Lab Assignments
Network Security
UCS727

Submitted to:
Dr. Navneet Sharma

Submitted by:
Hardik Dhamija 101615049

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Deemed to be University)
PATIALA-147001
PUNJAB
INDIA

Aug-Dec 2019

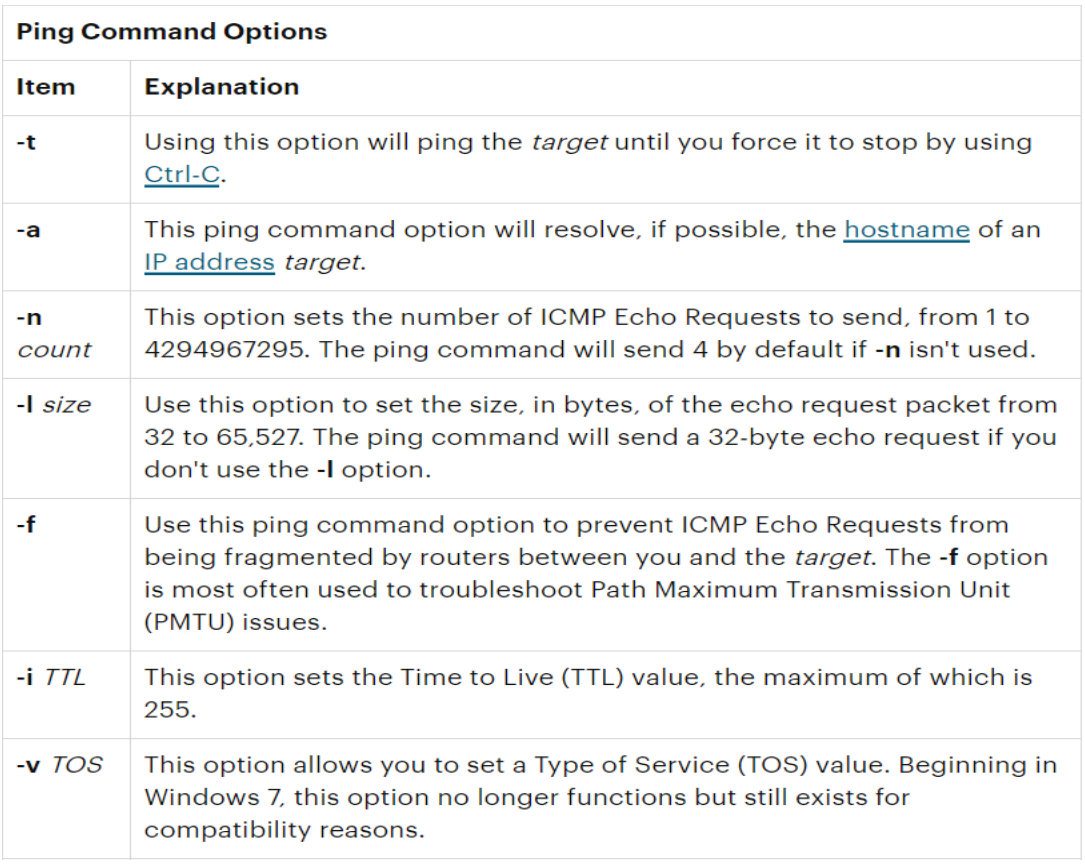
Experiment No.1

Aim- Use the ping command to ping different ip address and check different options associated

Theory -

Ping is a tool that sends test packets through the network to a destination of your choice and measures the response time.
Each line has several components with the following meaning:
>PING ninefortwo.be (185.18.9.249)

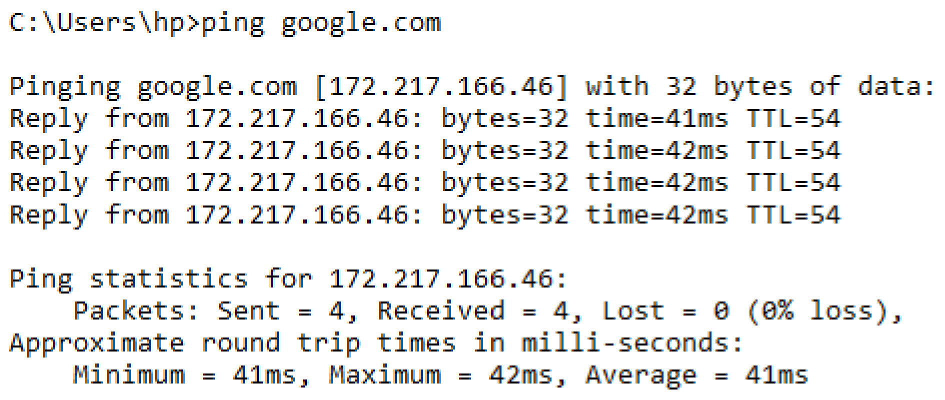
- 56 data bytes – If you ping a domain name, before the test, the tool will identify that domain name's IP address.
- time=43.488 ms – This is the amount of time it took for 1 test packet to be sent from your device, reach the destination and reach you back.
- 64 bytes – this is the size of the test packet
- ttl=54 – this is your test's 'time to live'. The TTL defines for how long the packet will travel before it dies
- 8 packets transmitted, 8 packets received, 0.0% packet loss – The amount of sent and received packets, and % packet loss.



