Name: Kwa Li Ying
Student ID: 1003833

50.020 Network Security Lab 5: Linux Firewall Exploration

Task 1: Using Firewall

IP Address of Machine A: 10.0.2.128

IP Address of Machine B: 10.0.2.129

Prevent A from doing telnet to Machine B

By default, Machine A is able to telnet Machine B:

```
[03/15/21]seed@VM:~/.../lab5$ telnet 10.0.2.129
Trying 10.0.2.129...
Connected to 10.0.2.129.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Mar 15 10:24:24 EDT 2021 from www.youtube.com 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.
```

Bash commands as shown in part1.sh are executed on Machine A to implement policies in iptables:

```
#!/bin/bash

# Prevent telnet to Machine B
sudo iptables -A OUTPUT -d 10.0.2.129 -p tcp --dport 23 -j DROP

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

The 1st command blocks outgoing packets that are destined for B's IP address and port number of 23 (which indicate that they are telnet packets). The 2nd, 3rd and 4th commands specify the default policy as ACCEPT, so that the additional rules declared in the filter table are for packets to be dropped or rejected.

To check that the policies are added properly, we run the following command:

```
[03/15/21]seed@VM:~/.../lab5$ 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             10.0.2.129           tcp dpt:telnet
```

Name: Kwa Li Ying
Student ID: 1003833

Now, Machine A is NOT able to telnet Machine B:

```
[03/15/21]seed@VM:~/.../lab5$ telnet 10.0.2.129
Trying 10.0.2.129...
```

Before moving on, the firewall is set back to its default state by running flush.sh so as to not affect the next part of the lab:

```
#!/bin/bash

# Clear all rules
sudo iptables -F

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

Name: Kwa Li Ying
Student ID: 1003833

Prevent B from doing telnet to Machine A

By default, Machine A is able to telnet Machine B:

```
[03/15/21]seed@VM:~$ telnet 10.0.2.128
Trying 10.0.2.128...
Connected to 10.0.2.128.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  6 12:47:43 EST 2021 from 10.0.2.129 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.
```

Bash commands as shown in part2.sh are executed on Machine B to implement policies in iptables:

```
#!/bin/bash

# Prevent telnet to Machine B
sudo iptables -A OUTPUT -d 10.0.2.128 -p tcp --dport 23 -j DROP

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

The 1st command blocks outgoing packets that are destined for A's IP address and port number of 23 (which indicate that they are telnet packets). The 2nd, 3rd and 4th commands specify the default policy as ACCEPT, so that the additional rules declared in the filter table are for packets to be dropped or rejected.

To check that the policies are added properly, we run the following command:

```
[03/15/21]seed@VM:~/.../lab5$ 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             10.0.2.128           tcp dpt:telnet
```

Now, Machine B is NOT able to telnet Machine A:

```
[03/15/21]seed@VM:~/.../lab5$ telnet 10.0.2.128
Trying 10.0.2.128...
```

Name: Kwa Li Ying
Student ID: 1003833

Before moving on, the firewall is set back to its default state by running flush.sh so as to not affect the next part of the lab:

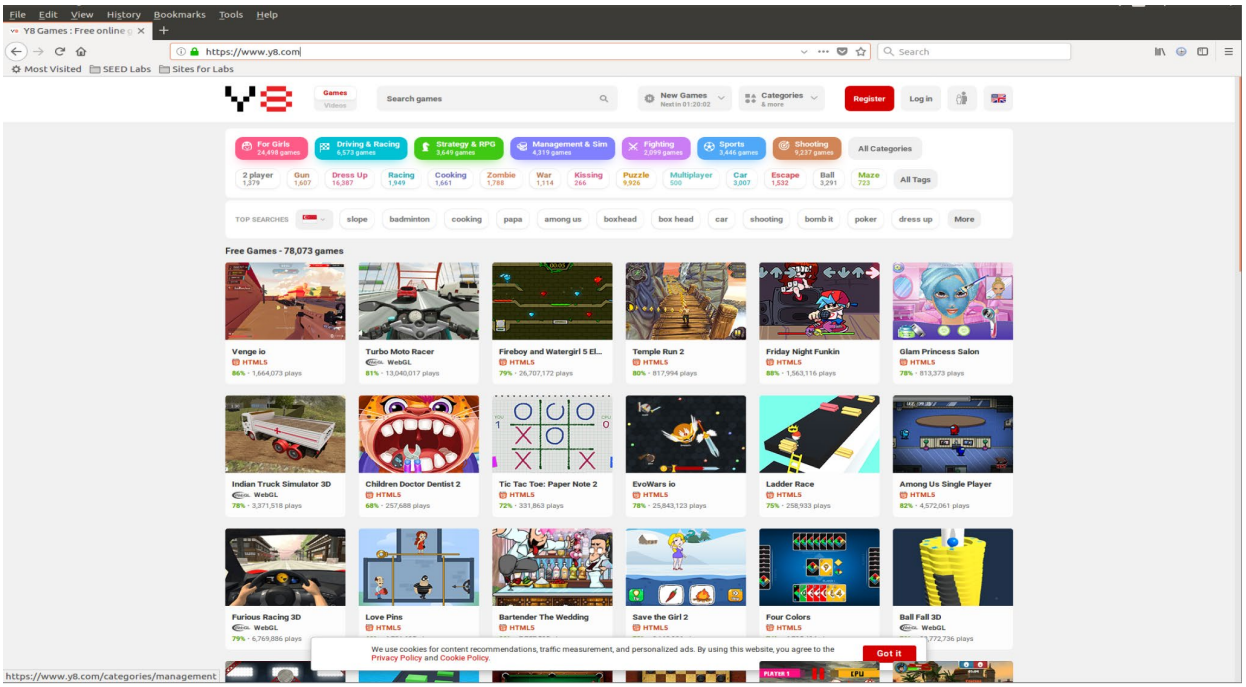
```
#!/bin/bash

# Clear all rules
sudo iptables -F

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

Name: Kwa Li Ying
Student ID: 1003833

Prevent A from visiting an external website

Website chosen: www.y8.com

By default, Machine A can visit www.y8.com:



The dig command is used to resolve the hostname to its IP address(es):

```
[03/15/21]seed@VM:~/.../task1$ dig www.y8.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.y8.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37928
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.y8.com.                    IN      A

;; ANSWER SECTION:
www.y8.com.             207     IN      CNAME   y8.com.
y8.com.                 207     IN      A       159.203.188.92
y8.com.                 207     IN      A       162.243.169.120
y8.com.                 207     IN      A       159.203.184.51
y8.com.                 207     IN      A       67.205.142.78

;; AUTHORITY SECTION:
y8.com.                 172706  IN      NS      ns2.y8.com.
y8.com.                 172706  IN      NS      ns1.y8.com.

;; ADDITIONAL SECTION:
ns1.y8.com.             172706  IN      A       69.55.48.167
ns2.y8.com.             172706  IN      A       141.0.173.138

;; Query time: 0 msec
;; SERVER: 10.0.2.129#53(10.0.2.129)
;; WHEN: Mon Mar 15 10:48:03 EDT 2021
;; MSG SIZE  rcvd: 185
```

Name: Kwa Li Ying
Student ID: 1003833

Bash commands as shown in part3.sh are executed on Machine B to implement policies in iptables:

