Lab5 Report

2021 03 15 15:11

Alex W
1003474

3 VMs

Machine A	10.0.2.11
Machine B	10.0.2.12
Machine C	10.0.2.10

task1
Prevent A from doing telnet to Machine B

on Machine B

```
# change default policy to accept
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT

# flush existing configs
sudo iptables -F

sudo iptables -A INPUT -s 10.0.2.11 -p tcp --dport 23 -j DROP
```

append the rule at filter table, INPUT chain, as incoming connections will go through there
-s: filter for machine A's IP, which is 10.0.2.11
as telnet uses tcp port 23, filter for that
the action is DROP to prevent telnet connection from being set up

```
# view the iptables rules
```

```
ubuntu@B  ~
sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       tcp  --  10.0.2.11            anywhere             tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
```

```
num  target     prot opt source               destination

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

on line 1, it correctly shows that the rule is added

```
ubuntu@A  ~
 telnet 10.0.2.12
Trying 10.0.2.12...
telnet: Unable to connect to remote host: Connection timed out
```

Machine A try to connect to B, stuck at trying, and eventually timed out
shows that the firewall rule is working correctly

Prevent B from doing telnet to Machine A

on Machine A

```
# change default policy to accept
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT

# flush existing configs
sudo iptables -F

sudo iptables -A INPUT -s 10.0.2.12 -p tcp --dport 23 -j DROP
```

append the rule at filter table, INPUT chain, as incoming connections will go through there
-s: filter for machine B's IP, which is 10.0.2.12
as telnet uses tcp port 23, filter for that
the action is DROP to prevent telnet connection from being set up

```
# view the iptables rules
```

```
ubuntu@A  ~
 sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       tcp  --  10.0.2.12            anywhere             tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

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

on line 1, it correctly shows that the rule is added

```
ubuntu@B  ~
 telnet 10.0.2.11
Trying 10.0.2.11...
telnet: Unable to connect to remote host: Connection timed out
```

Machine B try to connect to A, stuck at trying, and eventually timed out
shows that the firewall rule is working correctly

Prevent A from visiting an external web site.
site selected: sutd.edu.sg

find out the ip addresses

```
ubuntu@A  ~
 host sutd.edu.sg
sutd.edu.sg has address 45.60.67.5
sutd.edu.sg has address 45.60.65.5
sutd.edu.sg mail is handled by 0 sutd-edu-sg.mail.protection.outlook.com.
```

sutd website has 2 webservers: 45.60.67.5, 45.60.65.5

# add these two rules
sudo iptables -A OUTPUT -d 45.60.67.5 -j DROP
sudo iptables -A OUTPUT -d 45.60.65.5 -j DROP
add at OUTPUT chain of filter table, as it is prevent local traffic from reaching there
-d matches for destination ip address, which are the ip address of both sutd webservers

view the iptables rules

```
ubuntu@A  ~
 sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  anywhere             45.60.67.5
2    DROP       all  --  anywhere             45.60.65.5
```

using wget to get the webpage

```
ubuntu@A  ~
 wget sutd.edu.sg
--2021-03-16 02:40:10--  http://sutd.edu.sg/
Resolving sutd.edu.sg (sutd.edu.sg)... 45.60.67.5, 45.60.65.5
Connecting to sutd.edu.sg (sutd.edu.sg)|45.60.67.5|:80... failed: Connection timed out
```

```
Connecting to sutd.edu.sg (sutd.edu.sg)|45.60.67.5|:80... failed: Connection timed out.
Connecting to sutd.edu.sg (sutd.edu.sg)|45.60.65.5|:80... failed: Connection timed out.
```

it is stuck at connecting to sutd.edu.sg, then the connection timed out
shows that the firewall rule is working correctly

task2
the 5 filters are created as following

```
// Prevent A from doing telnet to Machine B
if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) &&
iph->daddr == in_aton("10.0.2.11"))
{
return NF_DROP;
}
```