Ping Command Options

Item	**Explanation**
-t	Using this option will ping the *target* until you force it to stop by using Ctrl-C.
-a	This ping command option will resolve, if possible, the hostname of an IP address *target*.
-n *count*	This option sets the number of ICMP Echo Requests to send, from 1 to 4294967295. The ping command will send 4 by default if **-n** isn't used.
-l *size*	Use this option to set the size, in bytes, of the echo request packet from 32 to 65,527. The ping command will send a 32-byte echo request if you don't use the **-l** option.
-f	Use this ping command option to prevent ICMP Echo Requests from being fragmented by routers between you and the *target*. The **-f** option is most often used to troubleshoot Path Maximum Transmission Unit (PMTU) issues.
-i *TTL*	This option sets the Time to Live (TTL) value, the maximum of which is 255.
-v *TOS*	This option allows you to set a Type of Service (TOS) value. Beginning in Windows 7, this option no longer functions but still exists for compatibility reasons.

Results-



```
C:\Users\hp>ping google.com

Pinging google.com [172.217.166.46] with 32 bytes of data:
Reply from 172.217.166.46: bytes=32 time=41ms TTL=54
Reply from 172.217.166.46: bytes=32 time=42ms TTL=54
Reply from 172.217.166.46: bytes=32 time=42ms TTL=54
Reply from 172.217.166.46: bytes=32 time=42ms TTL=54

Ping statistics for 172.217.166.46:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 41ms, Maximum = 42ms, Average = 41ms
```