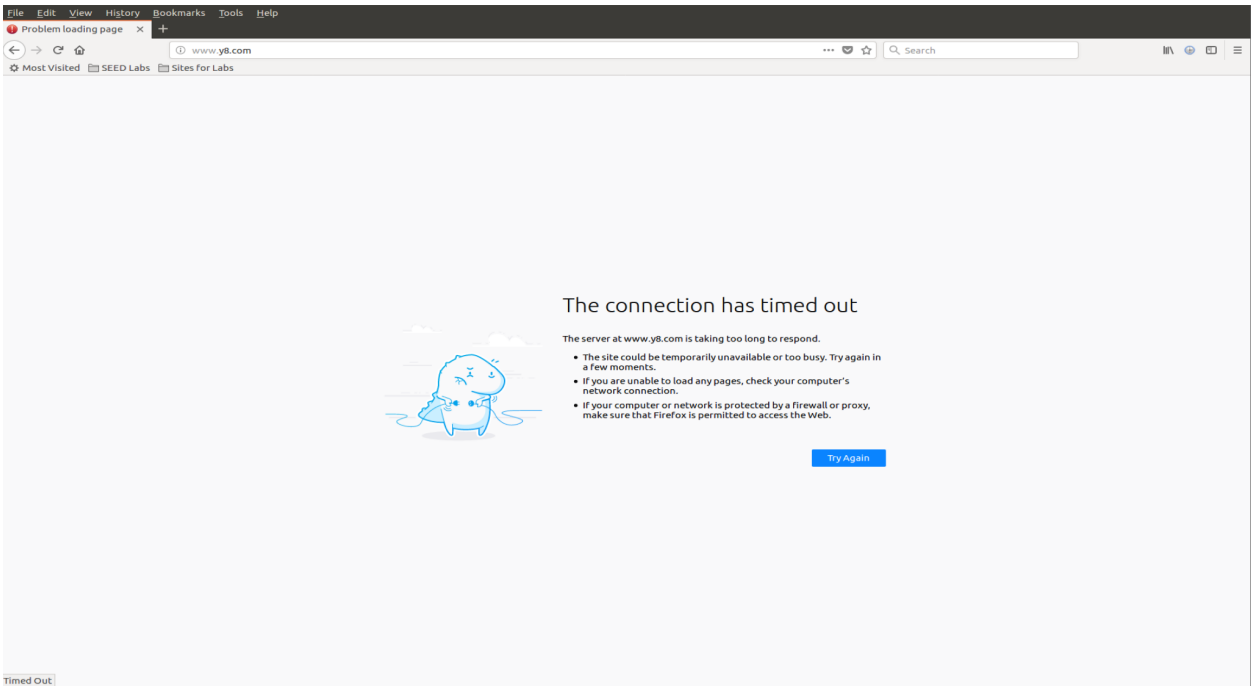
```
#!/bin/bash

# Prevent telnet to Machine B
sudo iptables -A OUTPUT -d 159.203.188.92 -p tcp -j DROP
sudo iptables -A OUTPUT -d 162.243.169.120 -p tcp -j DROP
sudo iptables -A OUTPUT -d 159.203.184.51 -p tcp -j DROP
sudo iptables -A OUTPUT -d 67.205.142.78 -p tcp -j DROP

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

The first 4 commands block outgoing packets that are destined for www.y8.com's IP addresses. The remaining commands specify the default policy as ACCEPT, so that the additional rules declared in the filter table are for packets to be dropped or rejected.

To check that the policies are added properly, we run the following command:

```
[03/15/21]seed@VM:~/.../task1$ 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             fw3.wel.nbg
DROP       tcp  --  anywhere             fw1.wel.nbg
DROP       tcp  --  anywhere             fw2.wel.nbg
DROP       tcp  --  anywhere             fw4.wel.nbg
```

Now, Machine A is NOT able to visit www.y8.com as the page can never load:



Name: Kwa Li Ying
Student ID: 1003833

Before moving on, the firewall is set back to its default state by running flush.sh so as to not affect the next part of the lab:

```
#!/bin/bash

# Clear all rules
sudo iptables -F

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

Name: Kwa Li Ying
Student ID: 1003833

Task 2: Implementing a Simple Firewall

Rule 1: Drop all packets

Refer to the C code written in rule1.c. The code is written to drop all incoming packets.

The Makefile is written as shown:

```
obj-m += rule1.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

The make command is executed and the various library files are created in the current directory:

```
[03/15/21]seed@VM:~/.../task2$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/ns/lab5/task2
 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Documents/ns/lab5/task2/rule1.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Documents/ns/lab5/task2/rule1.mod.o
  LD [M]  /home/seed/Documents/ns/lab5/task2/rule1.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[03/15/21]seed@VM:~/.../task2$ ls
Makefile       Module.symvers  rule1.ko     rule1.mod.o
modules.order  rule1.c         rule1.mod.c  rule1.o
```

Before the kernel module is inserted, Machine B can ping Machine A:

```
[03/15/21]seed@VM:~$ ping 10.0.2.128
PING 10.0.2.128 (10.0.2.128) 56(84) bytes of data.
64 bytes from 10.0.2.128: icmp_seq=1 ttl=64 time=0.085 ms
64 bytes from 10.0.2.128: icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from 10.0.2.128: icmp_seq=3 ttl=64 time=0.075 ms
^C
--- 10.0.2.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.069/0.076/0.085/0.009 ms
```

The module is then inserted into the kernel on Machine A:

```
[03/15/21]seed@VM:~/.../task2$ sudo insmod rule1.ko
[03/15/21]seed@VM:~/.../task2$ lsmod | grep rule1
rule1                  16384  0
```

Now, Machine B is unable to ping Machine A:

```
[03/15/21]seed@VM:~$ ping 10.0.2.128
PING 10.0.2.128 (10.0.2.128) 56(84) bytes of data.
^C
--- 10.0.2.128 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4098ms
```

Name: Kwa Li Ying
Student ID: 1003833

To clean up, the following command is run to remove the module from the kernel:

```
[03/15/21]seed@VM:~/.../task2$ sudo rmmod rule1
[03/15/21]seed@VM:~/.../task2$ lsmod | grep rule1
```

Name: Kwa Li Ying
Student ID: 1003833

Rule 2: Prevent A from doing telnet to Machine B

To achieve this, on Machine A, all packets that are outgoing packets (not including forwarded packets) that are telnet packets destined for Machine B should be dropped. Refer to the C code written in rule2.c.

The Makefile is written as shown:

```
obj-m += rule2.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

The make command is executed and the various library files are created in the current directory:

```
[03/15/21]seed@VM:~/.../rule2$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/ns/lab5/task2
/rule2 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Documents/ns/lab5/task2/rule2/rule2.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Documents/ns/lab5/task2/rule2/rule2.mod.o
  LD [M]  /home/seed/Documents/ns/lab5/task2/rule2/rule2.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
```

Before the kernel module is inserted, Machine A can telnet Machine B:

```
[03/15/21]seed@VM:~/.../rule2$ telnet 10.0.2.129
Trying 10.0.2.129...
Connected to 10.0.2.129.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Mar 15 21:04:16 EDT 2021 from 10.0.2.129 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.
```

The module is then inserted into the kernel on Machine A:

```
[03/15/21]seed@VM:~/.../rule2$ sudo insmod rule2.ko
[03/15/21]seed@VM:~/.../rule2$ lsmod | grep rule2
rule2                  16384  0
```

Now, Machine A is unable to telnet Machine B:

Name: Kwa Li Ying
Student ID: 1003833

```
[03/15/21]seed@VM:~/.../rule2$ telnet 10.0.2.129
Trying 10.0.2.129...
```

To clean up, the following command is run to remove the module from the kernel:

```
[03/15/21]seed@VM:~/.../rule2$ sudo rmmod rule2
[03/15/21]seed@VM:~/.../rule2$ lsmod | grep rule2
```

Name: Kwa Li Ying
Student ID: 1003833

Rule 3: Prevent B from doing telnet to Machine A

To achieve this, on Machine B, all packets that are outgoing packets (not including forwarded packets) that are telnet packets destined for Machine A should be dropped. Refer to the C code written in rule3.c.

