23
Rule added
[08/03/21]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[08/03/21]seed@VM:~$ 
```

From machine A, building the SSH Tunnel between machine A and Machine B so that we can eventually TELNET into Machine C:



```
[08/03/21]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
^C
[08/03/21]seed@VM:~$ ssh -L 8000:10.0.2.6:23 seed@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be establishe
d.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1
/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.5' (ECDSA) to the list of known
 hosts.
seed@10.0.2.5'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

0 packages can be updated.
0 updates are security updates.

Last login: Sun Aug  1 21:21:49 2021 from 10.0.2.4
[08/03/21]seed@VM:~$ 
```

Opening another Terminal window on Machine A, we can now telnet into machine C by typing the command: telnet localhost 8000:

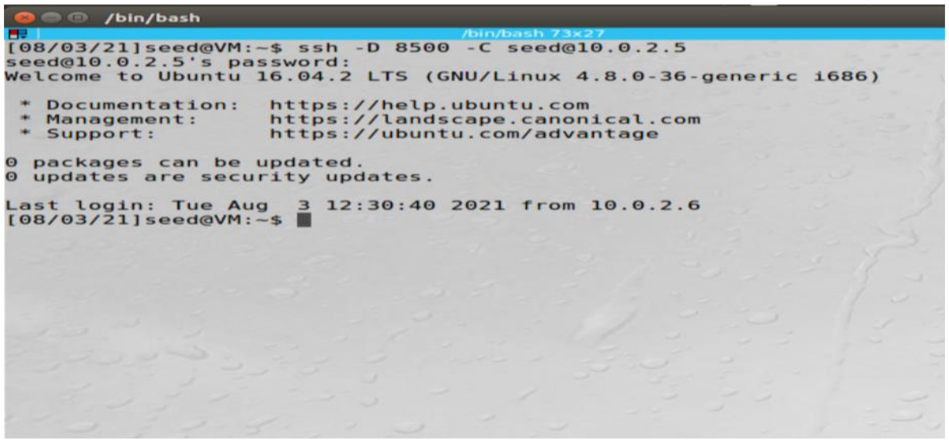


```
[08/03/21]seed@VM:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted
by
applicable law.

[08/03/21]seed@VM:~$ 
```

Observation: In this task the goal was to TELNET from Machine A to Machine B but there is a firewall on machine A that was blocking all TELNET traffic. To get around the firewall, we established an SSH tunnel from Machine A to Machine C, and then once the connection was established, we could TELNET into machine B. This essentially connects Machine A to Machine B via TELNET but is not blocked by the firewall due to the SSH tunnel.

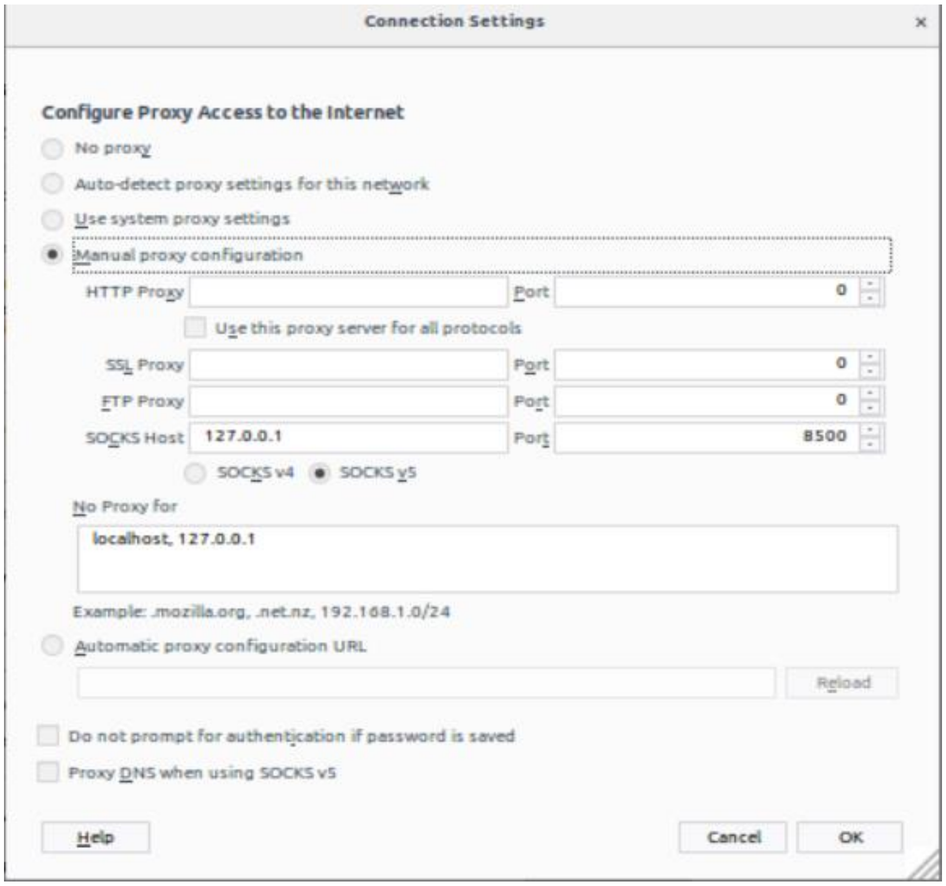
Explanation: On Machine A, the tunnel receives TCP packets from the telnet client (when we do telnet localhost 8000).
On receiving this packet, the tunnel forwards the TCP packets to Machine B – port 23. Here, the received
data is put into another TCP packet and sent to Machine B’s port 23 (as mentioned in the SSH connection). The firewall only sees the SSH traffic and not the telnet traffic, and hence this SSH tunnel
can be used to evade the firewall rule of blocking telnet connections.

3.b Connect to Facebook using SSH Tunnel
First, we connect Machine A to Machine B via an SSH tunnel:



```
[08/03/21]seed@VM:~$ ssh -D 8500 -C seed@10.0.2.5
seed@10.0.2.5'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

0 packages can be updated.
0 updates are security updates.

Last login: Tue Aug  3 12:30:40 2021 from 10.0.2.6
[08/03/21]seed@VM:~$ 
```

Then we adjust the Firefox browser settings to connect to localhost port 8500 (now Machine B) when connecting to a web server:


Connection Settings

Configure Proxy Access to the Internet

No proxy
Auto-detect proxy settings for this network
Use system proxy settings
[✓ Manual proxy configuration]

HTTP Proxy | Port 0
Use this proxy server for all protocols
SSL Proxy | Port 0
FTP Proxy | Port 0
SOCKS Host 127.0.0.1 | Port 8500
SOCKS v4 [✓ SOCKS v5]

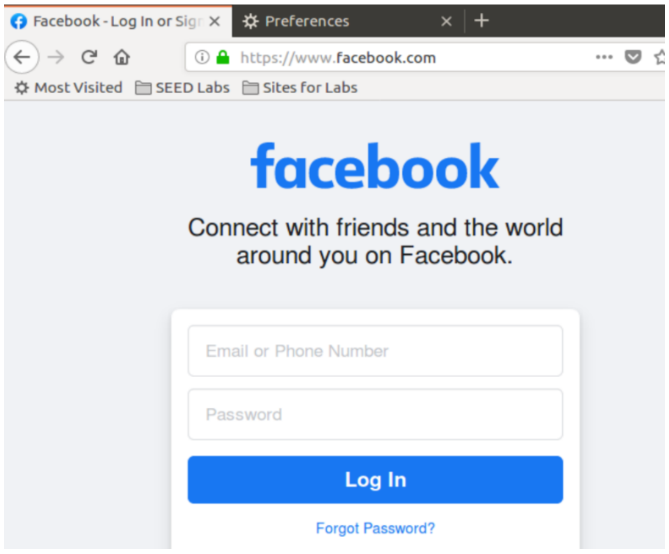
No Proxy for
localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

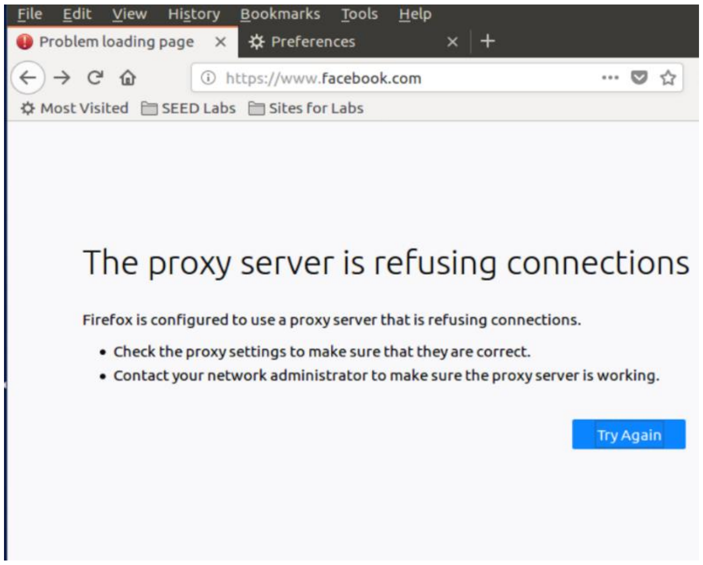
Automatic proxy configuration URL

Do not prompt for authentication if password is saved
Proxy DNS when using SOCKS v5

Now we can visit Facebook on Machine A using our SSH tunnel:



Next we break the SSH tunnel, clear the browser cache, and try to visit Facebook again, we receive an error from Firefox that the proxy server is refusing connections:

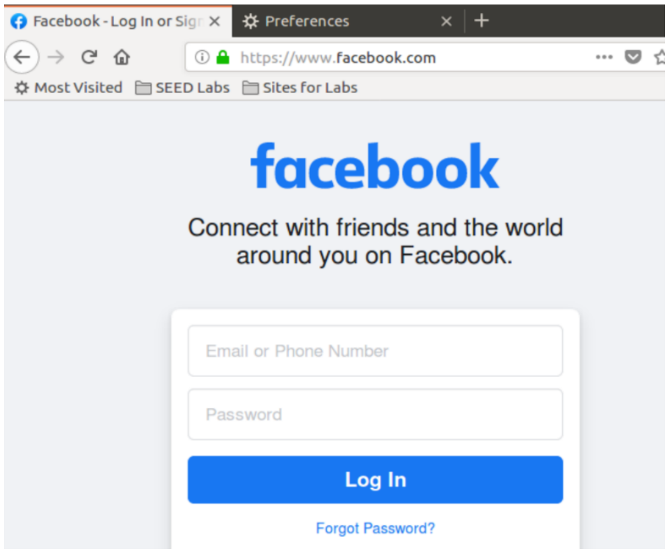


The proxy server is refusing connections

Firefox is configured to use a proxy server that is refusing connections.

- Check the proxy settings to make sure that they are correct.
- Contact your network administrator to make sure the proxy server is working.

Upon reestablishing the SSH tunnel we can visit Facebook again:



Connect with friends and the world around you on Facebook.

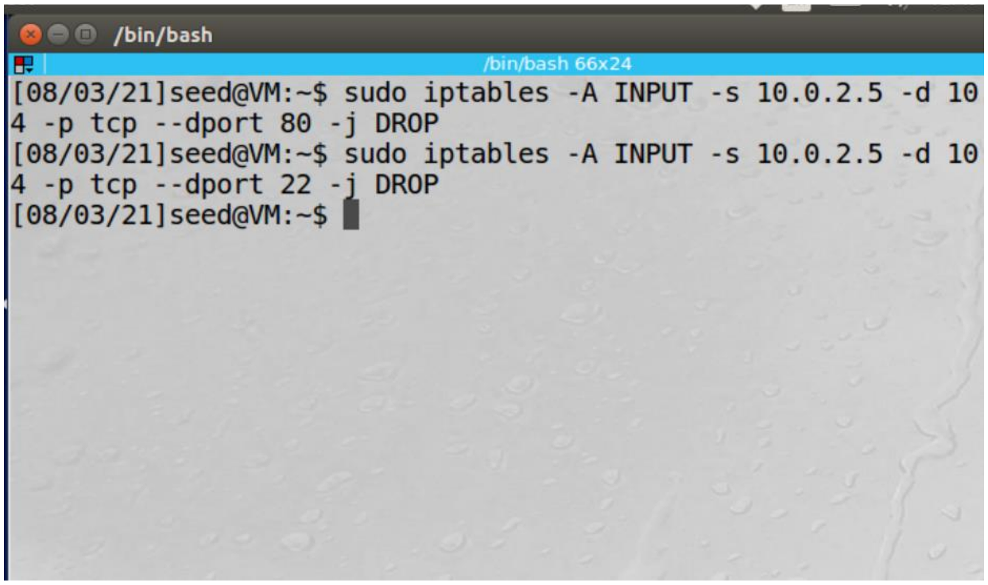
Observation: The goal of this task was to bypass the firewall which is blocking Machine A's access to the Facebook website. We established an SSH tunnel from Machine A and forward all traffic from localhost port 8500 to Machine B. Next we can adjust our Firefox settings to route all web traffic to localhost port 8500, which is then forwarded to Machine B. Now we can access Facebook on Machine A even though there is a rule in the firewall trying to prevent it.

Explanation: **I**n this task, the browser connects to the SSH proxy at port 8500 on the localhost, and the SSH sends the
TCP data over the tunnel to Machine B, which connects to the blocked website (based on the destination).
This SSH tunnel responds back via the same tunnel and since all the traffic is SSH and not web

traffic, the
firewall does not block anything.

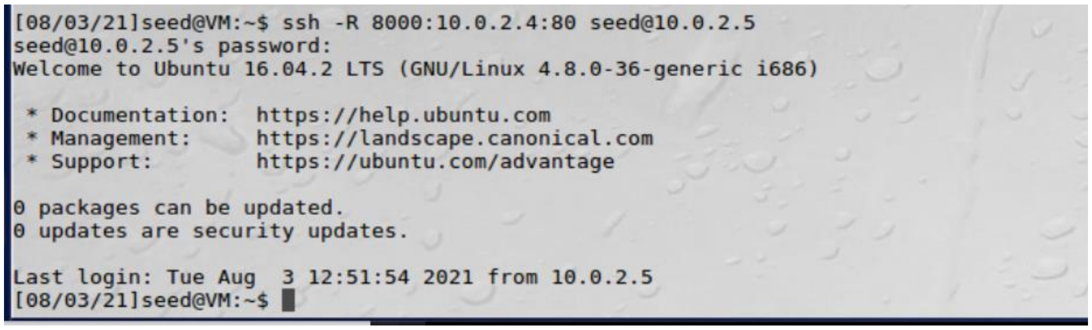
Evading Ingress Filtering
On Machine A, we run the web server which is protected by blocking any incoming HTTP/SSH traffic
from the external network (here just Machine B). This Machine can also be configured to block all the
SSH traffic from anywhere. The results will be similar.

firewall rules configured on Machine A:



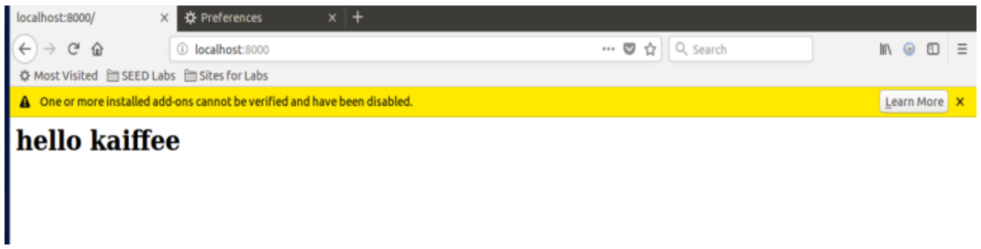
```
[08/03/21]seed@VM:~$ sudo iptables -A INPUT -s 10.0.2.5 -d 10
4 -p tcp --dport 80 -j DROP
[08/03/21]seed@VM:~$ sudo iptables -A INPUT -s 10.0.2.5 -d 10
4 -p tcp --dport 22 -j DROP
[08/03/21]seed@VM:~$
```

On trying to access this web server from the outside, we see that it is not possible since the http traffic is
blocked. We also cannot use the previous SSH tunnel mechanism for port forwarding that would have provided us with the access to the web server. Since the Machine blocks only incoming SSH tunnel, we set up a reverse SSH tunnel on Machine A which is not blocked by the firewall. This SSH tunnel will be used to access the protected web server.



```
[08/03/21]seed@VM:~$ ssh -R 8000:10.0.2.4:80 seed@10.0.2.5
seed@10.0.2.5'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

0 packages can be updated.
0 updates are security updates.

Last login: Tue Aug  3 12:51:54 2021 from 10.0.2.5
[08/03/21]seed@VM:~$
```

After establishing the above SSH tunnel, we can access the web server from the outside (Machine B – 10.0.2.5) by simply going to localhost:8000 on the external Machine. This is because the established SSH tunnel forwards the request to SSH client on Machine A, which further forwards the request to port 80 of the Machine A – 10.0.2.4 i.e. the web server. The following shows that we can successfully access the web server from the outside:



One or more installed add-ons cannot be verified and have been disabled.

hello kaiffee