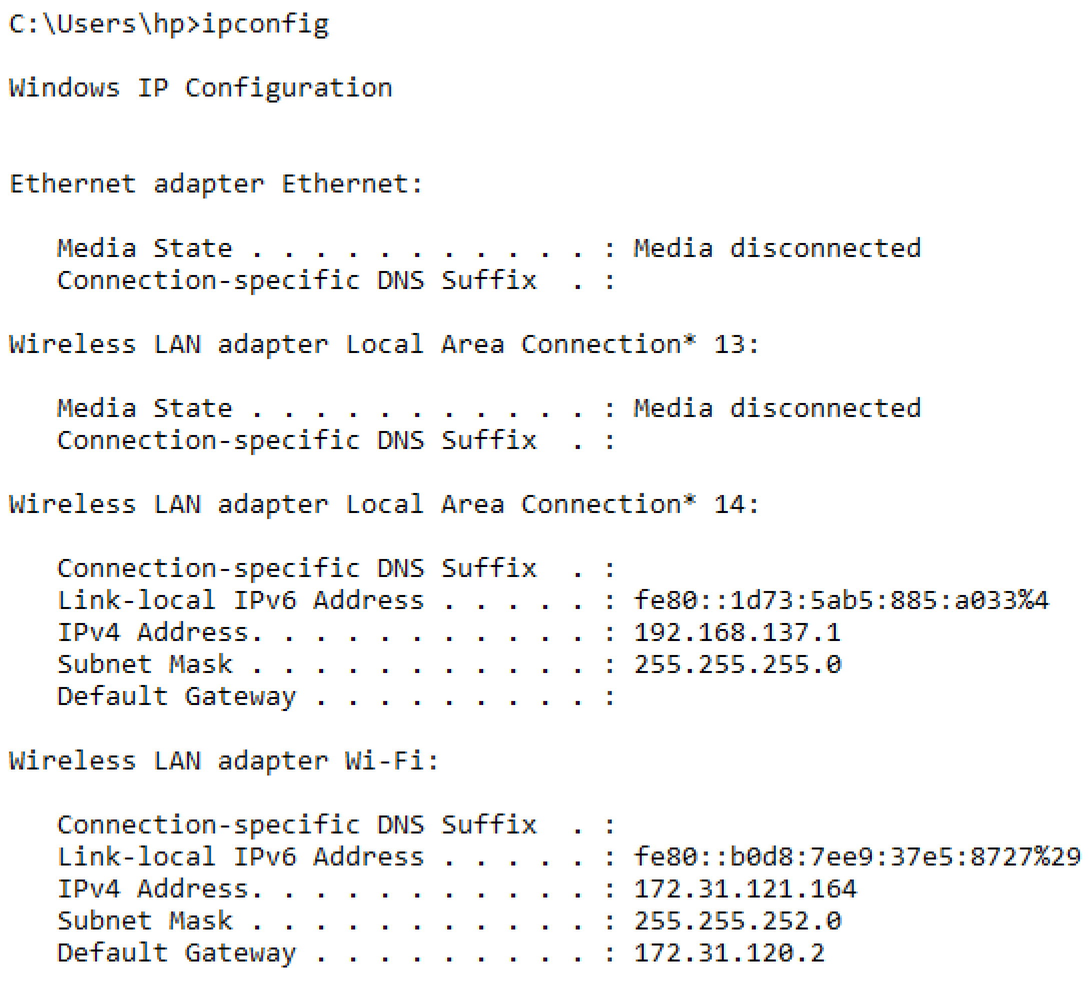
Experiment No.2

Aim- Check Ipconfig command and study the options.

Theory -

Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Used without parameters, ipconfig displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.

Output-



```
C:\Users\hp>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 13:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 14:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::1d73:5ab5:885:a033%4
   IPv4 Address. . . . . . . . . . . : 192.168.137.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b0d8:7ee9:37e5:8727%29
   IPv4 Address. . . . . . . . . . . : 172.31.121.164
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 172.31.120.2
```

Experiment No.3

Aim-Study the NS lookup and net stat command.

Theory -

NS lookup: Displays information that you can use to diagnose Domain Name System (DNS) infrastructure. Before using this tool, you should be familiar with how DNS works. The nslookup command-line tool is available only if you have installed the TCP/IP protocol.

Net stat: Netstat derived from the words *network* and *statistics* is a program that's controlled via commands issued in the command line. It delivers basic statistics on all network activities and informs users on which ports and addresses the corresponding connections (TCP, UDP) are running and which ports are open for tasks.

Results:

```
C:\Users\hp>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:1470         LAPTOP-3L32685D:3080   TIME_WAIT
  TCP    127.0.0.1:1544         LAPTOP-3L32685D:1545   ESTABLISHED
  TCP    127.0.0.1:1545         LAPTOP-3L32685D:1544   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1579   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1584   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1585   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1586   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1587   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1594   ESTABLISHED
  TCP    127.0.0.1:1561         LAPTOP-3L32685D:1640   ESTABLISHED
```

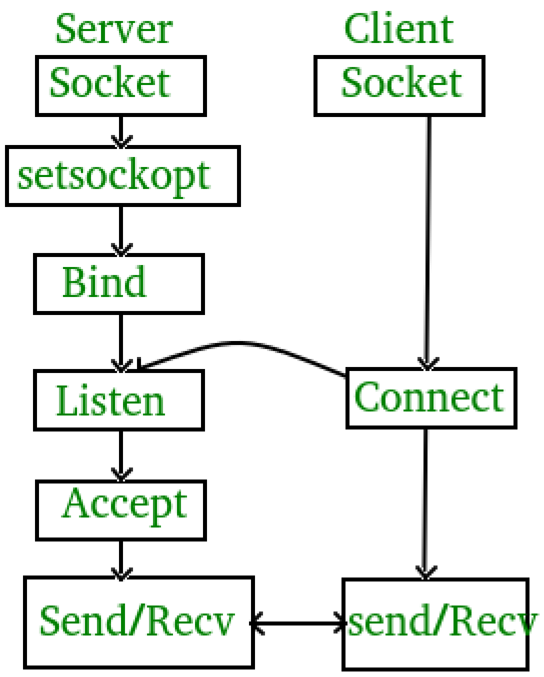
```
C:\Users\hp>nslookup
Default Server:  gateway.thapar.edu
Address:  172.31.1.6

> facebook.com
Server:  gateway.thapar.edu
Address:  172.31.1.6

Non-authoritative answer:
Name:    facebook.com
Addresses:  2a03:2880:f12f:83:face:b00c:0:25de
          31.13.79.35
```

Experiment No.4

Aim- Implement socket programming.

Theory-

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.



Code:

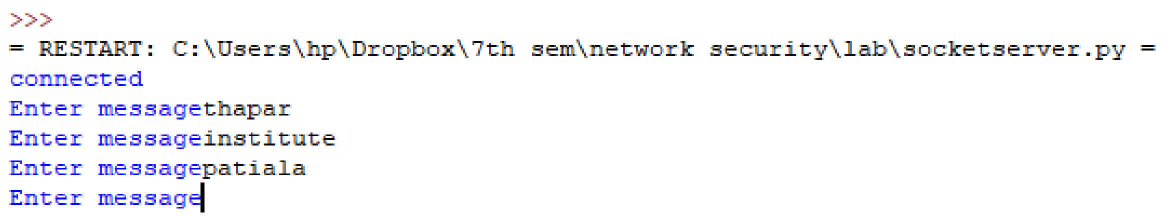
Server-

```
import socket
s = socket.socket()
port = 1471
s.bind(('', port))
s.listen(1)
c, addr = s.accept()
print("connected")
while 1:
   a=input('Enter message')
   c.send(a.encode())
```