The Makefile is written as shown:

```
obj-m += rule3.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

The make command is executed and the various library files are created in the current directory:

```
[03/15/21]seed@VM:~/.../rule3$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/ns/lab5/task2
/rule3 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Documents/ns/lab5/task2/rule3/rule3.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Documents/ns/lab5/task2/rule3/rule3.mod.o
  LD [M]  /home/seed/Documents/ns/lab5/task2/rule3/rule3.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[03/15/21]seed@VM:~/.../rule3$ ls
Makefile       Module.symvers  rule3.ko     rule3.mod.o
modules.order  rule3.c         rule3.mod.c  rule3.o
```

Before the kernel module is inserted, Machine B can telnet Machine A:

```
[03/15/21]seed@VM:~$ telnet 10.0.2.128
Trying 10.0.2.128...
Connected to 10.0.2.128.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Mar 15 21:06:48 EDT 2021 from 10.0.2.129 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

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

The module is then inserted into the kernel on Machine B:

```
[03/15/21]seed@VM:~/.../rule3$ sudo insmod rule3.ko
[03/15/21]seed@VM:~/.../rule3$ lsmod | grep rule3
rule3                  16384  0
```

Name: Kwa Li Ying
Student ID: 1003833

Now, Machine B is unable to telnet Machine A:

```
[03/15/21]seed@VM:~/.../rule3$ telnet 10.0.2.128
Trying 10.0.2.128...
```

To clean up, the following command is run to remove the module from the kernel:

```
[03/15/21]seed@VM:~/.../rule3$ sudo rmmod rule3
[03/15/21]seed@VM:~/.../rule3$ lsmod | grep rule3
```

Name: Kwa Li Ying
Student ID: 1003833

Rule 4: Prevent A from visiting an external web site

Website chosen: www.y8.com

From previous task, dig is used to get the IP address of the website:

- 67.205.142.78
- 159.203.188.92
- 159.203.184.51
- 162.243.169.120

To achieve this, on Machine A, all packets that are outgoing packets (not including forwarded packets) destined for these IP addresses should be dropped. Refer to the C code written in rule4.c.

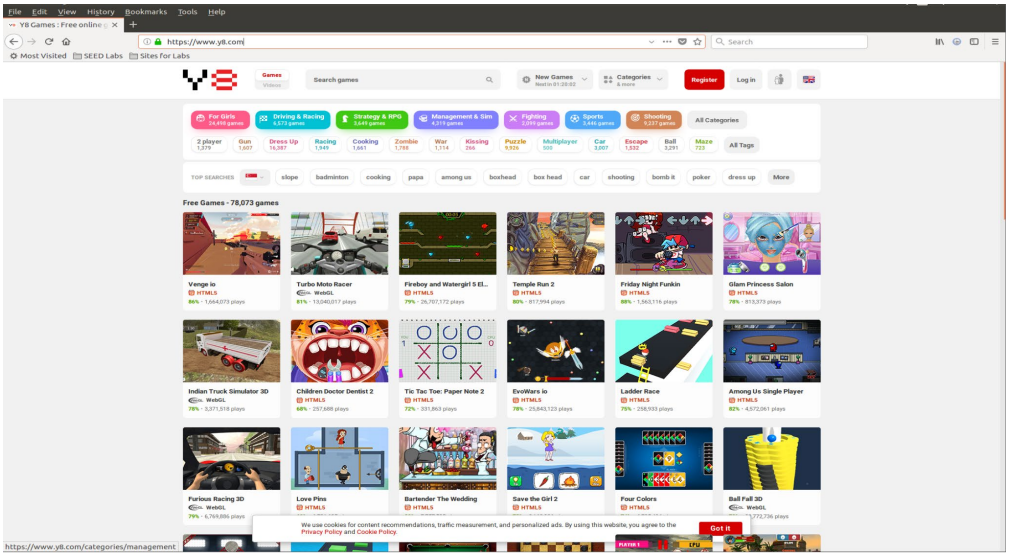
The Makefile is written as shown:

```
obj-m += rule4.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

The make command is executed and the various library files are created in the current directory:

```
[03/15/21]seed@VM:~/.../rule4$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/ns/lab5/task2
/rule4 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Documents/ns/lab5/task2/rule4/rule4.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Documents/ns/lab5/task2/rule4/rule4.mod.o
  LD [M]  /home/seed/Documents/ns/lab5/task2/rule4/rule4.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[03/15/21]seed@VM:~/.../rule4$ ls
Makefile       Module.symvers  rule4.ko      rule4.mod.o
modules.order  rule4.c         rule4.mod.c   rule4.o
```

Before the kernel module is inserted, Machine A can visit www.y8.com:

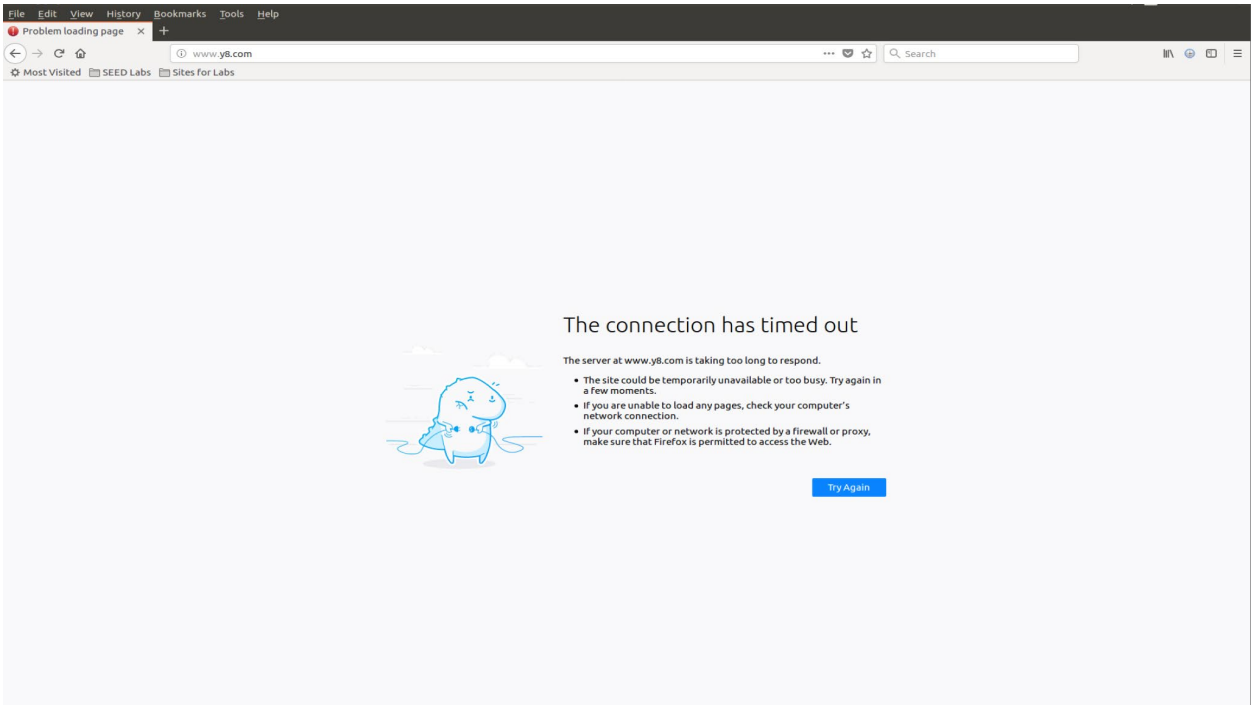


Name: Kwa Li Ying
Student ID: 1003833

The module is then inserted into the kernel on Machine A:

```
[03/15/21]seed@VM:~/.../rule4$ sudo insmod rule4.ko
[03/15/21]seed@VM:~/.../rule4$ lsmod | grep rule4
rule4                  16384  0
```

Now, Machine A is unable to visit www.y8.com:



To clean up, the following command is run to remove the module from the kernel:

```
[03/15/21]seed@VM:~/.../rule4$ sudo rmmod rule4
[03/15/21]seed@VM:~/.../rule4$ lsmod | grep rule4
```

Name: Kwa Li Ying
Student ID: 1003833

Rule 5: Prevent A from getting any telnet connections

To achieve this, on Machine A, all incoming packets that are telnet packets destined for Machine A should be dropped. Refer to the C code written in rule5.c.

The Makefile is written as shown:

```
obj-m += rule5.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

The make command is executed and the various library files are created in the current directory:

```
[03/15/21]seed@VM:~/.../rule5$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/ns/lab5/task2
/rule5 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Documents/ns/lab5/task2/rule5/rule5.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Documents/ns/lab5/task2/rule5/rule5.mod.o
  LD [M]  /home/seed/Documents/ns/lab5/task2/rule5/rule5.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[03/15/21]seed@VM:~/.../rule5$ ls
Makefile       Module.symvers  rule5.ko      rule5.mod.o
modules.order  rule5.c         rule5.mod.c   rule5.o
```

Before the kernel module is inserted, Machine B can telnet Machine A:

```
[03/15/21]seed@VM:~/.../rule3$ telnet 10.0.2.128
Trying 10.0.2.128...
Connected to 10.0.2.128.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Mar 15 21:14:10 EDT 2021 from 10.0.2.129 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

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

…and Machine C can telnet Machine A:

```
[03/15/21]seed@VM:~$ telnet 10.0.2.128
Trying 10.0.2.128...
Connected to 10.0.2.128.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Mar 15 21:42:59 EDT 2021 from 10.0.2.129 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.
```

Name: Kwa Li Ying
Student ID: 1003833

The module is then inserted into the kernel on Machine A:

```
[03/15/21]seed@VM:~/.../rule5$ sudo insmod rule5.ko
[03/15/21]seed@VM:~/.../rule5$ lsmod | grep rule5
rule5                  16384  0
```

Now, Machine B is unable to telnet Machine A:

```
[03/15/21]seed@VM:~/.../rule3$ telnet 10.0.2.128
Trying 10.0.2.128...
```

…and Machine C is unable to telnet Machine A:

```
[03/15/21]seed@VM:~$ telnet 10.0.2.128
Trying 10.0.2.128...
```

To clean up, the following command is run to remove the module from the kernel:

```
[03/15/21]seed@VM:~/.../rule5$ sudo rmmod rule5
[03/15/21]seed@VM:~/.../rule5$ lsmod | grep rule5
```

Name: Kwa Li Ying
Student ID: 1003833

Task 3: Evading Egress Filtering

IP Address of Machine A: 10.0.2.128

IP Address of Machine B: 10.0.2.129

IP Address of Machine C: 10.0.2.130

Block all the outgoing traffic to external telnet servers

Before the Firewall is set up, Machine A is able to telnet Machine B:

```
[03/15/21]seed@VM:~/.../task1$ telnet 10.0.2.128
Trying 10.0.2.128...
Connected to 10.0.2.128.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Mar 15 10:36:21 EDT 2021 from 10.0.2.129 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.
```

…as well as Machine C:

```
[03/15/21]seed@VM:~/.../task1$ telnet 10.0.2.130
Trying 10.0.2.130...
Connected to 10.0.2.130.
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

1 package 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.
```

Bash commands as shown in block_telnet.sh are executed on Machine A to implement policies in iptables:

```
#!/bin/bash

# Prevent telnet to Machine B
sudo iptables -A OUTPUT -p tcp --dport 23 -j DROP

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

Name: Kwa Li Ying
Student ID: 1003833

The first command blocks outgoing telnet packets. The remaining commands specify the default policy as ACCEPT, so that the additional rules declared in the filter table are for packets to be dropped or rejected.

To check that the policies are added properly, we run the following command:

```
[03/15/21]seed@VM:~/.../task3$ 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
```

Now, Machine A is NOT able to telnet Machine B:

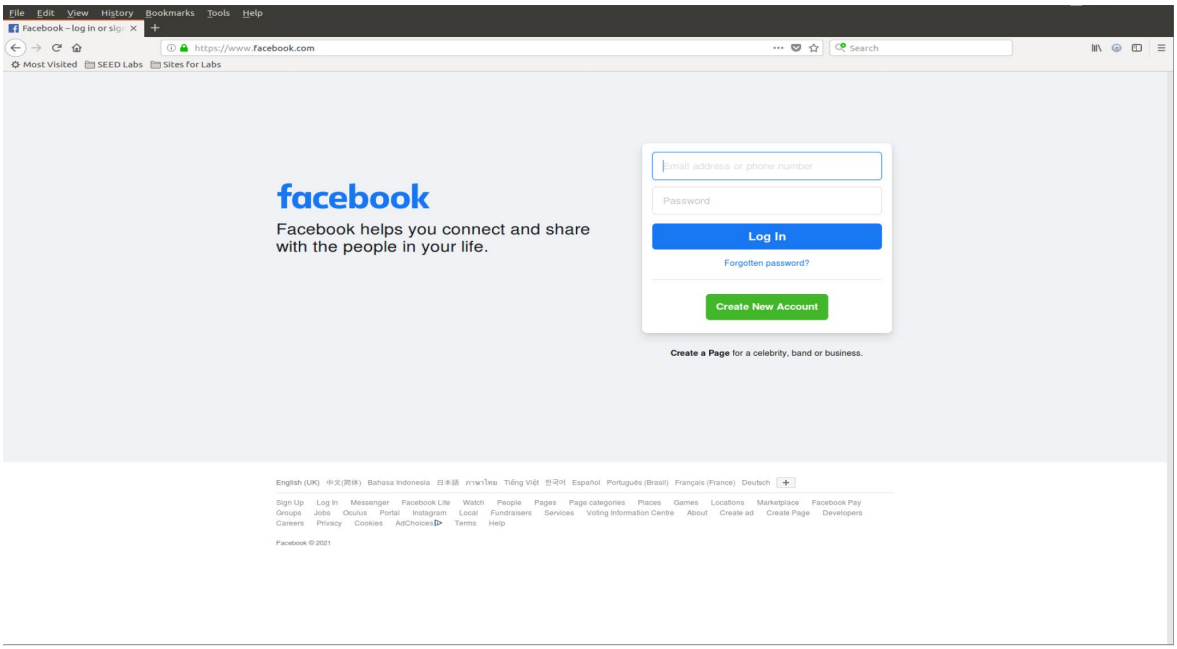
```
[03/15/21]seed@VM:~/.../task3$ telnet 10.0.2.129
Trying 10.0.2.129...
```