similar to iptables
it filters for tcp protocol and port 23, which represents the telnet port
filter for machine A's IP, which is 10.0.2.11
if the conditional match, it will return NF_DROP to drop the packet, preventing the connection from being set up
the action is DROP to prevent telnet connection from being set up

```
// Prevent B from doing telnet to Machine A
if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) &&
iph->daddr == in_aton("10.0.2.12"))
{
return NF_DROP;
}
```

similar to iptables
it filters for tcp protocol and port 23, which represents the telnet port
filter for machine A's IP, which is 10.0.2.12
if the conditional match, it will return NF_DROP to drop the packet, preventing the connection from being set up
the action is DROP to prevent telnet connection from being set up

this is to be installed in machine A

```
nfho.hook = hook_pre_routing_func; /* Handler function */
nfho.hooknum = NF_INET_PRE_ROUTING; /* First hook for IPv4 */
nfho.pf = PF_INET;
nfho.priority = NF_IP_PRI_FIRST; /* Make our function first */
```

```
nf_register_hook(&nfho);
```

the above 2 rules are inserted at PRE_ROUTING so that incoming packets can be filtered out

```
// Prevent A from visiting an external web site.
if (iph->daddr == in_aton("45.60.67.5") || iph->daddr ==
in_aton("45.60.65.5"))
{
return NF_DROP;
}
```

the if condition matches for destination ip address, which are the ip address of both sutd webservers
the action is DROP to prevent web connection from being set up

```
// Prevent A from browsing the web via http (80) and https
(443)
if (iph->protocol == IPPROTO_TCP && (tcph->dest == htons(80) ||
tcph->dest == htons(443)))
{
return NF_DROP;
}
```

the if condition matches for common ports used for http and https traffic, which are tcp 80 and 443
the action is DROP to prevent web connection from being set up

```
// Prevent A from using ssh
if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(22))
{
return NF_DROP;
}
return NF_ACCEPT; /* Accept other packets */
```

the if condition matches for tcp port 22, which is the port used for ssh
the action is DROP to prevent ssh connection from being set up

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

the above 3 rules are inserted at POST_ROUTING so that outgoing packets can be filtered out

run make to compile the module
make

```
ubuntu@A  ~/lab/lab5  master
 make
make -C /lib/modules/4.4.0-186-generic/build M=/home/ubuntu/lab/lab5 modules
make[1]: Entering directory '/usr/src/linux-headers-4.4.0-186-generic'
  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-4.4.0-186-generic'
```

run the following to insert the module to kernel
sudo insmod netfilter.ko

```
ubuntu@A  ~/lab/lab5  master
sudo insmod netfilter.ko
```

testing the filters
// Prevent A from doing telnet to Machine B

```
ubuntu@A  ~
 telnet 10.0.2.12
Trying 10.0.2.12...
telnet: Unable to connect to remote host: Connection timed out
```

Machine A try to connect to B, stuck at trying, and eventually timed out
shows that the netfilter rule is working correctly

// Prevent B from doing telnet to Machine A

```
ubuntu@B  ~
 telnet 10.0.2.11
Trying 10.0.2.11...
telnet: Unable to connect to remote host: Connection timed out
```

Machine B try to connect to A, stuck at trying, and eventually timed out
shows that the netfilter rule is working correctly

// Prevent A from visiting an external web site.

```
ubuntu@A  ~
 wget sutd.edu.sg
--2021-03-16 02:40:10--  http://sutd.edu.sg/
Resolving sutd.edu.sg (sutd.edu.sg)... 45.60.67.5, 45.60.65.5
Connecting to sutd.edu.sg (sutd.edu.sg)|45.60.67.5|:80... failed: Connection timed out.
Connecting to sutd.edu.sg (sutd.edu.sg)|45.60.65.5|:80... failed: Connection timed out.
```

it is stuck at connecting to sutd.edu.sg, then the connection timed out
shows that the firewall rule is working correctly