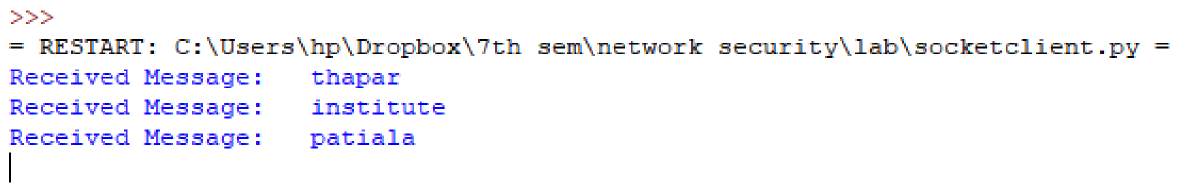
Client:

```
import socket
s = socket.socket()
port = 1471
s.connect(('127.0.0.1', port))
while 1:
    print("Received Message:  ",s.recv(1024).decode())
```

Output:



```
>>>
= RESTART: C:\Users\hp\Dropbox\7th sem\network security\lab\socketserver.py =
connected
Enter messagethapar
Enter messageinstitute
Enter messagepatiala
Enter message
```



```
>>>
= RESTART: C:\Users\hp\Dropbox\7th sem\network security\lab\socketclient.py =
Received Message:   thapar
Received Message:   institute
Received Message:   patiala
```

Experiment No.5

Aim-Implement following Substitution & Transportation technique

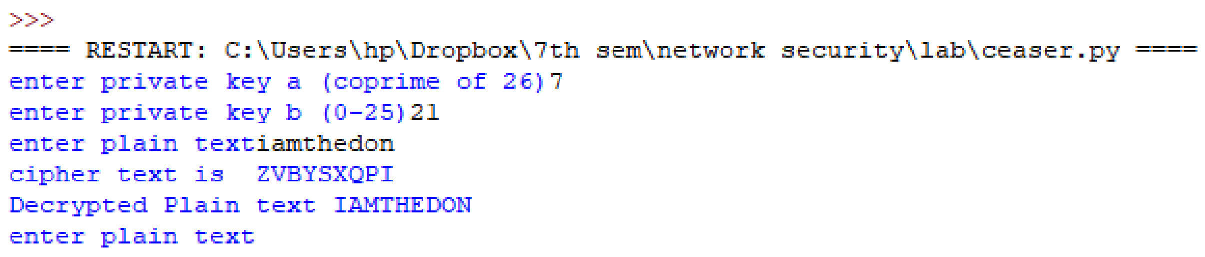
a) Caesar cipher   b) Playfair cipher

Code-

a) Caesar cipher

```
m=26
a=int(input('enter private key a (coprime of 26)'))
b=int(input('enter private key b (0-25)'))
while 1:
    plntxt=input('enter plain text').upper()
    cphrtxt="
    for i in plntxt:
        cphrtxt+=chr(((ord(i)-ord('A'))*a+b)%m+ord('A'))
    print("cipher text is ",cphrtxt)
    dtxt="
    ainv=0
    for i in range(1,26):
        if (a*i)%m==1:
            ainv=i
            break
    for i in cphrtxt:
        dtxt+=chr((((ord(i)-ord('A'))-b)*ainv)%m+ord('A'))
    print("Decrypted Plain text",dtxt)
```

Output:

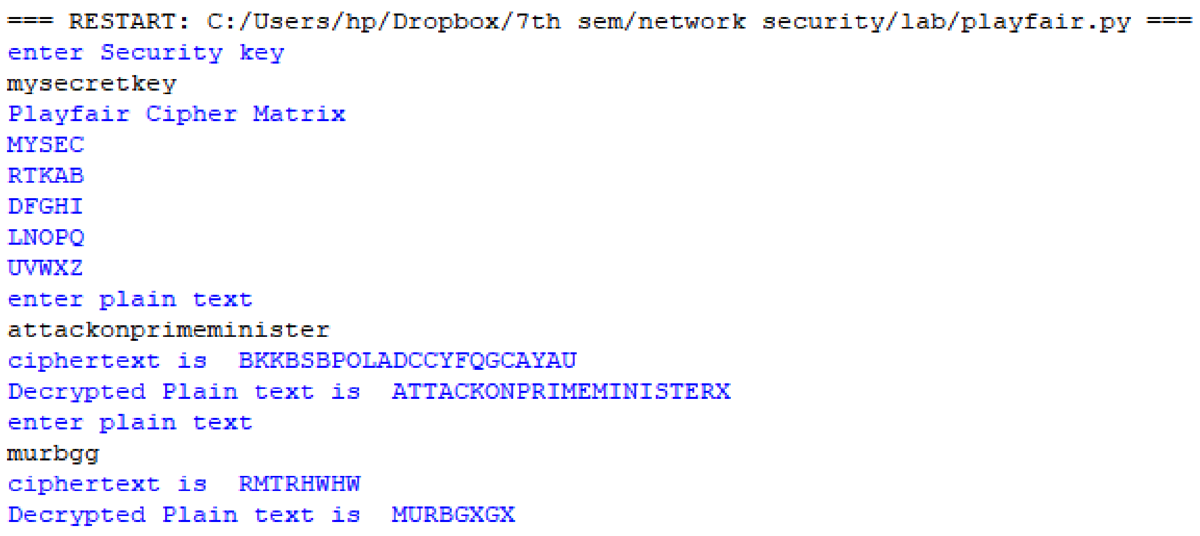


```
>>>
==== RESTART: C:\Users\hp\Dropbox\7th sem\network security\lab\ceaser.py ====
enter private key a (coprime of 26)7
enter private key b (0-25)21
enter plain textiamthedon
cipher text is  ZVBYSXQPI
Decrypted Plain text IAMTHEDON
enter plain text
```

