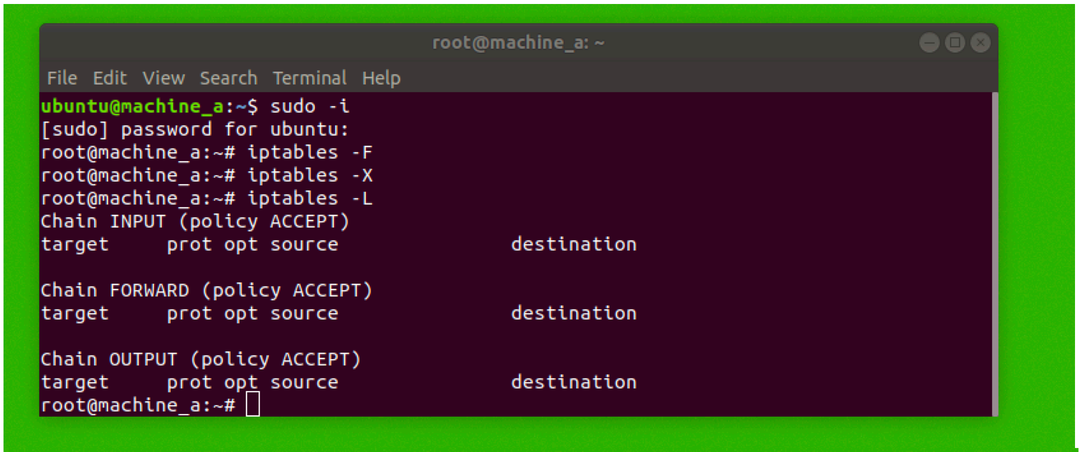
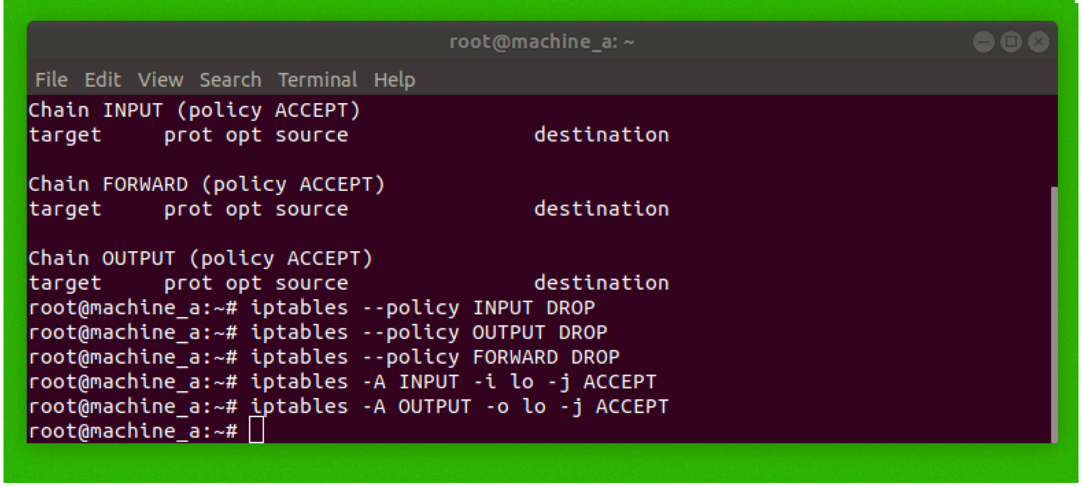
Gabriel Naples
4/17/19
Firewall Exploration Lab