…as well as Machine C:

```
[03/15/21]seed@VM:~/.../task3$ telnet 10.0.2.130
Trying 10.0.2.130...
```

<u>Block all the outgoing traffic to www.facebook.com</u>

Before the Firewall is set up, Machine A is able to visit www.facebook.com:



Name: Kwa Li Ying
Student ID: 1003833

The dig command is used to resolve the www.facebook.com to its IP address(es):

```
[03/15/21]seed@VM:~/.../task3$ dig www.facebook.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42687
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 9

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

;; ANSWER SECTION:
www.facebook.com.       1411    IN      CNAME   star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com. 37 IN      A       157.240.7.35

;; AUTHORITY SECTION:
c10r.facebook.com.      1411    IN      NS      d.ns.c10r.facebook.com.
c10r.facebook.com.      1411    IN      NS      b.ns.c10r.facebook.com.
c10r.facebook.com.      1411    IN      NS      c.ns.c10r.facebook.com.
c10r.facebook.com.      1411    IN      NS      a.ns.c10r.facebook.com.

;; ADDITIONAL SECTION:
a.ns.c10r.facebook.com. 1411    IN      A       129.134.30.11
a.ns.c10r.facebook.com. 1411    IN      AAAA    2a03:2880:f0fc:b:face:b00c:0:99
b.ns.c10r.facebook.com. 1411    IN      A       129.134.31.11
b.ns.c10r.facebook.com. 1411    IN      AAAA    2a03:2880:f0fd:b:face:b00c:0:99
c.ns.c10r.facebook.com. 1411    IN      A       185.89.218.11
c.ns.c10r.facebook.com. 1411    IN      AAAA    2a03:2880:f1fc:b:face:b00c:0:99
d.ns.c10r.facebook.com. 1411    IN      A       185.89.219.11
d.ns.c10r.facebook.com. 1411    IN      AAAA    2a03:2880:f1fd:b:face:b00c:0:99

;; Query time: 0 msec
;; SERVER: 10.0.2.129#53(10.0.2.129)
;; WHEN: Mon Mar 15 11:23:02 EDT 2021
;; MSG SIZE  rcvd: 333
```

Bash commands as shown in block_facebook.sh are executed on Machine A to implement policies in iptables:

```
#!/bin/bash

# Prevent telnet to Machine B
sudo iptables -A OUTPUT -d 157.240.7.35 -p tcp -j DROP

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

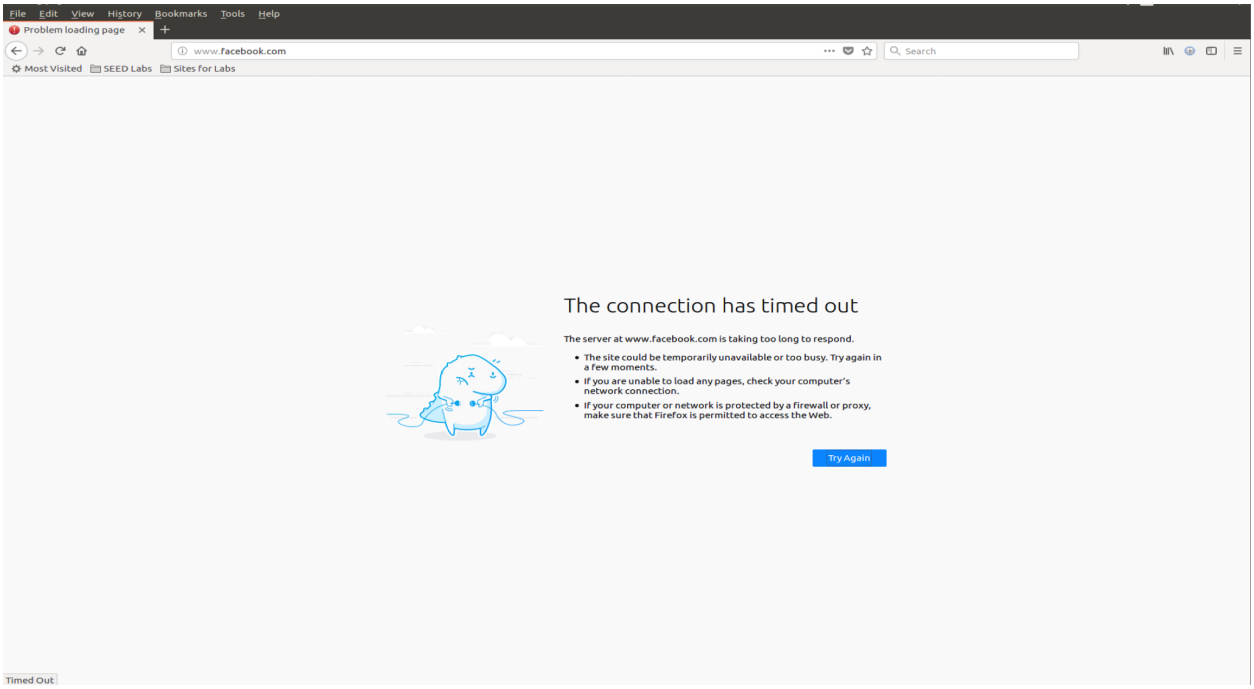
The first commands blocks outgoing packets that are destined for www.facebook.com. The remaining commands specify the default policy as ACCEPT, so that the additional rules declared in the filter table are for packets to be dropped or rejected.

To check that the policies are added properly, we run the following command:

```
[03/15/21]seed@VM:~/.../task3$ 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-01-sin6.facebook.com
```

Name: Kwa Li Ying
Student ID: 1003833

Now, Machine A is NOT able to visit www.facebook.com:



Name: Kwa Li Ying
Student ID: 1003833

Task 3a: Telnet to Machine B through the firewall

To establish an SSH tunnel between Machine A and B and get the tunnel to forward TCP data received on port 8000 on Machine A to port 23 on Machine C, the following command is run on Machine A:

```
[03/15/21]seed@VM:~/.../task3$ ssh -L 8000:10.0.2.130:23 seed@10.0.2.129
seed@10.0.2.129'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: Mon Mar 15 17:37:37 2021 from 10.0.2.128
```

To telnet to Machine C via the tunnel, the following command is run on a new terminal on Machine A:

```
[03/15/21]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:
Last login: Mon Mar 15 17:46:38 EDT 2021 from 10.0.2.130 on pts/18
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.