b) Playfair cipher

```
key=input('enter Security key\n').upper()
a={chr(i):0 for i in range(65,65+26)}
#print(a)
a['J']=1
p=''
for i in key:
    if a[i]==0:
        p+=i
        a[i]=1
for i in range(65,65+26) :
    if a[chr(i)]==0:
        p+=chr(i)
print("Playfair Cipher Matrix")
for i in range(5,26,5):
    print(p[i-5:i])
while 1:
    #encryption
    plntxt=input('enter plain text\n').upper()
    plntxt=plntxt.replace('J','I')
    cphrtxt=''
    while plntxt!='':
        pln=plntxt[:2]
        plntxt=plntxt[2:]
        if(len(pln)==1):
            pln+='X'
        elif(pln[0]==pln[1]!='X'):
            plntxt=pln[1]+plntxt
            pln=pln[0]+'X'
        r1=int(p.find(pln[0])/5)
        c1=p.find(pln[0])%5
        r2=int(p.find(pln[1])/5)
        c2=p.find(pln[1])%5
        if(r1==r2):
            c1=(c1+1)%5
            c2=(c2+1)%5
        elif(c1==c2):
            r1=(r1+1)%5
            r2=(r2+1)%5
        else:
            tmp=c1
            c1=c2
            c2=tmp
        cphrtxt+=p[r1*5+c1]+p[r2*5+c2]
    print("ciphertext is ",cphrtxt)
    #decryption
    dec=''
    while cphrtxt!='':
        pln=cphrtxt[:2]
        cphrtxt=cphrtxt[2:]
        r1=int(p.find(pln[0])/5)
        c1=p.find(pln[0])%5
        r2=int(p.find(pln[1])/5)
        c2=p.find(pln[1])%5
        if(r1==r2):
            c1=(c1-1)%5
            c2=(c2-1)%5
        elif(c1==c2):
            r1=(r1-1)%5
            r2=(r2-1)%5
        else:
            tmp=c1
            c1=c2
            c2=tmp
        dec+=p[r1*5+c1]+p[r2*5+c2]
    print('Decrypted Plain text is ',dec)
```

Output:



```
=== RESTART: C:/Users/hp/Dropbox/7th sem/network security/lab/playfair.py ===
enter Security key
mysecretkey
Playfair Cipher Matrix
MYSEC
RTKAB
DFGHI
LNOPQ
UVWXZ
enter plain text
attackonprimeminister
ciphertext is  BKKBSBPOLADCCYFQGCAYAU
Decrypted Plain text is  ATTACKONPRIMEMINISTERX
enter plain text
murbgg
ciphertext is  RMTRHWHW
Decrypted Plain text is  MURBGXGX
```

Experiment No.6

Aim-Implement following Algorithms:

a) DES   b) RSA

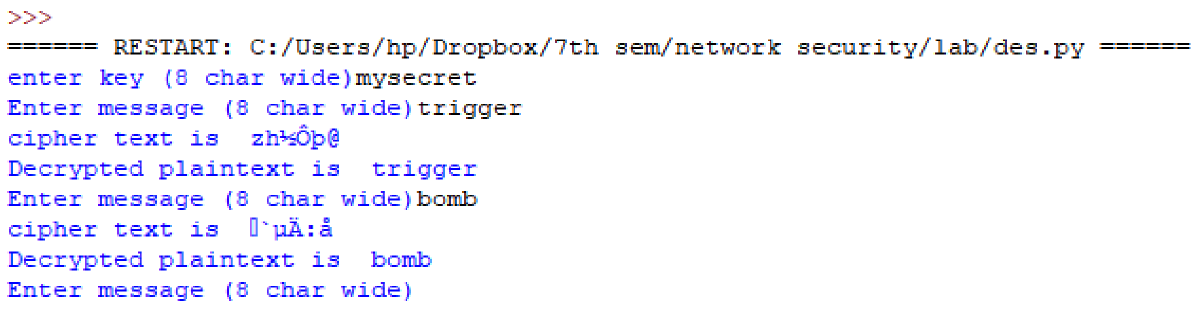
Code-

a) DES

```
k=input("enter key (8 char wide)")[:8]
def strtoint(s):
    res=0
    for i in s:
        res=(res<<8)+ord(i)
    return res
def inttostr(s):
    res=''
    c=(1<<8)-1
    while s>0:
        res=chr(s&c)+res
        s=s>>8
    return res
k=strtoint(k)
while 1:
    plntxt=input("Enter message (8 char wide)")[:8]
    plntxt=strtoint(plntxt)
    kn=k
    for i in range(1,17):
        a=plntxt>>32
        b=plntxt-(a<<32)
        kn=((k<<i)&((1<<64)-1))>>32
        tmp=a^b^kn
        a=b
        b=tmp
        plntxt=(a<<32)+b
    print("cipher text is ",inttostr(plntxt))
    kn=k
    for i in range(16,0,-1):
        a=plntxt>>32
        b=plntxt-(a<<32)
        kn=((k<<i)&((1<<64)-1))>>32
        tmp=a^b^kn
        b=a
        a=tmp
        plntxt=(a<<32)+b
```

```
print("Decrypted plaintext is ",inttostr(plntxt))
```

Output:

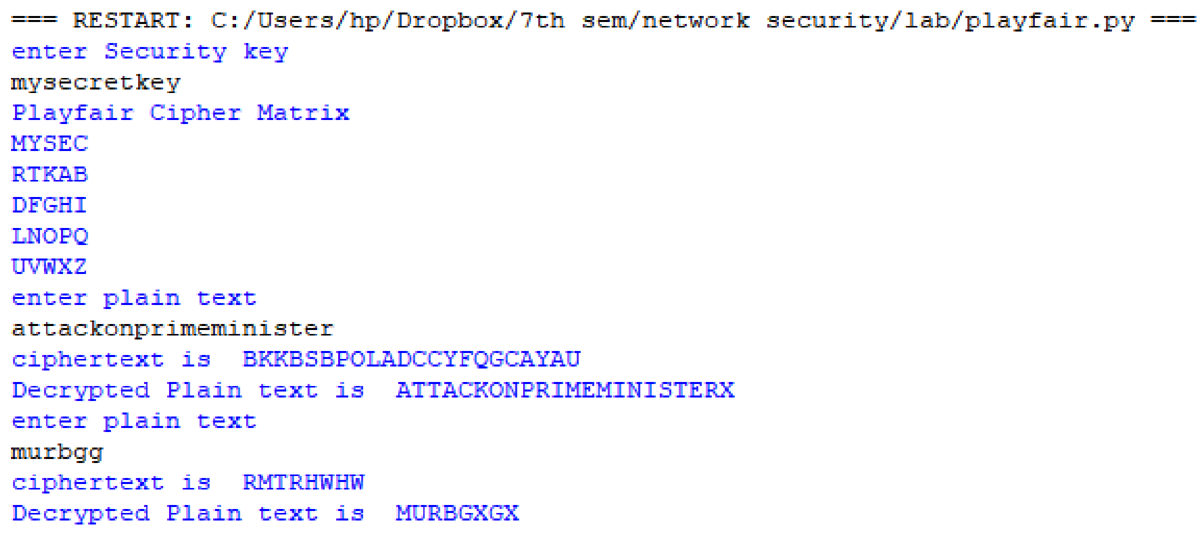


```
>>>
====== RESTART: C:/Users/hp/Dropbox/7th sem/network security/lab/des.py ======
enter key (8 char wide)mysecret
Enter message (8 char wide)trigger
cipher text is  zh½Ôp@
Decrypted plaintext is  trigger
Enter message (8 char wide)bomb
cipher text is  ▯`µÄ:å
Decrypted plaintext is  bomb
Enter message (8 char wide)
```

b) RSA

```
p=int(input('Enter p\n'))
q=int(input('Enter q\n'))
n=p*q
nn=(p-1)*(q-1)
e=0
for i in range(2,nn):
    rem=i
    prem=nn
    while rem>0:
        trem=prem%rem
        prem=rem
        rem=trem
    if prem==1:
        e=i
        break
d=0
for i in range(1,nn):
    if (i*e)%nn==1:
        d=i
        break
print('p={}, q={}, n={}, phi={}, e={}, d={}'.format(p,q,n,nn,e,d))
while 1:
    pln=int(input('Enter data\n'))
    cphr=pow(pln,e,n)
    print('Encrypted Data: ',cphr)
    desc=pow(cphr,d,n)
    print('Decrypted Data: ',desc)
```

Output:



```
=== RESTART: C:/Users/hp/Dropbox/7th sem/network security/lab/playfair.py ===
enter Security key
mysecretkey
Playfair Cipher Matrix
MYSEC
RTKAB
DFGHI
LNOPQ
UVWXZ
enter plain text
attackonprimeminister
ciphertext is  BKKBSBPOLADCCYFQGCAYAU
Decrypted Plain text is  ATTACKONPRIMEMINISTERX
enter plain text
murbgg
ciphertext is  RMTRHWHW
Decrypted Plain text is  MURBGXGX
```

Experiment No.7

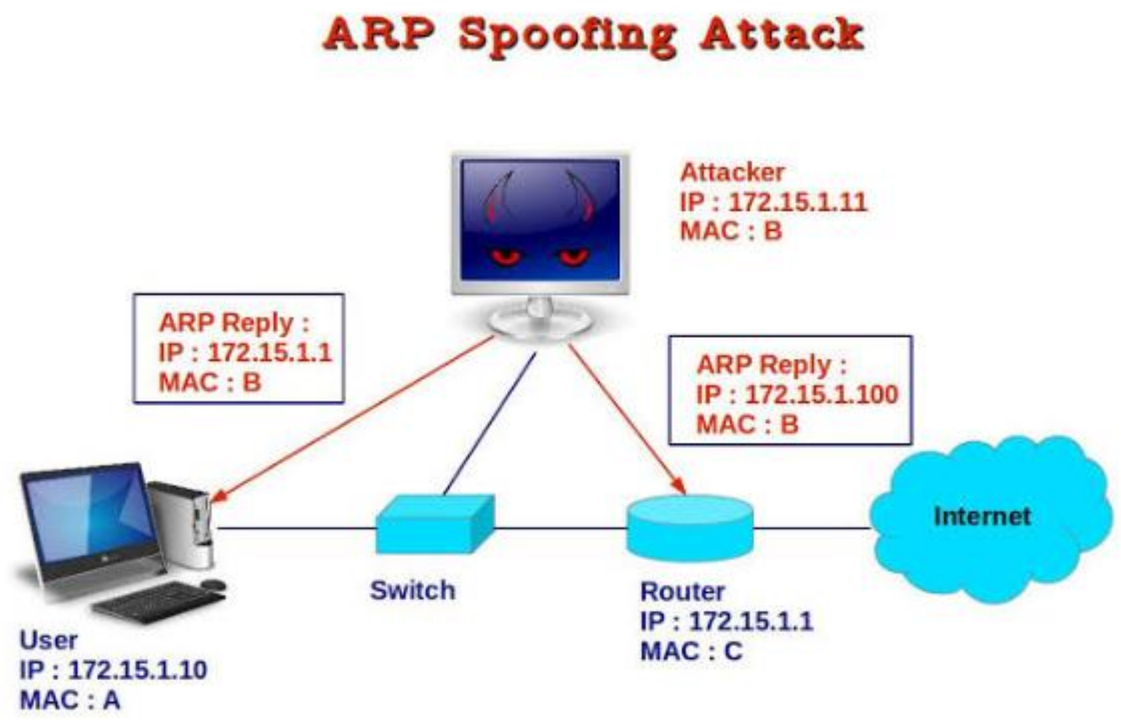
Aim-Study and Perform the ARP spoofing attack

Theory-

ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address. ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. ARP spoofing attacks can only occur on local area networks that utilize the Address Resolution Protocol.

Here is how ARP works –

- When one machine needs to communicate with another, it looks up its ARP table.
- If the MAC address is not found in the table, the ARP_request is broadcasted over the network.
- All machines on the network will compare this IP address to its IP address.
- If one of the machines in the network identifies this address, then it will respond to the ARP_request with its IP and MAC address.
- The requesting computer will store the address pair in its ARP table and communication will take place.



Experiment No.8

Aim-Perform and analuze DOS attack using Ping.

Theory-

A denial-of-service (DoS) attack is a type of cyber attack in which a malicious actor aims to render a computer or other device unavailable to its intended users by interrupting the device's normal functioning. DoS attacks typically function by overwhelming or flooding a targeted machine with requests until normal traffic is unable to be processed, resulting in denial-of-service to addition users. A DoS attack is characterized by using a single computer to launch the attack.
The primary focus of a DoS attack is to oversaturate the capacity of a targeted machine, resulting in denial-of-service to additional requests. The multiple attack vectors of DoS attacks can be grouped by their similarities.
DoS attacks typically fall in 2 categories:

Buffer overflow attacks

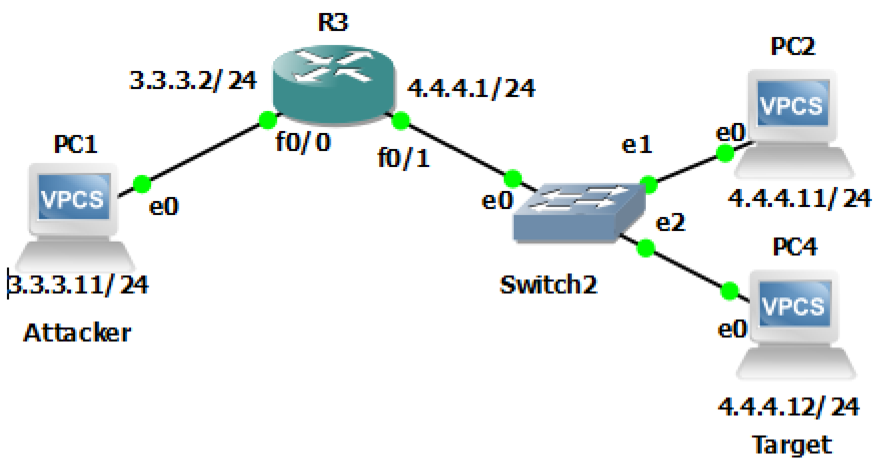
An attack type in which a memory buffer overflow can cause a machine to consume all available hard disk space, memory, or CPU time. This form of exploit often results in sluggish behavior, system crashes, or other deleterious server behaviors, resulting in denial-of-service.

Flood attacks

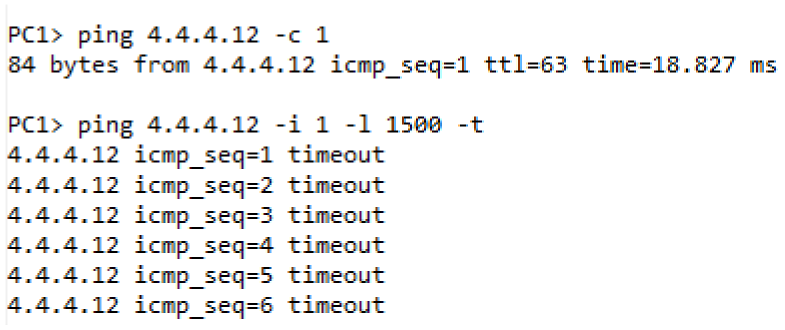
By saturating a targeted server with an overwhelming amount of packets, a malicious actor is able to oversaturate server capacity, resulting in denial-of-service. In order for most DoS flood attacks to be successful, the malicious actor must have more available bandwidth than the target.

Performing DOS attack (Buffer Overflow):

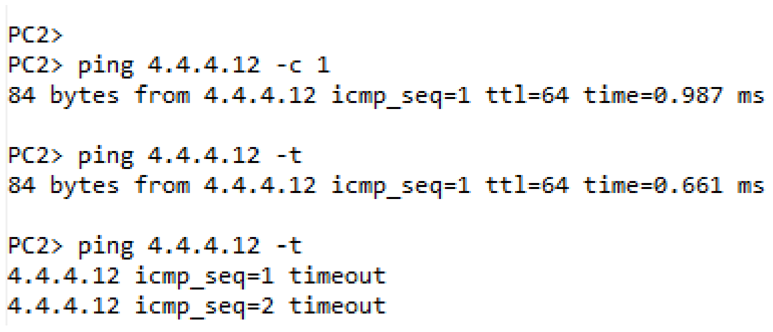
A Machine from the network sends ping requests with large packet size, which result in buffer overflow in the target machine and the machine crashes.



- PC1 sends a packet with large size to PC4.
- PC4 crashes due to buffer overflow.
- This results in Denial of service to all machines ex: PC2.



```
PC1> ping 4.4.4.12 -c 1
84 bytes from 4.4.4.12 icmp_seq=1 ttl=63 time=18.827 ms

PC1> ping 4.4.4.12 -i 1 -l 1500 -t
4.4.4.12 icmp_seq=1 timeout
4.4.4.12 icmp_seq=2 timeout
4.4.4.12 icmp_seq=3 timeout
4.4.4.12 icmp_seq=4 timeout
4.4.4.12 icmp_seq=5 timeout
4.4.4.12 icmp_seq=6 timeout
```



```
PC2>
PC2> ping 4.4.4.12 -c 1
84 bytes from 4.4.4.12 icmp_seq=1 ttl=64 time=0.987 ms

PC2> ping 4.4.4.12 -t
84 bytes from 4.4.4.12 icmp_seq=1 ttl=64 time=0.661 ms

PC2> ping 4.4.4.12 -t
4.4.4.12 icmp_seq=1 timeout
4.4.4.12 icmp_seq=2 timeout
```