Section 1

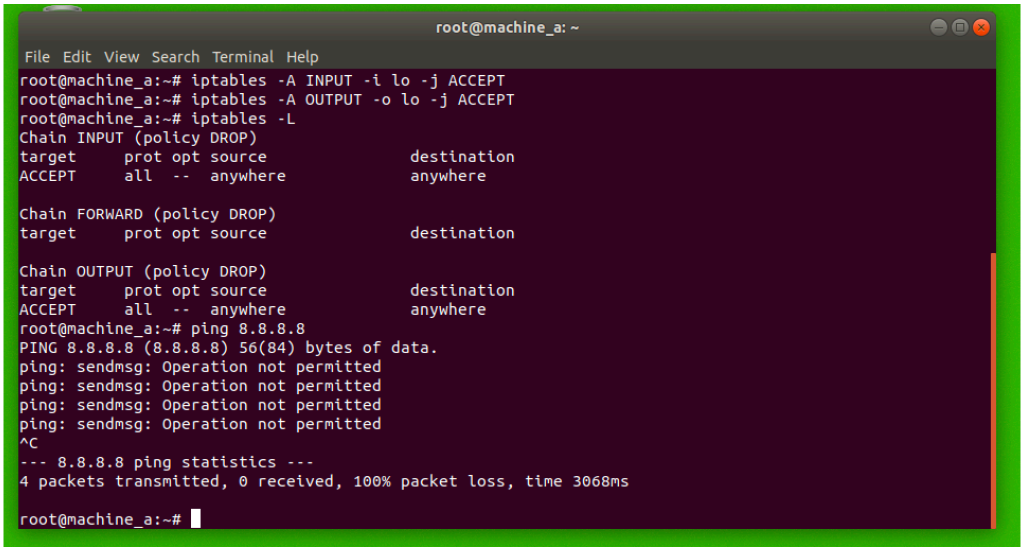
Task 1



```
ubuntu@machine_a:~$ sudo -i
[sudo] password for ubuntu:
root@machine_a:~# iptables -F
root@machine_a:~# iptables -X
root@machine_a:~# 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
root@machine_a:~#
```



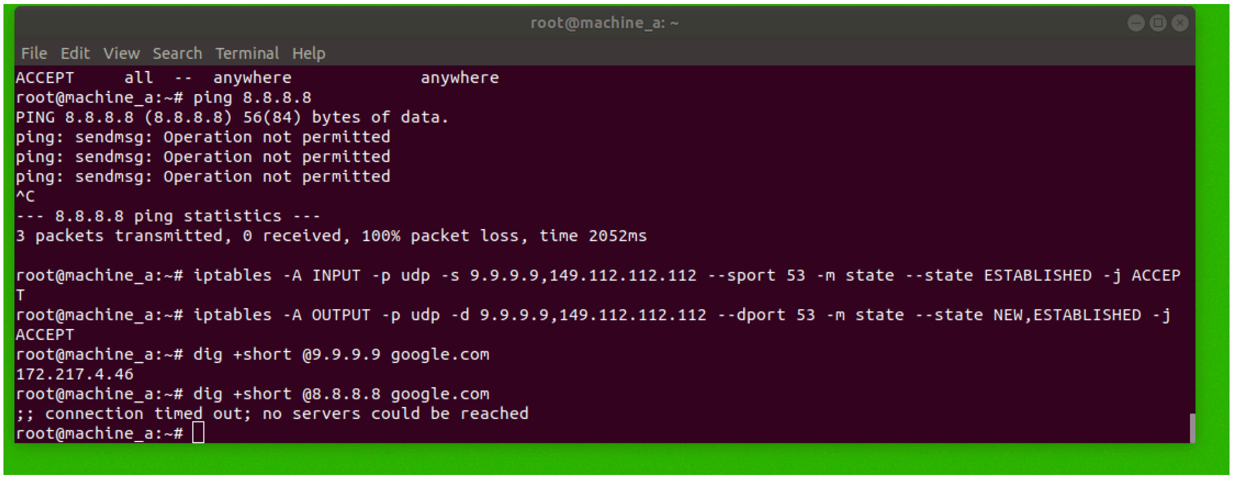
```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@machine_a:~# iptables --policy INPUT DROP
root@machine_a:~# iptables --policy OUTPUT DROP
root@machine_a:~# iptables --policy FORWARD DROP
root@machine_a:~# iptables -A INPUT -i lo -j ACCEPT
root@machine_a:~# iptables -A OUTPUT -o lo -j ACCEPT
root@machine_a:~#
```



```
root@machine_a:~# iptables -A INPUT -i lo -j ACCEPT
root@machine_a:~# iptables -A OUTPUT -o lo -j ACCEPT
root@machine_a:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
root@machine_a:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3068ms

root@machine_a:~# 
```

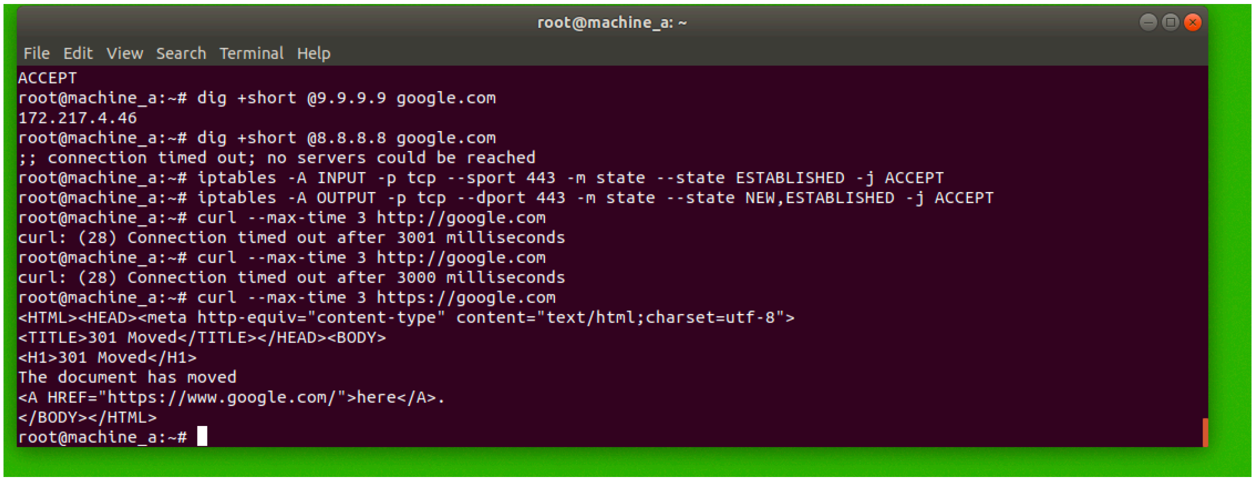
2. IPTables acts as a packet filter type of firewall.
3. A packet filtering firewall filters packets without any knowledge of previous packets or connections. A stateful firewall is able to keep track of ongoing connections that are happening over UDP and TCP. The advantage of this would be if you want the user to be able to access websites but did not want others to connect to your computer on the web ports. A common use in modern industry is putting a packet filtering firewall on the internet facing router that blocks obvious malicious traffic and reduces the load on the stateful inspection firewall place deeper in the system.

Task 2



```
ACCEPT     all  --  anywhere             anywhere
root@machine_a:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2052ms

root@machine_a:~# iptables -A INPUT -p udp -s 9.9.9.9,149.112.112.112 --sport 53 -m state --state ESTABLISHED -j ACCEP
T
root@machine_a:~# iptables -A OUTPUT -p udp -d 9.9.9.9,149.112.112.112 --dport 53 -m state --state NEW,ESTABLISHED -j
ACCEPT
root@machine_a:~# dig +short @9.9.9.9 google.com
172.217.4.46
root@machine_a:~# dig +short @8.8.8.8 google.com
;; connection timed out; no servers could be reached
root@machine_a:~# 
```

2. The first DNS query works because the DNS resolver being used is 9.9.9.9 and not 8.8.8.8. The firewall rules are configured to use the specific DNS resolver that was approved by the company and that resolver works to distribute the IP addresses. If the company approved of a resolver with 8.8.8.8 and also had access to it, then the second command would have worked if the rules were configured for it.