[03/15/21]seed@VM:~$ hostname -I
10.0.2.130
```

We have successfully telnet to Machine C from Machine A through a ssh tunnel between Machine A and Machine B!

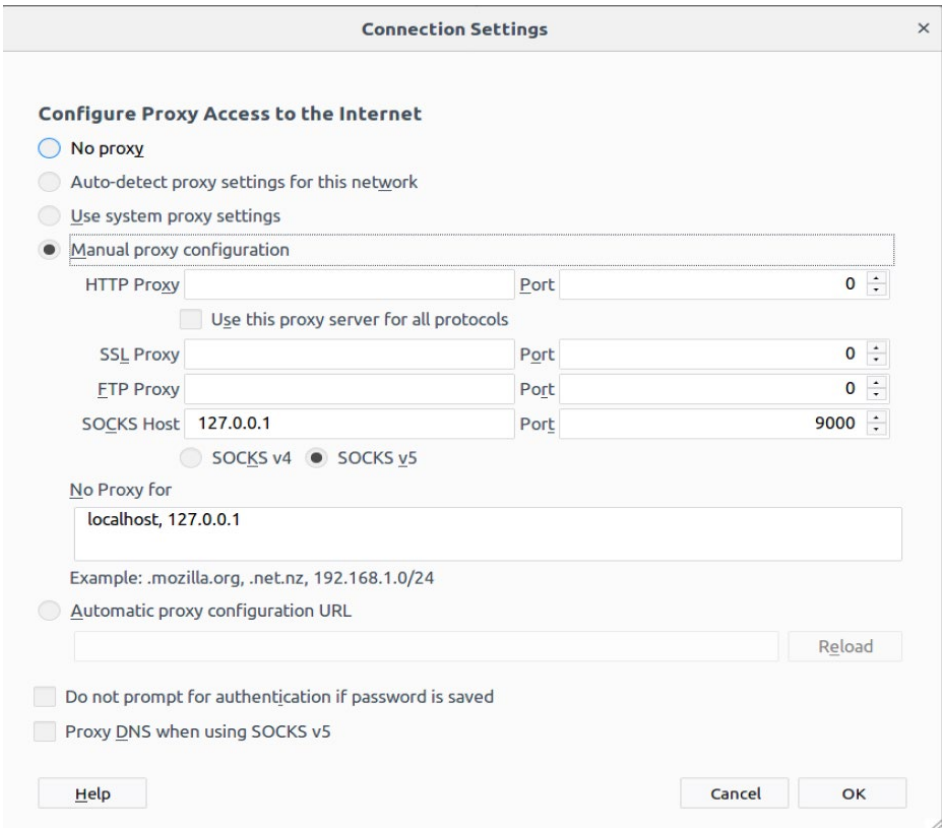
Task 3b: Connect to Facebook using SSH Tunnel

To establish an SSH tunnel between Machine A port 9000 and Machine B using dynamic port forwarding, the following command is run on Machine A:

```
[03/15/21]seed@VM:~/.../task3$ ssh -D 9000 -C seed@10.0.2.129
seed@10.0.2.129'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: Mon Mar 15 17:48:25 2021 from 10.0.2.128
```

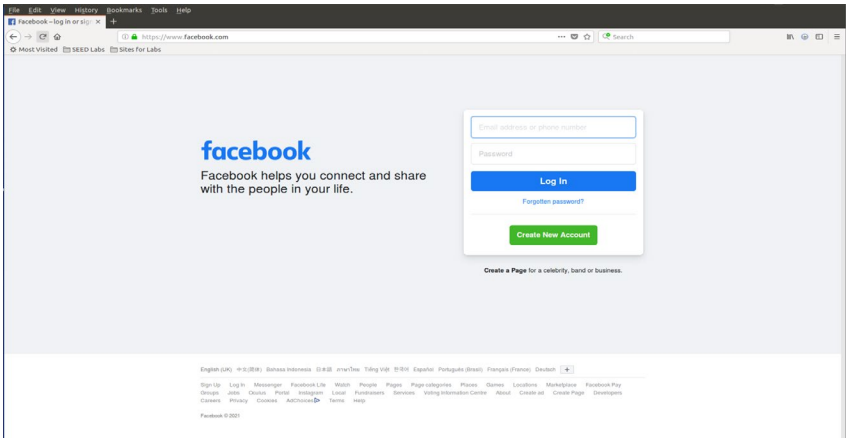
Name: Kwa Li Ying
Student ID: 1003833

To ask Firefox to connect to localhost:9000 every time it needs to connect to a web server, we set up te SOCKS proxy in Firefox:



1. Run Firefox and go visit the Facebook page. Can you see the Facebook page? Please describe your observation.

Yes, the Facebook page can be seen:



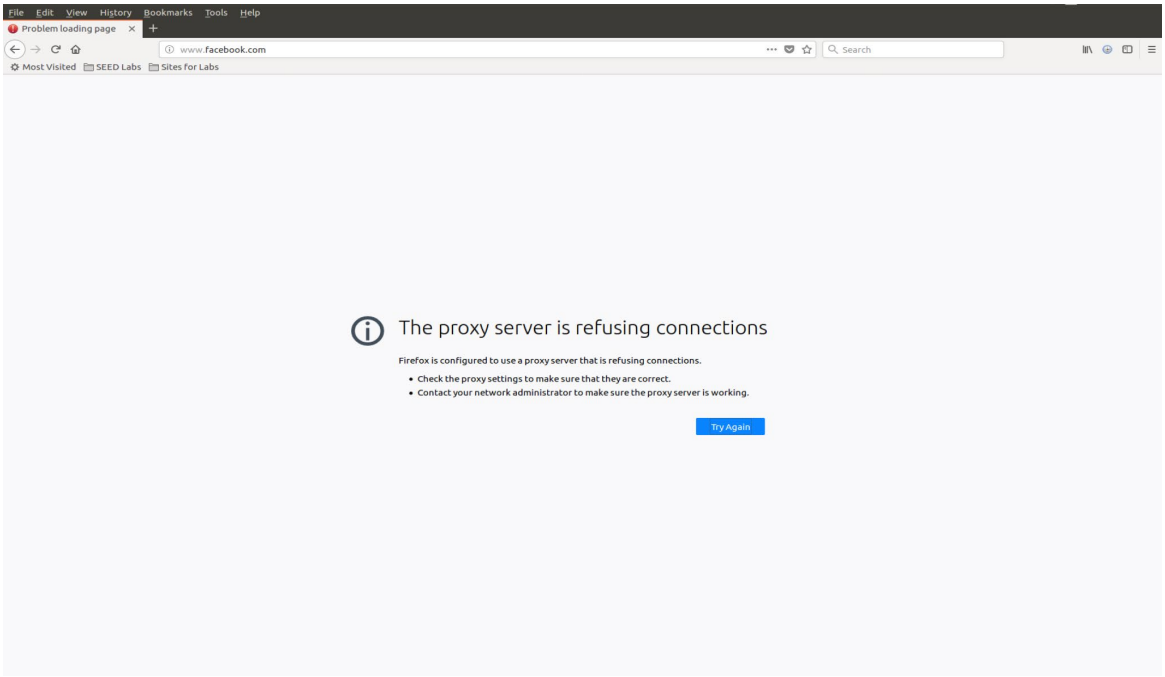
Name: Kwa Li Ying
Student ID: 1003833

2. After you get the facebook page, break the SSH tunnel, clear the Firefox cache, and try the connection again. Please describe your observation.

The SSH tunnel is broken as shown:

```
[03/15/21]seed@VM:~$ hostname -I
10.0.2.129
[03/15/21]seed@VM:~$ exit
logout
```

The browser's cache is cleared and the connection is tried again:



As shown, the page does not load and the reason given is that the proxy server is refusing connections. This is expected because the SSH tunnel is broken and the proxy is no longer in place.

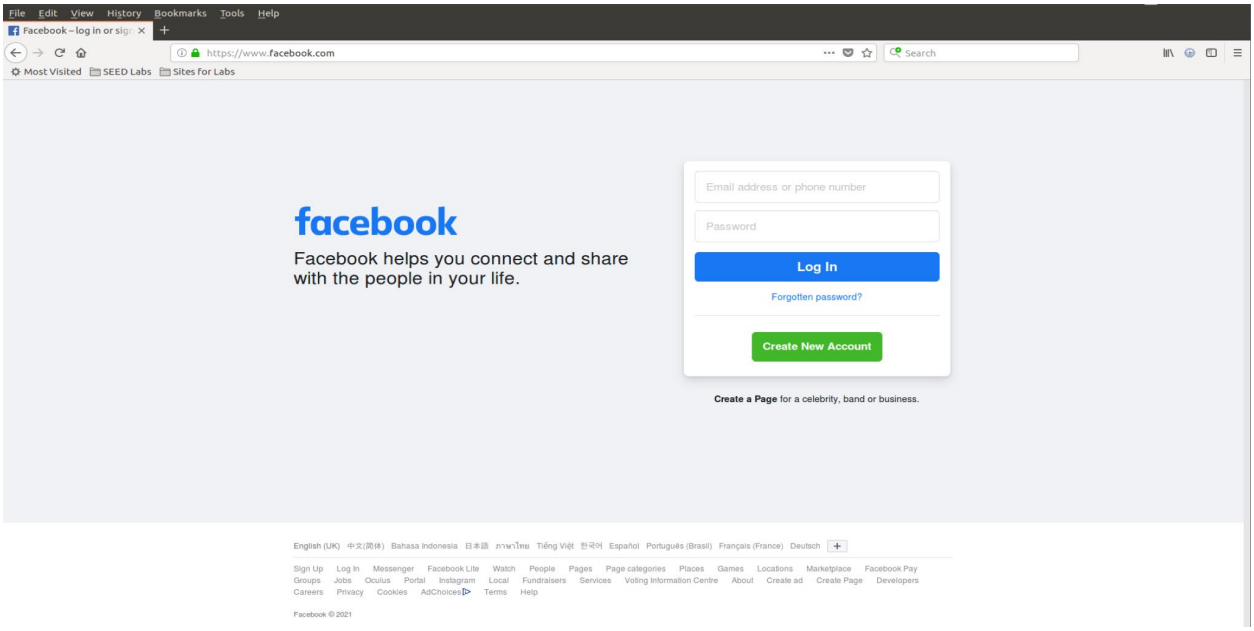
3. Establish the SSH tunnel again and connect to Facebook. Describe your observation.

The SSH tunnel is re-established:

```
[03/15/21]seed@VM:~/.../task3$ ssh -D 9000 -C seed@10.0.2.129
seed@10.0.2.129'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: Mon Mar 15 17:57:54 2021 from 10.0.2.128
```

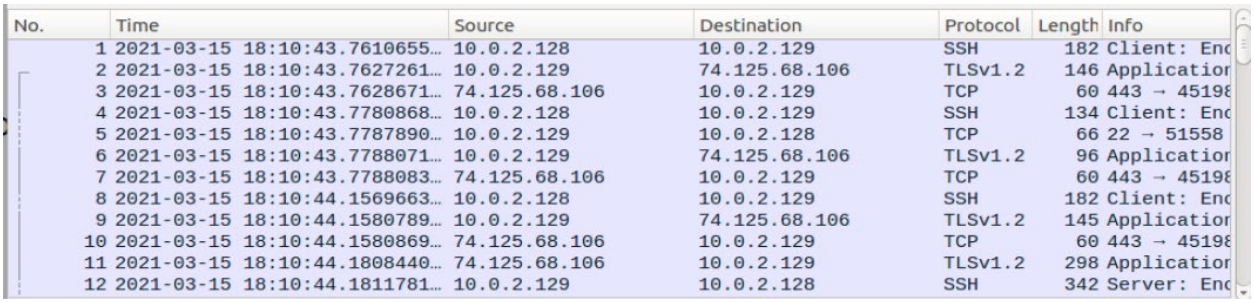
Name: Kwa Li Ying
Student ID: 1003833

The Facebook page is successfully reloaded:



4. Please explain what you have observed, especially on why the SSH tunnel can help bypass the egress filtering. You should use Wireshark to see what exactly is happening on the wire. Please describe your observations and explain them using the packets that you have captured.

As the SSH tunneling is set up and Firefox is configured to connect to localhost:9000 every time it needs to connect to a web server, the HTTP requests are sent to localhost:9000 and in turn forwarded to Machine B via the SSH tunnel. Machine B then forwards these packets to www.facebook.com when the HTTP request is sent to www.facebook.com to render their webpage on Machine A's browser. The Wireshark screenshot below captures the packets that are forwarded to Machine B before they are sent to www.facebook.com's IP address.



No.	Time	Source	Destination	Protocol	Length	Info
1	2021-03-15 18:10:43.7610655…	10.0.2.128	10.0.2.129	SSH	182	Client: Enc
2	2021-03-15 18:10:43.7627261…	10.0.2.129	74.125.68.106	TLSv1.2	146	Applicatior
3	2021-03-15 18:10:43.7628671…	74.125.68.106	10.0.2.129	TCP	60	443 → 45198
4	2021-03-15 18:10:43.7780868…	10.0.2.128	10.0.2.129	SSH	134	Client: Enc
5	2021-03-15 18:10:43.7787890…	10.0.2.129	10.0.2.128	TCP	66	22 → 51558
6	2021-03-15 18:10:43.7788071…	10.0.2.129	74.125.68.106	TLSv1.2	96	Applicatior
7	2021-03-15 18:10:43.7788083…	74.125.68.106	10.0.2.129	TCP	60	443 → 45198
8	2021-03-15 18:10:44.1569663…	10.0.2.128	10.0.2.129	SSH	182	Client: Enc
9	2021-03-15 18:10:44.1580789…	10.0.2.129	74.125.68.106	TLSv1.2	145	Applicatior
10	2021-03-15 18:10:44.1580869…	74.125.68.106	10.0.2.129	TCP	60	443 → 45198
11	2021-03-15 18:10:44.1808440…	74.125.68.106	10.0.2.129	TLSv1.2	298	Applicatior
12	2021-03-15 18:10:44.1811781…	10.0.2.129	10.0.2.128	SSH	342	Server: Enc

Since Machine A drops outgoing packets destined for www.facebook.com (egress filtering), it is not able to load the webpage on its own. By using SSH tunneling, it sends out the HTTP requests via SSH to Machine B, before these requests are forwarded to www.facebook.com. This way, the egress filtering is bypassed.

Name: Kwa Li Ying
Student ID: 1003833

Task 4: Evade Ingress Filtering

IP Address of Machine A: 10.0.2.128

IP Address of Machine B: 10.0.2.129

Setting up the Firewall on Machine A

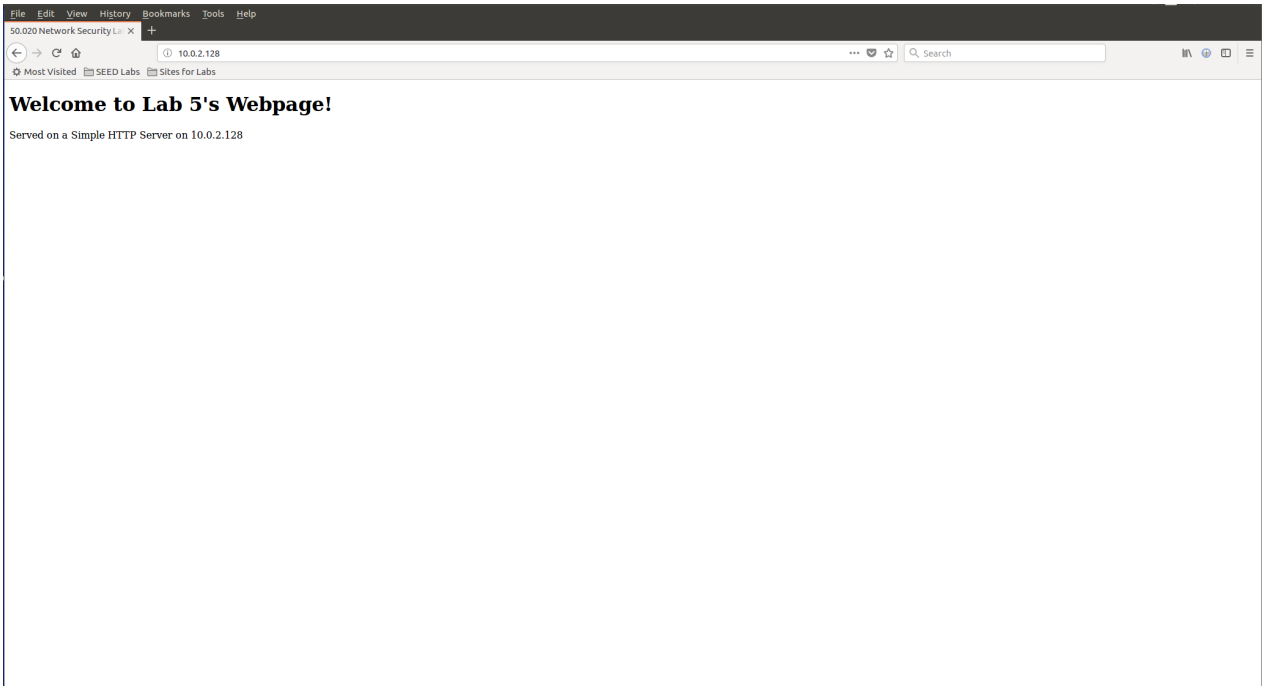
Machine A runs a simple HTTP server with the following contents from index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>50.020 Network Security Lab 5</title>
  </head>
  <body>
    <h1>Welcome to Lab 5's Webpage!</h1>
    <p>Served on a Simple HTTP Server on 10.0.2.128</p>
  </body>
</html>
```