// Prevent A from browsing the web via http (80) and https

// Prevent A from browsing the web via http (80) and https (443)

```
ubuntu@A  ~
 wget google.com
--2021-03-16 03:38:44--  http://google.com/
Resolving google.com (google.com)... 74.125.24.101, 74.125.24.113, 74.125.24.100, ...
Connecting to google.com (google.com)|74.125.24.101|:80... failed: Connection timed out.
```

// Prevent A from using ssh

```
ubuntu@A  ~
 ssh ubuntu@10.0.2.12
ssh: connect to host 10.0.2.12 port 22: Connection timed out
```

task3

block all outgoing traffic to external telnet servers
use the following rule to block

sudo iptables -A OUTPUT -p tcp --dport 23 -j DROP
since telnet is at tcp port 23, and we want to block outgoing connection, the rule is added at OUTPUT chain

```
ubuntu@A  ~
 sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
```

when trying to connect to telnet at machine C (10.0.2.10), the following happens

```
ubuntu@A  ~
  telnet 10.0.2.10
Trying 10.0.2.10...
```

and it eventually timed out

checking wireshark, there is no packets that match telnet

block all outgoing traffic to www.facebook.com
sudo iptables -A OUTPUT -d 69.171.250.35 -j DROP
the rule is added at OUTPUT to prevent any outgoing connection to destination IP of facebook.com, which is 69.171.250.35

```
ubuntu@A  ~
 sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
```

```
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
2    DROP       all  --  anywhere             edge-star-mini-shv-01-any2.facebook.com
```

```
ubuntu@A  ~
 wget facebook.com
--2021-03-16 04:51:41--  http://facebook.com/
Resolving facebook.com (facebook.com)... 69.171.250.35, 2a03:2880:f1ff:83:face:b00c:0:25de
Connecting to facebook.com (facebook.com)|69.171.250.35|:80... |
```

trying to connect to facebook via wget, results in eventual time out

task3a

set up ssh tunnel between A and B, and use it to forward telnet packets to C:

ssh -L 8000:10.0.2.10:23 10.0.2.12

this command maps localhost:8000 to 10.0.2.10:23 (machine C's telnet), through machine B's ssh tunnel

```
ubuntu@A  ~
 ssh -L 8000:10.0.2.10:23 10.0.2.12
The authenticity of host '10.0.2.12 (10.0.2.12)' can't be established.
ECDSA key fingerprint is SHA256:To0xhpId1WU70HqhceyEGRJzt866KN2LcQenseUc4/Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.12' (ECDSA) to the list of known hosts.
ubuntu@10.0.2.12's password:
```

test on machine A

telnet localhost 8000

```
ubuntu@A  ~
 telnet localhost 8000
Trying ::1...
Connected to localhost.
Escape character is '^]'.
Kali GNU/Linux Rolling
C login: |
```

A is able to successfully telnet to C, bypassing egress filter

wireshark observation

(ip.dst == 10.0.2.12) && (ip.src == 10.0.2.11)

No.	Time	Source	Destination	Protoc	Le	Info
756	175.319243	10.0.2.11	10.0.2.12	SSHv2		Client: Protocol (SSH-2.0-OpenSSH_7.2p2
760	175.322845	10.0.2.11	10.0.2.12	SSHv2		Client: Key Exchange Init
764	175.361585	10.0.2.11	10.0.2.12	SSHv2		Client: Elliptic Curve Diffie-Hellman Key E
785	176.648530	10.0.2.11	10.0.2.12	SSHv2		Client: New Keys
787	176.687383	10.0.2.11	10.0.2.12	SSHv2		Client: Encrypted packet (len=44)
791	176.687955	10.0.2.11	10.0.2.12	SSHv2		Client: Encrypted packet (len=68)
793	176.688610	10.0.2.11	10.0.2.12	SSHv2		Client: Encrypted packet (len=372)
803	177.627397	10.0.2.11	10.0.2.12	SSHv2		Client: Encrypted packet (len=84)

808	177.629579	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=112)
814	177.798883	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=596)
866	191.410635	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
872	191.970032	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
1004	218.298574	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=84)
1073	230.909911	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1078	232.177983	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
1087	232.675887	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1094	232.927403	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1101	233.102535	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1110	233.272258	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1126	233.419997	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=52)
1132	233.420858	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=84)
1137	233.421977	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=60)
1142	233.422801	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1163	236.030175	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
1179	236.034327	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=72)
1209	237.812040	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=84)
1256	252.940613	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=52)
1261	252.941640	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=84)
1266	252.942728	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=60)
1271	252.943776	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
1383	312.946343	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=72)

here shows that A is only communicating with B via SSH, that is not blocked by iptables

1168	236.031184	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1252	252.939715	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1257	252.940832	10.0.2.12	10.0.2.10	TELNET	Telnet Data ...
1259	252.941126	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1262	252.941825	10.0.2.12	10.0.2.10	TELNET	Telnet Data ...
1264	252.942260	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1267	252.942888	10.0.2.12	10.0.2.10	TELNET	Telnet Data ...
1269	252.943362	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1272	252.943926	10.0.2.12	10.0.2.10	TELNET	Telnet Data ...
1273	252.944371	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1279	252.984046	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...
1376	312.945507	10.0.2.10	10.0.2.12	TELNET	Telnet Data ...

the actual telnet session is carried out between B (10.0.2.12) and C (10.0.2.10).
as the telnet session is encrypted inside ssh tunnel, it is not blocked by iptables

task3b
run the following command to set up a dynamic port forwarding tunnel
ssh -D 9000 -C ubuntu@10.0.2.12

wget facebook.com

as the ubuntu 16 is running in server mode without gui, i'll be using proxychains to emulate adding socks5 proxy, and

using wget to emulate firefox

setting up proxy
sudo apt install proxychains -y
sudo nano /etc/proxychains.conf

```
  GNU nano 2.5.3                          File: /etc/proxychains.conf

#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5 127.0.0.1 9000
```

question1
run
proxychains wget facebook.com
this allows wget to use the socks5 proxy set up in proxychains

the following results are gathered

```
ubuntu@A  ~                                                                                    2021-03-16 05:17:43
 proxychains wget facebook.com
ProxyChains-3.1 (http://proxychains.sf.net)
URL transformed to HTTPS due to an HSTS policy
--2021-03-16 05:17:54--  https://facebook.com/
Resolving facebook.com (facebook.com)... |DNS-request| facebook.com
|S-chain|-<>-127.0.0.1:9000-<><>-4.2.2.2:53-<><>-OK
|DNS-response| facebook.com is 157.240.192.35
157.240.192.35
Connecting to facebook.com (facebook.com)|157.240.192.35|:443... |S-chain|-<>-127.0.0.1:9000-<><>-157.240.192.35:443-<><>-OK
connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.facebook.com/ [following]
--2021-03-16 05:17:54--  https://www.facebook.com/
Resolving www.facebook.com (www.facebook.com)... |DNS-request| www.facebook.com
|S-chain|-<>-127.0.0.1:9000-<><>-4.2.2.2:53-<><>-OK
|DNS-response| www.facebook.com is 157.240.198.35
157.240.198.35
Connecting to www.facebook.com (www.facebook.com)|157.240.198.35|:443... |S-chain|-<>-127.0.0.1:9000-<><>-157.240.198.35:443-<><>-OK
connected.
HTTP request sent, awaiting response... 302 Found
Location: https://www.facebook.com/unsupportedbrowser [following]
--2021-03-16 05:17:55--  https://www.facebook.com/unsupportedbrowser
Reusing existing connection to www.facebook.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.4'

index.html.4                        [  <=>                                                  ] 149.22K   412KB/s    in 0.4s

2021-03-16 05:17:56 (412 KB/s) - 'index.html.4' saved [152800]
```

wget is able to download content from facebook.com, evading the egress filter

question2

question2

```
ubuntu@A  ~
 proxychains wget facebook.com
ProxyChains-3.1 (http://proxychains.sf.net)
URL transformed to HTTPS due to an HSTS policy
--2021-03-16 05:19:23--  https://facebook.com/
Resolving facebook.com (facebook.com)... |DNS-request| facebook.com
|S-chain|-<>-127.0.0.1:9000-<--timeout
|DNS-response|: facebook.com does not exist
failed: Unknown error.
wget: unable to resolve host address 'facebook.com'
```

the tunnel is broken hence localhost:9000 proxy server is no longer functional

from the screenshot above, it appears as if there is no internet connection

question3

```
ubuntu@A  ~                                                                                    2021-03-16 05:19:23
 proxychains wget facebook.com
ProxyChains-3.1 (http://proxychains.sf.net)
URL transformed to HTTPS due to an HSTS policy
--2021-03-16 05:20:15--  https://facebook.com/
Resolving facebook.com (facebook.com)... |DNS-request| facebook.com
|S-chain|-<>-127.0.0.1:9000-<><>-4.2.2.2:53-<><>-OK
|DNS-response| facebook.com is 157.240.23.35
157.240.23.35
Connecting to facebook.com (facebook.com)|157.240.23.35|:443... |S-chain|-<>-127.0.0.1:9000-<><>-157.240.23.35:443-<><>-OK
connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.facebook.com/ [following]
--2021-03-16 05:20:15--  https://www.facebook.com/
Resolving www.facebook.com (www.facebook.com)... |DNS-request| www.facebook.com
|S-chain|-<>-127.0.0.1:9000-<><>-4.2.2.2:53-<><>-OK
|DNS-response| www.facebook.com is 157.240.239.35
157.240.239.35
Connecting to www.facebook.com (www.facebook.com)|157.240.239.35|:443... |S-chain|-<>-127.0.0.1:9000-<><>-157.240.239.35:443-<><>-OK
connected.
HTTP request sent, awaiting response... 302 Found
Location: https://www.facebook.com/unsupportedbrowser [following]
--2021-03-16 05:20:16--  https://www.facebook.com/unsupportedbrowser
Reusing existing connection to www.facebook.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.5'

index.html.5                        [  <=>                                                  ] 149.22K   401KB/s    in 0.4s

2021-03-16 05:20:17 (401 KB/s) - 'index.html.5' saved [152799]
```

same as question1

wget is able to download content from facebook.com, evading the egress filter

using wireshark

14…	1778.896807	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=68)
14…	1778.918486	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1778.920665	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=60)
14…	1778.933302	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1778.962847	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=316)
14…	1778.999498	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=164)
14…	1779.035491	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=204)
14…	1779.279478	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1779.284435	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=52)
14…	1779.308261	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1779.310492	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=44)
14…	1779.405360	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1779.405947	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=100)
14…	1779.480251	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=148)
14…	1779.551488	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=212)

14…	1779.929667	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=60)
14…	1779.929832	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1779.930607	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=228)
14…	1779.965480	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1780.614807	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1780.698988	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)
14…	1780.847236	10.0.2.11	10.0.2.12	SSHv2	Client: Encrypted packet (len=36)

here shows that A is only communicating with B via SSH, that is not blocked by iptables
the actual communication with facebook.com is relayed through the SSH tunnel, represented by encrypted packets, hence they are not filtered out by the firewall

task4
setup
use caddy as a webserver on machine A

sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo apt-key add -
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee -a /etc/apt/sources.list.d/caddy-stable.list
sudo apt update
sudo apt install caddy

running the file server
sudo caddy file-server

```
ubuntu@A  ~
 sudo caddy file-server
2021/03/16 12:49:44.845 WARN    admin   admin endpoint disabled
2021/03/16 12:49:44.845 INFO    http    server is listening only on the HTTP port, so no automatic HTTPS will
{"server_name": "static", "http_port": 80}
2021/03/16 12:49:44.845 INFO    tls.cache.maintenance   started background certificate maintenance      {"cach
2021/03/16 12:49:44.846 INFO    autosaved config        {"file": "/home/ubuntu/.config/caddy/autosave.json"}
2021/03/16 05:49:44 Caddy 2 serving static files on :80
2021/03/16 12:49:44.846 INFO    tls     cleaned up storage units
```

disable incoming ssh and webserver
sudo iptables -A INPUT -p tcp --dport 22 -j DROP
sudo iptables -A INPUT -p tcp --dport 80 -j DROP

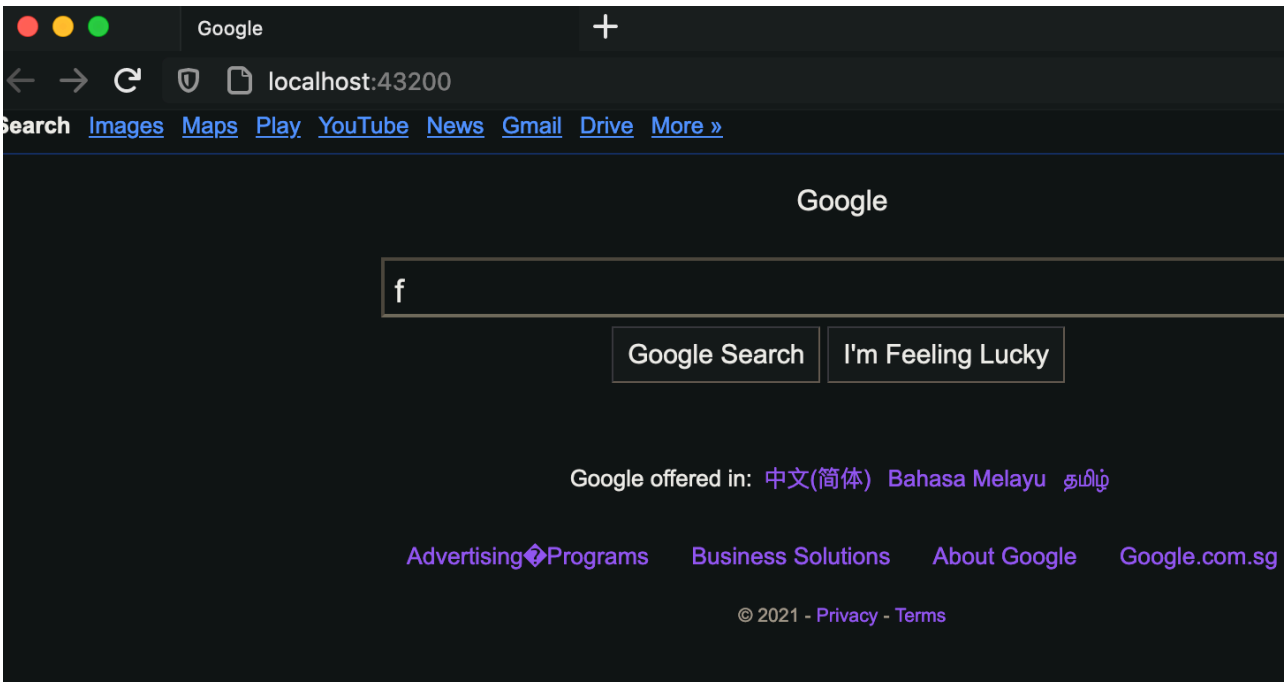
```
ubuntu@A  ~/lab/lab5  master
 sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
2    DROP       tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
```

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

machine A cant accept ssh connection, as it is blocked by firewall
however, machine A can create ssh connection to any computers in external networks
in this case, a reverse ssh tunnel can be set up

```
ssh -R 43200:localhost:80 alex@10.0.2.1
```

on user's home machine, 10.0.2.1
he is able to visit localhost:43200 to access machine A's webserver as shown



this shows that the task is completed successfully, with all the restricts bypassed