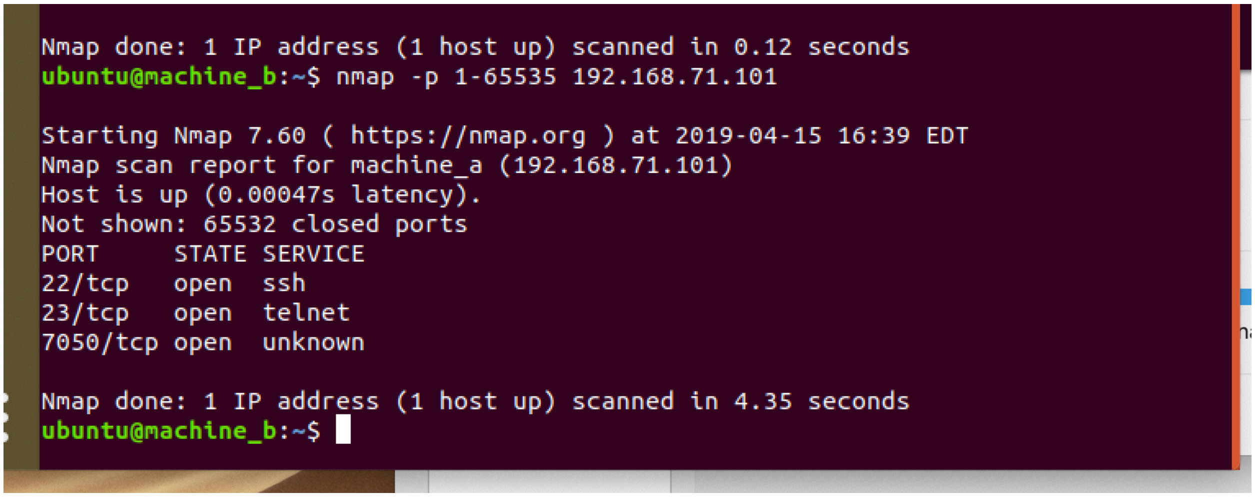
```
ACCEPT
root@machine_a:~# dig +short @9.9.9.9 google.com
172.217.4.46
root@machine_a:~# dig +short @8.8.8.8 google.com
;; connection timed out; no servers could be reached
root@machine_a:~# iptables -A INPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
root@machine_a:~# iptables -A OUTPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
root@machine_a:~# curl --max-time 3 http://google.com
curl: (28) Connection timed out after 3001 milliseconds
root@machine_a:~# curl --max-time 3 http://google.com
curl: (28) Connection timed out after 3000 milliseconds
root@machine_a:~# curl --max-time 3 https://google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.com/">here</A>.
</BODY></HTML>
root@machine_a:~# 
```

2. I navigated to Reddit, Amazon, and PSU and all three of these websites worked, likely because all of these use https and have secure connections.

3. I think that the HTTPS-only policy is very reasonable especially in a company setting. It is likely rare anyone would ever need or want to navigate to a site that is not secured with HTTPS and having this policy in place stops a lot of accidents where employees that may not understand secure websites can access a malicious server that can damage the company.

Sections 2

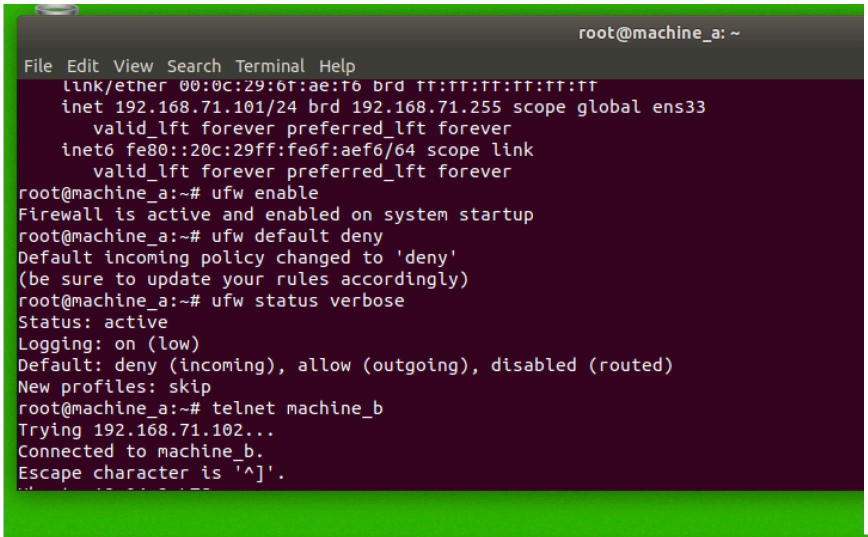
Task 1



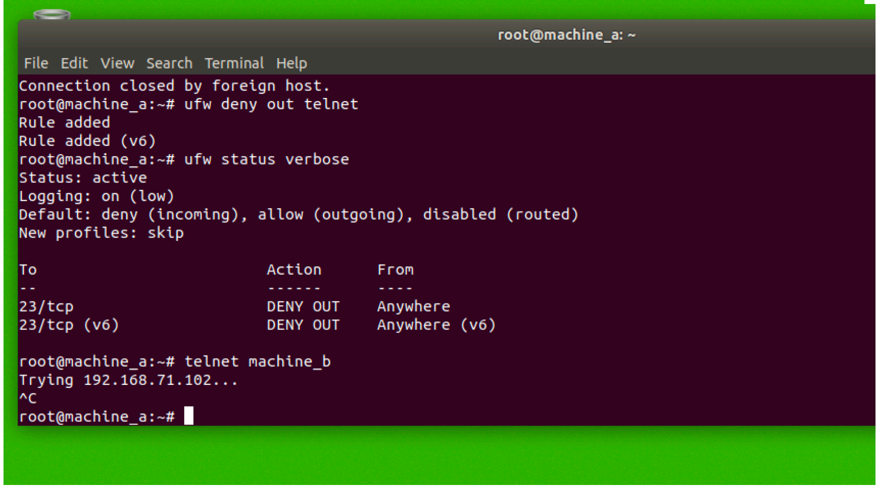
```
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
ubuntu@machine_b:~$ nmap -p 1-65535 192.168.71.101

Starting Nmap 7.60 ( https://nmap.org ) at 2019-04-15 16:39 EDT
Nmap scan report for machine_a (192.168.71.101)
Host is up (0.00047s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
23/tcp   open  telnet
7050/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.35 seconds
ubuntu@machine_b:~$ 
```

2. The developer’s site is running on port 7050 which is the only other open tcp port that isn’t ssh or telnet. For this command I also had to adjust the command to scan for ports 1-65535 which is all possible ports since the default was only 1000.

Task 2 – 3

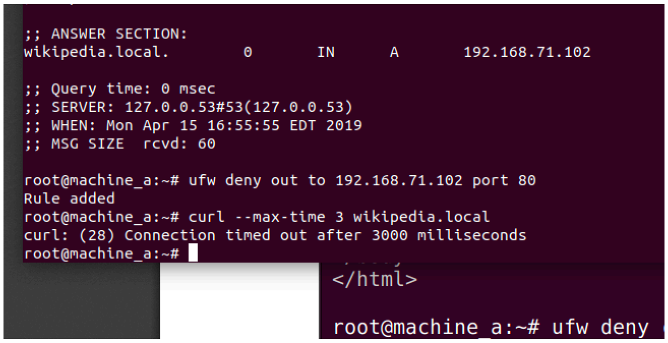


```
    link/ether 00:0c:29:6f:ae:f6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.71.101/24 brd 192.168.71.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe6f:aef6/64 scope link
       valid_lft forever preferred_lft forever
root@machine_a:~# ufw enable
Firewall is active and enabled on system startup
root@machine_a:~# ufw default deny
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
root@machine_a:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
root@machine_a:~# telnet machine_b
Trying 192.168.71.102...
Connected to machine_b.
Escape character is '^]'.
```



```
Connection closed by foreign host.
root@machine_a:~# ufw deny out telnet
Rule added
Rule added (v6)
root@machine_a:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
23/tcp                     DENY OUT    Anywhere
23/tcp (v6)                DENY OUT    Anywhere (v6)

root@machine_a:~# telnet machine_b
Trying 192.168.71.102...
^C
root@machine_a:~# 
```

Task 4



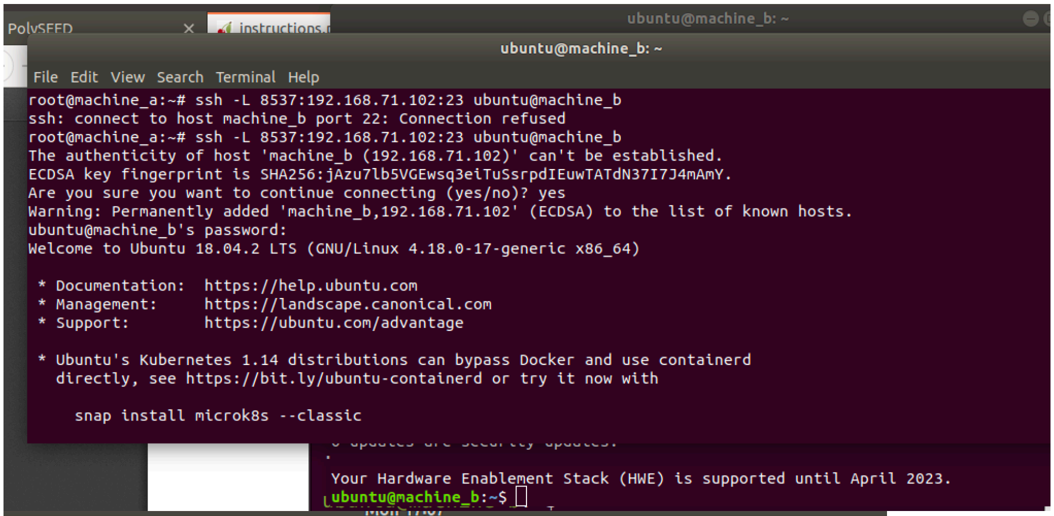
```
;; ANSWER SECTION:
wikipedia.local.        0       IN      A       192.168.71.102

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Mon Apr 15 16:55:55 EDT 2019
;; MSG SIZE  rcvd: 60

root@machine_a:~# ufw deny out to 192.168.71.102 port 80
Rule added
root@machine_a:~# curl --max-time 3 wikipedia.local
curl: (28) Connection timed out after 3000 milliseconds
root@machine_a:~# 
```

```
</html>

root@machine_a:~# ufw deny
```

2. ufw is a netfilter type firewall. It is used to block certain network packets or ip addresses.

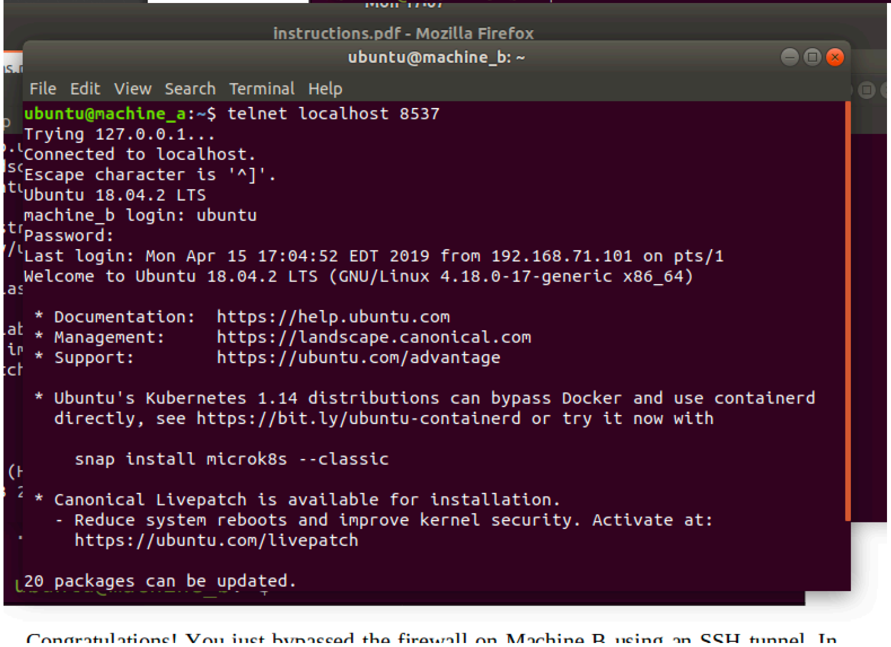
Section 3

Task 1



```
root@machine_a:~# ssh -L 8537:192.168.71.102:23 ubuntu@machine_b
ssh: connect to host machine_b port 22: Connection refused
root@machine_a:~# ssh -L 8537:192.168.71.102:23 ubuntu@machine_b
The authenticity of host 'machine_b (192.168.71.102)' can't be established.
ECDSA key fingerprint is SHA256:jAzu7lb5VGEwsq3eiTuSsrpdIEuwTATdN37I7J4mAmY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'machine_b,192.168.71.102' (ECDSA) to the list of known hosts.
ubuntu@machine_b's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-17-generic x86_64)

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

 * Ubuntu's Kubernetes 1.14 distributions can bypass Docker and use containerd
   directly, see https://bit.ly/ubuntu-containerd or try it now with

     snap install microk8s --classic
```

```
Your Hardware Enablement Stack (HWE) is supported until April 2023.
ubuntu@machine_b:~$
```



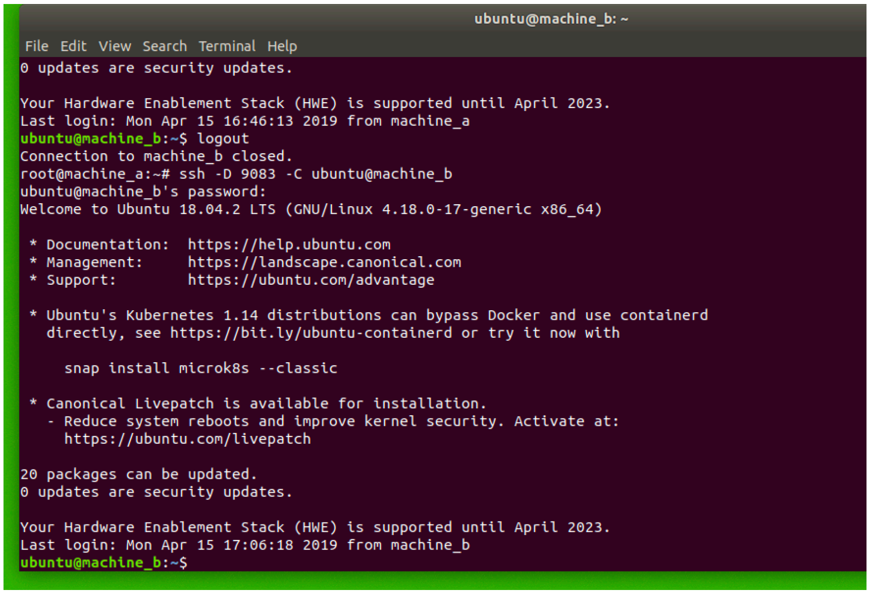
```
ubuntu@machine_a:~$ telnet localhost 8537
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 18.04.2 LTS
machine_b login: ubuntu
Password:
Last login: Mon Apr 15 17:04:52 EDT 2019 from 192.168.71.101 on pts/1
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-17-generic x86_64)

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

 * Ubuntu's Kubernetes 1.14 distributions can bypass Docker and use containerd
   directly, see https://bit.ly/ubuntu-containerd or try it now with

     snap install microk8s --classic

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

20 packages can be updated.
```

Congratulations! You just bypassed the firewall on Machine B using an SSH tunnel. In

instructions.pdf - Mozilla Firefox
ubuntu@machine_b: ~
File Edit View Search Terminal Help

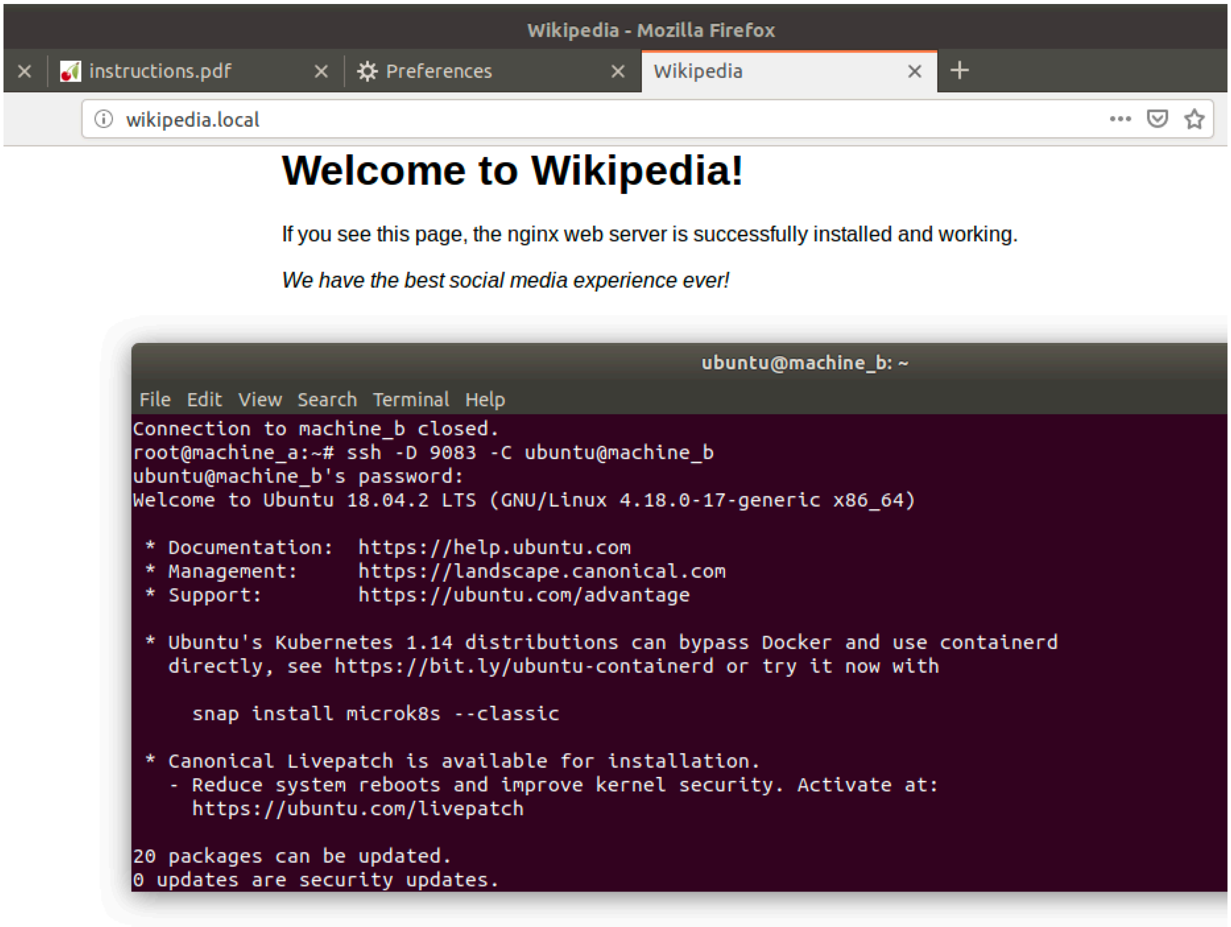
2. The terminal with ssh must keep running because that is the network tunnel. The ssh command is creating a tunnel from port 8537 on the local host and tunneling into port 23 on machine b, which is the telnet port. The telnet command then accesses this network tunnel by interacting with the localhost port 8537 instead of trying to directly access port 23 on the destination machine. This is what enables it to get to the telnet of machine b and to keep the tunnel open the command window must be open.

Task 2



```
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Mon Apr 15 16:46:13 2019 from machine_a
ubuntu@machine_b:~$ logout
Connection to machine_b closed.
root@machine_a:~# ssh -D 9083 -C ubuntu@machine_b
ubuntu@machine_b's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-17-generic x86_64)

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

 * Ubuntu's Kubernetes 1.14 distributions can bypass Docker and use containerd
   directly, see https://bit.ly/ubuntu-containerd or try it now with

     snap install microk8s --classic

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

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

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Mon Apr 15 17:06:18 2019 from machine_b
ubuntu@machine_b:~$
```

Task 3



Welcome to Wikipedia!

If you see this page, the nginx web server is successfully installed and working.

We have the best social media experience ever!

```
Connection to machine_b closed.
root@machine_a:~# ssh -D 9083 -C ubuntu@machine_b
ubuntu@machine_b's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-17-generic x86_64)

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

 * Ubuntu's Kubernetes 1.14 distributions can bypass Docker and use containerd
   directly, see https://bit.ly/ubuntu-containerd or try it now with

     snap install microk8s --classic

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

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