To start the server on port 80, the following command is run on Machine A:

```
[03/15/21]seed@VM:~/.../task4$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Before the firewall is set up, Machine B can visit the website served by Machine A:



Name: Kwa Li Ying
Student ID: 1003833

…as well as SSH to Machine A:

```
[03/15/21]seed@VM:~$ ssh seed@10.0.2.128
seed@10.0.2.128'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: Mon Mar 15 19:18:55 2021 from 10.0.2.130
```

Bash commands as shown in block_task4.sh are executed on Machine A to implement policies in iptables:

```
#!/bin/bash

# Prevent Machine B from accessing web server
sudo iptables -A INPUT -s 10.0.2.129 -p tcp --dport 80 -j DROP

# Prevent Machine B from accessing SSH server
sudo iptables -A INPUT -s 10.0.2.129 -p tcp --dport 22 -j DROP

# Set default policy as ACCEPT
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

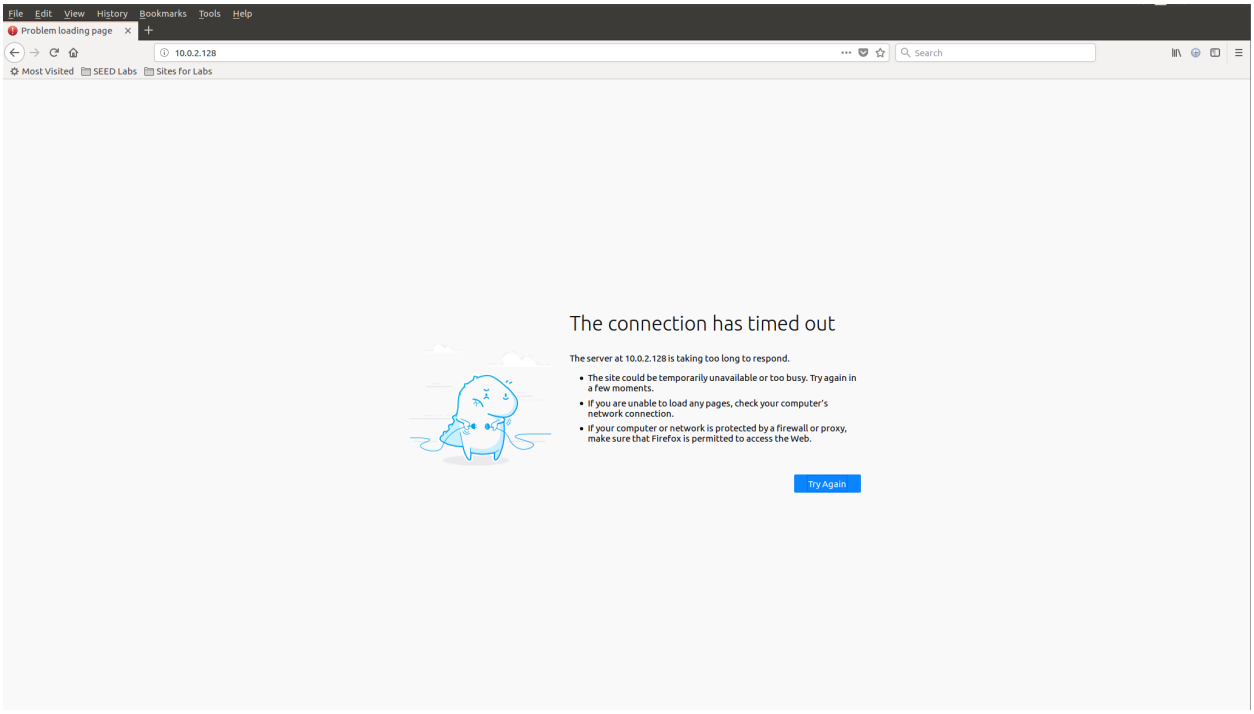
The first command blocks incoming telnet packets from Machine B. The second command blocks incoming HTTP requests from Machine B. The remaining commands specify the default policy as ACCEPT, so that the additional rules declared in the filter table are for packets to be dropped or rejected.

To check that the policies are added properly, we run the following command:

```
[03/15/21]seed@VM:~/.../task4$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.129           anywhere             tcp dpt:http
DROP       tcp  --  10.0.2.129           anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Name: Kwa Li Ying
Student ID: 1003833

Now, Machine B cannot visit the website served by Machine A:



…as well as SSH to Machine A:

```
[03/15/21]seed@VM:~$ ssh seed@10.0.2.128
ssh: connect to host 10.0.2.128 port 22: Connection timed out
```

Name: Kwa Li Ying
Student ID: 1003833

Setting up a Reverse SSH tunnel on Machine A

A reverse SSH tunnel is set up by running the following command on Machine A:

```
[03/15/21]seed@VM:~/.../task4$ ssh -R 8000:localhost:80 seed@10.0.2.129
seed@10.0.2.129'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.
```

This forwards the connections given to port 8000 on Machine B to Machine A via the reverse SSH tunnel and then to localhost:80 on Machine A's side.

Machine B can then access Machine A's website by typing localhost:8000 into the browser: