

Task 1:

Host U can communicate with VPN Server.

```

SSH-in-browser SSH-in-browser
Expanded Security Maintenance for Applications is not enabled.
34 updates can be applied immediately.
22 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
16 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
New release '22.04.3 LTS' available.
Run 'do-release-upgrade!' to upgrade to it.

Last login: Wed Feb 28 13:44:31 2024 from 55.235.244.34
lakshmiswaminathan08@seed-ubuntu:~$ sudo su
root@seed-ubuntu:/home/lakshmiswaminathan08# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c70af9603218 handsonsecurity/seed-ubuntu:large "bash -c ' tail -f /...'" 3 minutes ago
o Up 3 minutes ago client-10.9.0.5
3511b9d2b82 handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" 3 minutes ago
o Up 3 minutes ago host-192.168.60.6
2587040d58c handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" 3 minutes ago
o Up 3 minutes ago host-192.168.60.5
dfd6763bae4b handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" 3 minutes ago
o Up 3 minutes ago host-vxlan-router
root@seed-ubuntu:/home/lakshmiswaminathan08# docker exec -it c bash
Error response from daemon: multiple IDs found with provided prefix: c
root@seed-ubuntu:/home/lakshmiswaminathan08# docker exec -it c70 bash
root@c70af9603218:~# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.512 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.094 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.100 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.176 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.136 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.106 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.115 ms
64 bytes from 10.9.0.11: icmp_seq=8 ttl=64 time=0.112 ms
64 bytes from 10.9.0.11: icmp_seq=9 ttl=64 time=0.128 ms
64 bytes from 10.9.0.11: icmp_seq=10 ttl=64 time=0.111 ms
64 bytes from 10.9.0.11: icmp_seq=11 ttl=64 time=0.130 ms
64 bytes from 10.9.0.11: icmp_seq=12 ttl=64 time=0.122 ms
64 bytes from 10.9.0.11: icmp_seq=13 ttl=64 time=0.111 ms
64 bytes from 10.9.0.11: icmp_seq=14 ttl=64 time=0.104 ms
64 bytes from 10.9.0.11: icmp_seq=15 ttl=64 time=0.112 ms
[...]

```

VPN Server can communicate with Host V.

```

SSH-in-browser SSH-in-browser
root@4fd46763bae4b:~# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.137 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.136 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=64 time=0.115 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=64 time=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=64 time=0.128 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=64 time=0.128 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=64 time=0.111 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=64 time=0.130 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=64 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=64 time=0.111 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=64 time=0.104 ms
[...]
root@4fd46763bae4b:~# tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
14:52:17.108594 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 1, length 64
14:52:17.108673 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 1, length 64
14:52:18.137762 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 2, length 64
14:52:18.137841 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 2, length 64
14:52:19.161849 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 3, length 64
14:52:19.161911 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 3, length 64
14:52:20.185776 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 4, length 64
14:52:20.185842 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 4, length 64
14:52:21.209751 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 5, length 64
14:52:21.209827 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 5, length 64
14:52:22.233789 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 6, length 64
14:52:22.233852 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 6, length 64
14:52:22.329684 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
14:52:22.329840 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
14:52:22.329858 ARP, Reply 192.168.60.11 is-at 02:42:00:a8:c0:b, length 28
14:52:22.329865 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
14:52:23.257819 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 7, length 64
14:52:23.257884 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 7, length 64
14:52:24.281752 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 8, length 64
14:52:24.281821 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 8, length 64
14:52:25.305742 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 3, seq 9, length 64
14:52:25.305802 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 3, seq 9, length 64
[...]

```

Host U is not able to communicate with Host V.

```

SSH-in-browser
Users logged in:
IPv4 address for br-08ea46e306eb: 192.168.60.1
IPv4 address for br-a69a48571036: 10.9.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for ens4: 10.128.0.2

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

16 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' is available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Feb 28 14:41:29 2024 from 35.235.244.34
lakshmiswaminathan08@seed-ubuntu:~$ sudo su
root@seed-ubuntu:/home/lakshmiswaminathan08# cd ~/Downloads/VPN
root@seed-ubuntu:~/Downloads/VPN# docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
c70af9603218 handsonsecurity/seed-ubuntu:large "bash -c ' tail -f /...'" About an hour ago
Up About an hour client-10.9.0.5
3511b39d2b82 handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" About an hour ago
Up About an hour host-192.168.60.6
2587f040d586 handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" About an hour ago
Up About an hour host-192.168.60.5
df6763bae4b handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" About an hour ago
Up About an hour server-router
root@seed-ubuntu:~/Downloads/VPN# docker exec -it c70 bash
root@c70af9603218:~# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
26 packets transmitted, 0 received, 100% packet loss, time 25608ms
root@c70af9603218:~#

```



```

SSH-in-browser
lakshmiswaminathan08@seed-ubuntu:~$ sudo su
root@seed-ubuntu:/home/lakshmiswaminathan08# cd ~/Downloads/VPN
root@seed-ubuntu:~/Downloads/VPN# docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
c70af9603218 handsonsecurity/seed-ubuntu:large "bash -c ' tail -f /...'" 2 hours ago
3511b39d2b82 handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" 2 hours ago
Up 2 hours host-192.168.60.6
2587f040d586 handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" 2 hours ago
Up 2 hours host-192.168.60.5
df6763bae4b handsonsecurity/seed-ubuntu:large "bash -c ' ip route ...'" 2 hours ago
Up 2 hours server-router
root@seed-ubuntu:~/Downloads/VPN# docker exec -it df6 bash
root@df6763bae4b:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 10.9.0.11 netmask 255.255.255.0 broadcast 10.9.0.255
    ether 02:42:0a:09:00:0b txqueuelen 0 (Ethernet)
      RX packets 90 bytes 13591 (13.5 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 21 bytes 1946 (1.9 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.60.11 netmask 255.255.255.0 broadcast 192.168.60.255
    ether 02:42:0a:09:00:0b txqueuelen 0 (Ethernet)
      RX packets 105 bytes 14755 (14.7 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 38 bytes 3500 (3.5 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
      RX packets 197 bytes 197 (197.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 2 bytes 197 (197.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@df6763bae4b:~# tcpdump -i eth1 -n
tcpdump: verbose output suppressed; use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes

```

Task 2: Create and Configure TUN Interface

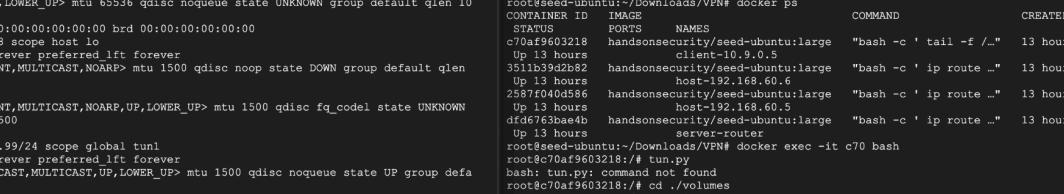
Task 2.a: Name of the Interface

```

lakshmiswaminathan08@seed-ubuntu:~$ sudo su
root@seed-ubuntu:/home/lakshmiswaminathan08# cd ~/Downloads/VPN
root@seed-ubuntu:~/Downloads/VPN# docker exec -it c70 bash
root@c70af9603218:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 100
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.0.0.1 scope host lo
        valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
3: tun1: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
8: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group defa
    ult
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@c70af9603218:~#

```

Task 2.b: Set up the TUN Interface



```
root@c70af9603218:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 brd 0.0.0.0 scope host lo
            valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
4: tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 brd 192.168.53.255 scope global tun1
        valid_lft forever preferred_lft forever
6: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ffff:ffff:ff:link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@c70af9603218:~#
```



```
root@seed-ubuntu:/home/lakshmi_swaminathan08# cd ~/Downloads/VPN
root@seed-ubuntu:~/Downloads/VPN# docker ps
CONTAINER ID        IMAGE               STATUS             PORTS          NAMES
c70af9603218        handsonsecurity/seed-ubuntu:large   "bash -c 'tail -f /...'"   13 hours ago
351b39d2b2         handsonsecurity/seed-ubuntu:large   "bash -c 'ip route ..."   13 hours ago
2587f04d586        handsonsecurity/seed-ubuntu:large   "bash -c 'ip route ..."   13 hours ago
fdf6763bae4b       handsonsecurity/seed-ubuntu:large   "bash -c 'ip route ..."   13 hours ago
Up 13 hours          server-router
root@seed-ubuntu:~/Downloads/VPN# docker exec -it c70 bash
root@c70af9603218:~/# tun.py
bash: ./tun.py: command not found
root@c70af9603218:~/# ed ./volumes
root@c70af9603218:~/volumes# tun.py
Interface Name: tun1
^Ctraceback (most recent call last):
  File "./tun.py", line 24, in <module>
KeyboardInterrupt

root@c70af9603218:~/volumes# tun.py
Interface Name: tun1
[]
```

Task 2.c: Read from the TUN Interface

Task 2.d: Write to the TUN Interface

```
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN    = 0x0001
IFF_TAP    = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'lakshmi%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        ip = IP(packet)
        print(ip.summary())
        # Send out a spoof packet using the tun interface
        newip = IP(src='192.168.53.3', dst=ip.src)
        newpkt = newip/ip.payload
        os.write(tun, bytes(newpkt))
```

```
root@cf70af9603218:~# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.3: icmp_seq=6 ttl=64 time=3.81 ms
64 bytes from 192.168.53.3: icmp_seq=7 ttl=64 time=3.57 ms
64 bytes from 192.168.53.3: icmp_seq=8 ttl=64 time=3.19 ms
64 bytes from 192.168.53.3: icmp_seq=9 ttl=64 time=3.29 ms
64 bytes from 192.168.53.3: icmp_seq=10 ttl=64 time=3.31 ms
64 bytes from 192.168.53.3: icmp_seq=11 ttl=64 time=3.35 ms
64 bytes from 192.168.53.3: icmp_seq=12 ttl=64 time=2.59 ms
64 bytes from 192.168.53.3: icmp_seq=13 ttl=64 time=3.25 ms
64 bytes from 192.168.53.3: icmp_seq=14 ttl=64 time=3.15 ms

```

```
root@c70af9603218:/# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
[]
```

```
ssh.cloud.google.com/v2/ssh/projects/seed-ubuntu-410600/zones/us-central1-a/instances/seed-ubuntu?authuser=0&hl=en_US&projectNumber=633632528987&useAd...  
SSH-in-browser  
TX packets 8 bytes 652 (652.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
root@cf70af9603212:/# tcpdump -i lkhsm0 -n  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on lkhsm0, link-type RAW (Raw IP), capture size 262144 bytes  
04:35:39.305736 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 36, length 64  
04:35:39.386972 IP [loop]  
04:35:40.409741 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 37, length 64  
04:35:40.411130 IP [loop]  
04:35:41.433729 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 38, length 64  
04:35:41.437405 IP [loop]  
04:35:42.457760 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 39, length 64  
04:35:42.459022 IP [loop]  
04:35:43.481693 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 40, length 64  
04:35:43.483091 IP [loop]  
04:35:44.505707 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 41, length 64  
04:35:44.506711 IP [loop]  
04:35:45.529725 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 42, length 64  
04:35:45.530694 IP [loop]  
04:35:46.553797 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 43, length 64  
04:35:46.554809 IP [loop]  
04:35:47.577756 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 44, length 64  
04:35:47.578790 IP [loop]  
04:35:48.601763 IP 192.168.53.99 > 192.168.53.1: ICMP echo request, id 21, seq 45, length 64
```

Task 2 -Final code:

```
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'lakshmi%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        ip = IP(packet)
        print(ip.summary())
        # Send out a spoof packet using the tun interface
        newip = IP(src='192.168.53.3', dst=ip.src)
        newpkt = newip/ip.payload
        arb_data=b'Any arbitrary data'
        os.write(tun, arb_data)
```

Task 3: Send the IP Packet to VPN Server Through a Tunnel

A screenshot of a web-based terminal interface. At the top, there are two circular status indicators: one orange and one green, followed by the URL 'ssh.cloud.google.com/v2/ssh/projects/seed-ubuntu-410600/zones/us-central1-a/instances/seed-ubuntu?authuser=0&hl=en_US&projectNumber=633632528987&useAdminProx...'. Below the URL is a toolbar with icons for upload (blue arrow), download (green arrow), copy (ctrl+c), and settings (gear). The main area is titled 'SSH-in-browser' and shows a root shell prompt: 'root@c70af9603218:~# ping 192.168.53.3'. The response 'PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.' is visible below the prompt.



The screenshot shows an SSH session running in a browser window titled "SSH-in-browser". The session is connected to a root user on a host with IP address 192.168.60.5. The user has run a ping command to the same IP address, resulting in a 100% packet loss. The browser interface includes standard navigation controls (back, forward, search) and file transfer buttons for "UPLOAD FILE" and "DOWNLOAD FILE".

```
root@c70af9603218:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
21 packets transmitted, 0 received, 100% packet loss, time 20484ms

root@c70af9603218:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
```

tun_client code

```
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'lakshmi\0', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Create UDP socket
SERVER_IP='10.9.0.11'
SERVER_PORT=9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        # Send the packet via the tunnel
        sock.sendto(packet, (SERVER_IP, SERVER_PORT))
```

tun_server code

```
#!/usr/bin/env python3
from scapy.all import *
IP_A = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))
while True:
    data, (ip, port) = sock.recvfrom(2048)
    print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
    pkt = IP(data)
    print(" Inside: {} --> {}".format(pkt.src, pkt.dst))
```

Task 4: Set Up the VPN Server

```
root@c70af9603218:# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.

root@c70af9603218:/volumes# tun_client.py
Interface Name: lakshmi0

root@dfd6763bae4b:# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
19:39:07.047475 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 34, seq 11, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 11, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 12, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 12, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 13, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 13, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 14, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 14, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 15, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 15, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 16, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 16, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 17, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 17, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 18, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 18, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 19, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 19, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 20, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 20, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 21, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 21, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 22, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 22, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 23, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 23, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 24, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 24, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo request, id 34, seq 25, length 64
19:39:07.047486 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 34, seq 25, length 64

root@2857f040d586:# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
19:28:41.530859 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 5, length 64
19:28:41.530892 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 5, length 64
19:28:42.554939 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 6, length 64
19:28:42.554968 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 6, length 64
19:28:42.685662 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
19:28:42.686065 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
19:28:42.686093 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
19:28:42.686095 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
19:28:43.578774 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 7, length 64
19:28:43.578806 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 7, length 64
19:28:44.602584 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 8, length 64
19:28:44.602627 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 8, length 64
19:28:45.626632 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 9, length 64
19:28:45.626668 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 9, length 64
19:28:46.650663 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 10, length 64
19:28:46.650698 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 10, length 64
19:28:47.674597 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 11, length 64
19:28:47.674623 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 11, length 64
19:28:48.698619 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 12, length 64
19:28:48.698726 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 12, length 64
19:28:49.722699 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 13, length 64
19:28:49.722735 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 13, length 64
19:28:50.746764 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 14, length 64
19:28:50.746802 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 14, length 64
19:28:51.770876 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 15, length 64
19:28:51.770908 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 15, length 64
19:28:52.794601 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 16, length 64
19:28:52.794634 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 16, length 64
19:28:53.818821 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 17, length 64
19:28:53.818825 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 17, length 64
19:28:54.843191 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 18, length 64
19:28:54.843227 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 18, length 64
19:28:55.866755 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 19, length 64
19:28:55.866788 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 19, length 64
19:28:56.890842 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 20, length 64
19:28:56.890875 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 20, length 64
19:28:57.914522 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 21, length 64
19:28:57.914553 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 21, length 64
19:28:58.938608 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 22, length 64
19:28:58.938644 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 22, length 64
19:28:59.962661 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 23, length 64
19:28:59.962697 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 23, length 64
19:29:00.986927 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 24, length 64
19:29:00.986961 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 33, seq 24, length 64
19:29:02.010644 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 33, seq 25, length 64
```

tun_server code

```
#!/usr/bin/env python3
import fcntl
import struct
import os
import time
from scapy.all import *
TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'lakshmi%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.50/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

IP_A = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))
while True:
    data, (ip, port) = sock.recvfrom(2048)
    print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
    pkt = IP(data)
    print(" Inside: {} --> {}".format(pkt.src, pkt.dst))
    os.write(tun,bytes(pkt))
```

Tun_client code

```
#!/usr/bin/env python3
import fcntl
import struct
import os
import time
from scapy.all import *
TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'lakshmi%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))
# Create UDP socket
SERVER_IP='10.9.0.11'
SERVER_PORT=9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        # Send the packet via the tunnel
        sock.sendto(packet, (SERVER_IP, SERVER_PORT))
```

Task 5: Handling Traffic in Both Directions

The image shows four browser windows side-by-side, each displaying a terminal session via SSH-in-browser.

- Top Left Window:** Shows a ping session from host 192.168.60.5 to 192.168.60.5. The output shows ICMP sequence numbers 1 through 25 being sent and received.
- Top Right Window:** Shows a similar ping session, but with a different timestamp for each packet, indicating round-trip times.
- Middle Left Window:** Shows a command being executed in a running container on host 192.168.60.5. The command is `root@seed-ubuntu:/home/lakshmiswaminathan0# docker exec -it c70a... bash` followed by `root@70af9603218:/# cd ./volumes` and `root@70af9603218:/# ./tun_client.py`.
- Middle Right Window:** Shows a `tcpdump` session on interface eth0. It captures ICMP echo requests and replies, showing the flow of traffic between the two hosts.
- Bottom Left Window:** Shows another command being executed in a container on host 192.168.60.5. The command is `root@70af9603218:/# ./tun_server.py`.
- Bottom Right Window:** Shows a `tcpdump` session on interface eth0, capturing ICMP echo requests and replies, mirroring the traffic seen in the middle right window.
- Bottom Center Window:** Shows a terminal session on host 410600, listing Docker containers and their status. It includes commands like `docker ps` and `telnet 192.168.60.5`.

Task 6: Tunnel-Breaking Experiment

Telnet connection established between client and host within a private network(192.168.60.5)

The image shows two separate browser windows, both titled "ssh.cloud.google.com/v2/ssh". Each window has a "SSH-in-browser" tab and includes standard browser navigation controls (back, forward, search, etc.).

Left Window Terminal Output:

```
lakshmini@lakshminathan08:~$ sudo su
root@seed-ubuntu:/home/lakshmini@lakshminathan08# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
c70af9603218        handsonsecurity/seed-ubuntu:large   "bash -c ' tail -f ..."  34 hours ago
358bf33d5e65        handsonsecurity/seed-ubuntu:large   "bash -c ' ip route ..."  34 hours ago
ago 35 minutes
ago 35 minutes
2587f040d586        handsonsecurity/seed-ubuntu:large   "bash -c ' ip route ..."  34 hours ago
ago 35 minutes
ago 35 minutes
dfd6763bae4b       handsonsecurity/seed-ubuntu:large   "bash -c ' ip route ..."  34 hours ago
ago 35 minutes
ago 35 minutes
root@seed-ubuntu:/home/lakshmini@lakshminathan08# docker exec -it c70 bash
root@c70af9603218:~# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
2587f040d586 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-1052-gcp x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Fri Mar 1 00:10:57 UTC 2024 from 192.168.53.99 on pts/3
seed@2587f040d586:~$ pwd
/home/seed
seed@2587f040d586:~$
```

Right Window Terminal Output:

```
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
[...]
[Truncated]
KeyboardInterrupt
root@c70af9603218:~$ volumes#
root@c70af9603218:~$
```

Bottom Status Bar:

Debugging in the... Dropbox - The Dat... My Dashboard - T... Learn Data Structu... All Bookmarks

VPN Tunnel is broken by stopping tun_client/tun_server for a while and the telnet connection is frozen. Column't type any cmds

```

lakshminathan08@seed-ubuntu:~$ sudo su
root@seed-ubuntu:/home/lakshminathan08# docker ps
CONTAINER ID IMAGE STATUS PORTS NAMES
c70af9603218 handsonsecurity/seed-ubuntu:large "bash -c ' tail -f /_'" 34 hours ago Up 35 minutes client-10.9.0.5
3511b9d2b253 handsonsecurity/seed-ubuntu:large "bash -c ' ip route _'" 34 hours ago Up 35 minutes host-192.168.60.6
2307f10383c0 handsonsecurity/seed-ubuntu:large "bash -c ' ip route _'" 34 hours ago Up 35 minutes host-192.168.60.5
dfdf6763bae4b handsonsecurity/seed-ubuntu:large "bash -c ' ip route _'" 34 hours ago Up 35 minutes server-router
root@seed-ubuntu:/home/lakshminathan08# docker exec -it c70 bash
root@c70af9603218:~# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^}'.
Ubuntu 20.04.1 LTS
2587f0404d586 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-1052-gcp x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Fri Mar 1 00:10:57 UTC 2024 from 192.168.53.99 on pts/3
seed@2587f0404d586:~$ pwd
/home/seed
seed@2587f0404d586:~$ hdbuioebiewjpoed

```



```

From socket <=: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <=: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
^Traceback (most recent call last):
  File "./tun_client.py", line 31, in <module>
    ready, _ = select.select([sock, tun], [], [])
KeyboardInterrupt

```



```

root@c70af9603218:/volumes# tun_client.py
Interface Name: lakshmi0
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket <=: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5

```



```

From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <=: 192.168.53.99 --> 192.168.60.5
From socket <=: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <=: 192.168.53.99 --> 192.168.60.5
From socket <=: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <=: 192.168.53.99 --> 192.168.60.5
From socket <=: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
^Traceback (most recent call last):
  File "./tun_server.py", line 31, in <module>
    ready, _ = select.select([sock, tun], [], [])
KeyboardInterrupt

```



```

root@dfdf6763bae4b:/volumes# tun_server.py
Interface Name: lakshmi0
From socket <=: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <=: 192.168.53.99 --> 192.168.60.5

```

Once the connection is revived having the tunnel broken for a while, the text that was typed when the connection is broken is showing up.

Task 7: Routing Experiment on Host V

Default route for 192.168.60.5 is 192.168.60.11- Host u(client) is successfully able to ping host v.

After deleting the default route, connection broke between client and hosts in private network

Connection between client and private network is preserved once the default route is added back.