PRACTICAL 1(INS)

Caesar Cipher:

Code:

```
import java.util.Scanner; public class CaesarCipher{     public static
final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";     public
static String encrypt(String plaintext,int shiftkey)
    {
        plaintext = plaintext.toLowerCase();
String ciphertext = "";         for (int i = 0;
i<plaintext.length();i++)
        {             int charposition =
ALPHABET.indexOf(plaintext.charAt(i));             int keyval =
(shiftkey+charposition)%26;             char replaceval =
ALPHABET.charAt(keyval);             ciphertext+=replaceval;
        }         return
ciphertext;
    }     public static String decrypt(String ciphertext, int
shiftkey)
    {
        ciphertext=ciphertext.toLowerCase();
String plaintext="";         for (int
i=0;i<ciphertext.length();i++)
        {             int charposition =
ALPHABET.indexOf(ciphertext.charAt(i));             int keyval =
(charposition-shiftkey)%26;             if (keyval<0)
            {
keyval=ALPHABET.length()+keyval;
            }             char
replaceval=ALPHABET.charAt(keyval);
plaintext+=replaceval;
        }         return
plaintext;
    }
    public static void main(String[]args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string for encryption: ");
String msg = new String();         msg = sc.next();
        System.out.println(encrypt(msg, 3));
        System.out.println(decrypt(encrypt(msg, 3),3));
sc.close();
    }
}
```

Output:

```
Enter the string for encryption:
CaesarCipher
fdhvduflskhu
caesarcipher
```

Monoalphabetic:

Code:

```
import java.util.Scanner;
public class monocipher {
public static char     p[]
=
{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v',
'w','x','y','z'};
public static char
ch[] =
{'Q','W','E','R','T','Y','U','I','O','P','A','S','D','F','G','H','J','K','L','Z','X','C',
'V','B','N','M'};     public static String
doencryption(String s){         char c[] = new
char[(s.length())];         for (int i = 0;
i<s.length();i++){             for (int j =
0;j<26;j++){                 if(p[j] ==
s.charAt(i)){                     c[i] = ch[j];
break;
                }
            }         }
return(new String(c));
    }     public static String
dodecryption(String s){         char p1[] = new
char[(s.length())];         for (int i = 0;
i<s.length();i++){             for (int j=0;
j<26;j++){                 if (ch[j] ==
s.charAt(i)){                     p1[i]=p[j];
break;
                }
            }         }
return (new String (p1));
    }
    public static void main(String[]args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a message: ");
        String en = doencryption(sc.next().toLowerCase());
        System.out.println("Encrypted Message: "+en);
        System.out.println("Decrypted Message: "+ dodecryption(en));
sc.close();
    }
}
```

Output:

```
Enter a message:
Attack
Encrypted Message: QZZQEA
Decrypted Message: attack
```

Railfence:

Code:

```
import java.util.Scanner;  public class
RailFenceCipher {   public static void
main(String[] args) {  try
{
RailFenceBasic rf=new RailFenceBasic();
Scanner sc = new Scanner(System.in);  int
depth;
String plainText,cipherText,decryptedText;
System.out.println("Enter plain Text : ");  plainText=sc.nextLine();
System.out.println("Enter Depth of Encryption : ");
depth=sc.nextInt();
cipherText=rf.Encryption(plainText,depth);
System.out.println("Encrypted Text is:\n"+cipherText);
decryptedText=rf.Decryption(cipherText,depth);
System.out.println("Decrypted Text is:\n"+decryptedText);
}catch(Exception e)
{}  }  private static class
RailFenceBasic {
//int depth;
String Encryption(String plainText, int depth)throws Exception {
int r=depth,len=plainText.length();  int c=len/depth;
if(len%depth != 0){  c = c+1;  }  char mat[][]=new char[r][c];
int k=0;
String cipherText="";  for(int
i=0;i<c;i++)
{
for(int j=0;j<r;j++){
if(k!=len)
{
mat[j][i]=plainText.charAt(k++);
} else mat[j][i]='X';
}  }  for(int
i=0;i<r;i++)
{
for(int j=0;j<c;j++)
{  cipherText+=mat[i][j];
```

```
}
}
return cipherText;
}
String Decryption(String cipherText, int depth) {
int r=depth,len=cipherText.length();
int c=len/depth;
char mat[][]=new char[r][c];
int k=0;
String plainText="";
for(int i=0;i<r;i++)
{
for(int j=0;j<c;j++)
{
mat[i][j]=cipherText.charAt(k++);
}
}
for(int i=0;i<c;i++)
{
for(int j=0;j<r;j++)
{
plainText+=mat[j][i];
}
}
return plainText;
}
}}
```

```
Enter plain Text :
Shravani
Enter Depth of Encryption :
2
Encrypted Text is:
Srvnhaai
Decrypted Text is:
Shravani
```

Simple Columnar:

Code:

jyoti > columnar.java > columnar > main(String[])

```java
import java.util.*;
public class columnar{
Run | Debug
public static void main(String sap[]){
Scanner sc = new Scanner(System.in);
System.out.print(s:"\nEnter plaintext(enter in lower case): ");
String message = sc.next();
System.out.print(s:"\nEnter key in numbers: ");
String key = sc.next();
int columnCount = key.length();
int rowCount = (message.length()+columnCount)/columnCount;
int plainText[][] = new int[rowCount][columnCount];
int cipherText[][] = new int[rowCount][columnCount];
System.out.print(s:"\n-----Encryption-----\n");
cipherText = encrypt(plainText, cipherText, message, rowCount, columnCount,key);
String ct = "";
for(int i=0; i<columnCount; i++)
{
for(int j=0; j<rowCount; j++)
{
if(cipherText[j][i] == 0)
ct = ct + 'x';
else{
ct = ct + (char)cipherText[j][i];
}
}
}
System.out.print("\nCipher Text: " + ct);
System.out.print(s:"\n\n\n-----Decryption-----\n");
plainText = decrypt(plainText, cipherText, ct, rowCount, columnCount, key);
String pt = "";
for(int i=0; i<rowCount; i++)
{
for(int j=0; j<columnCount; j++)
{
if(plainText[i][j] == 0)
pt = pt + "";
else{
pt = pt + (char)plainText[i][j];
}
}
}
System.out.print("\nPlain Text: " + pt);
System.out.println();
}
static int[][] encrypt(int plainText[][], int cipherText[][], String message, int
rowCount, int columnCount, String key){
int i,j;
int k=0;
for(i=0; i<rowCount; i++)
{
for(j=0; j<columnCount; j++)
```

```
{
if(k < message.length())
{
plainText[i][j] = (int)message.charAt(k);
k++;
}
else
{
break;
}
}
}
for(i=0; i<columnCount; i++)
{
int currentCol= ( (int)key.charAt(i) - 48 ) -1;
for(j=0; j<rowCount; j++)
{
cipherText[j][i] = plainText[j][currentCol];
}
}
System.out.print(s:"Cipher Array(read column by column): \n");
for(i=0;i<rowCount;i++){
for(j=0;j<columnCount;j++){
System.out.print((char)cipherText[i][j]+"\t");
```

Output:

```
Enter plaintext(enter in lower case): tomorrowisholiday

Enter key in numbers: 3124

-----Encryption-----
Cipher Array(read column by column):
m       t       o       o
o       r       r       w
h       i       s       o
d       l       i       a
        y
Cipher Text: mohdxtrilyorsixowoax


-----Decryption-----
Plain Array(read row by row):
t       o       m       o
r       r       o       w
r       r       o       w
i       s       h       o
l       i       d       a
y

Plain Text: tomorrowisholiday
```

PRACTICAL 2(INS) Code:

```
import math
def gcd(a,h):
    while(1):
        temp= a%h
        if(temp==0):
            return h
        a = h
        h =temp
p=3
q=7
n=p*q
e=2
phi=(p-1)*(q-1)
while(e<phi):
    if(gcd(e,phi)==1):
        break
    else:
        e=e+1

k=2
d=(1+(k*phi))/e
msg=12.0
print("Message data=", msg)
c=pow(msg,e)
c=math.fmod(c,n)
print("Encrypted data=",c)
m=pow(c,d)
m=math.fmod(m,n)
print("original msg sent=",m)
```

Output:

```
Message data= 12.0
Encrypted data= 3.0
original msg sent= 12.0
```

PRACTICAL 3(INS)

Code: Implementing MD5 algorithm

```
import hashlib result = hashlib.md5(b'Network
Security') result1 = hashlib.md5(b'Network
Securiti') print("The byte equivalent of hash is:
", end="") print(result.digest())
print("The byte equivalent of hash is: ", end="")
print(result1.digest())
```

Output:

```
The byte equivalent of hash is: b'\xe9`\x9b\x04\t\x93\x00\x9e\x0e6\xb9\xa4\xd7\x16\x1b\x87'
The byte equivalent of hash is: b'\xa5O\x85\xec\xf8\xda\xb9J]rHjH\x1f\x86\xc5'
```

Code: Implementing SHA algorithm

```
import hashlib
str = input("Enter the value to encode: ") result
= hashlib.sha1(str.encode())
print("The hexadecimal equivalent of SHA1 is: ")
print(result.hexdigest())
```

Output:

```
Enter the value to encode: shravani
The hexadecimal equivalent of SHA1 is:
c073de8a41880bb61d579638f0505a7e4353b113
PS C:\shravani\INS> python -u "c:\shravani\INS\MAC.py
Enter the value to encode: 123456
The hexadecimal equivalent of SHA1 is:
7c4a8d09ca3762af61e59520943dc26494f8941b
```

PRACTICAL 4(INS) Code:

```
from Crypto.PublicKey import RSA
from Crypto.Signature import pkcs1_15
from Crypto.Hash import SHA256

key=RSA.generate(2048)
private_key=key.export_key()
public_key=key.publickey().export_key()

original_document=b"This is the original document content."
modified_document=b"This is the modified document content."


original_hash= SHA256.new(original_document)
modified_hash= SHA256.new(modified_document)


signature=pkcs1_15.new(RSA.import_key(private_key)).sign(original_hash)

try:
    pkcs1_15.new(RSA.import_key(public_key)).verify(modified_hash,signature)
    print("Signature is Valid")
except(ValueError, TypeError):
    print("Signature is Invalid")
```

Output:

```
Signature is Invalid
```

PRACTICAL 5(INS)

Code: Diffie Hellman

```
P = int(input("Enter prime no. greater than 1: "))
if P>1:     for i in range(2,(P//2)+1):
if (P % i) == 0:
            print(P, "is not a prime number.")
break     else:
        print(P, "is a prime number.")         G =
(int(input("Enter the value of G:")))         a =
(int(input("Enter the private key for A:")))         b
= (int(input("Enter the private key for B:")))
        X = (G**a) % P         Y
= (G**b) % P         print("X =
",X,"Y = ",Y)
        Ka = (Y**a) % P
Kb = (X**b) % P
        print("Ka = ", Ka, "Kb = ",Kb)
```

Output:

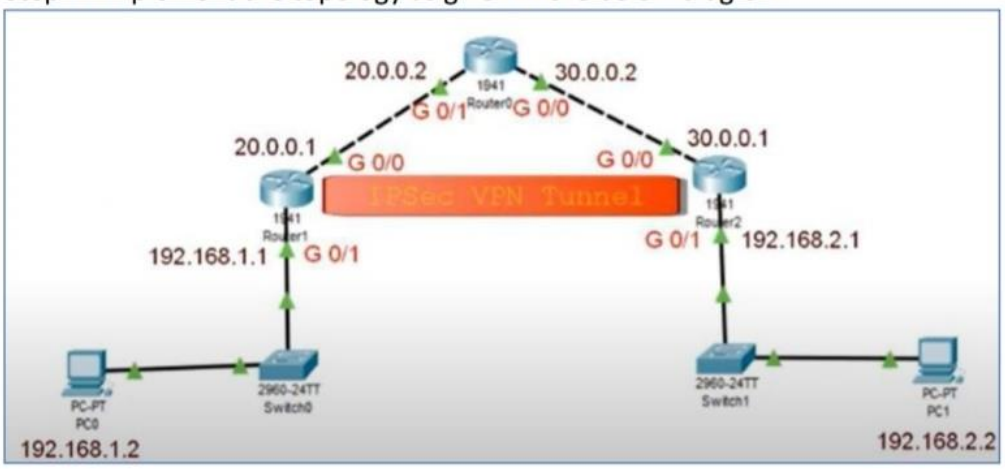
```
Enter prime no. greater than 1: 23
23 is a prime number.
Enter the value of G:9
Enter the private key for A:4
Enter the private key for B:3
X =  6 Y =  16
Ka =  9 Kb =  9
```

Practical – 6

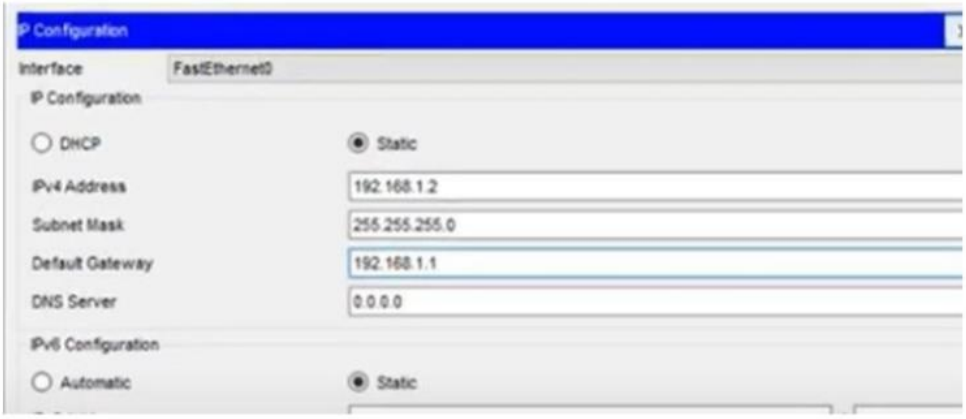
Aim: Configure IPsec on network devices to provide secure communication and protect against unauthorized access and attacks.

Part 1: Implementing the Topology using Cisco Packet Tracer, configure the IP and set the IP route

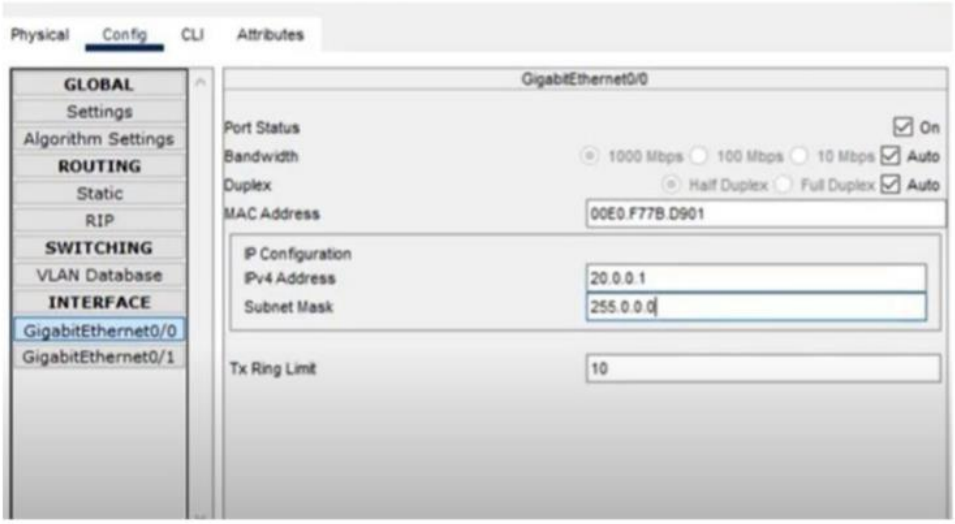
Step1 : Implement the topology as given in the below diagram



Step 2 : Configure the IP address of the PC and gateway

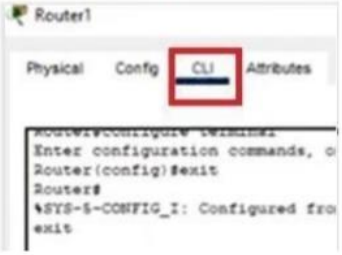


Step 3: Configure all the router as



Part 2: Configure the Hostname on all Routers and enable the security package on R1 and R2, Ping on PC from the other(All packets are lost)

Step 1: Click on router 1 click on CLI and execute the following command



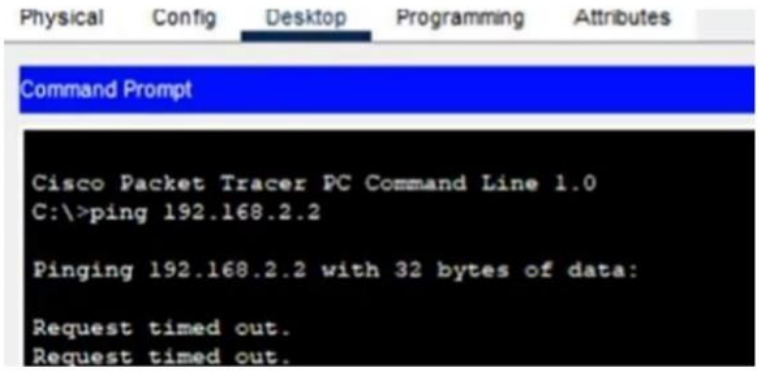
Router>**enable**
Router#**configure terminal**

Router(config)#**ip route**
0.0.0.0 0.0.0.0 20.0.0.2

Step 2 : Click on router 2 and click on CLI and execute the following command
Router>**enable**
Router#**configure terminal**

Router(config)#**ip route 0.0.0.0 0.0.0.0 30.0.0.2**

Step3: Now ping one PC from another from the command line(Ping should fail)



Part 3: Apply the Access Control List(ACL) at Router 1 and 2, Set the ISALMP policy and ISAKMP key, Set IPSec transform set

Step 1: Go to the CLI of router 1 and type the following command and enable security package

Router>**enable**
Router#**configure terminal**
Router(config)#**hostname R1**

R1(config)#**license boot module c1900 technology-package securityk9**

R1(config)#**exit**

R1#**copy run startup-config**

R1#**reload**
R1>**enable**
R1#**show version**

Step2: Repeat the above step for Router 2 with hostname R2
Step 3: Enable the policy. To do so go to CLI and type the following command
R1>**enable**
R1#**configure terminal**

R1(config)#access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.2.0

0.0.0.255

R1(config)#crypto isakmp policy 10
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 5

R1(config-isakmp)#exit

R1(config)#crypto isakmp key ismile address 30.0.0.1

R1(config)#crypto ipsec transform-set R1->R2 esp-aes 256 esp-sha-hamc
R1(config)#crypto ipsec transform-set R2->R1 esp-aes 256 esp-sha-hamc

Step4: Repeat the above step for Router 2
R2>**enable** R2#**configure terminal**

R2 (config)#access-list 100 permit ip 192.168.2.0 0.0.0.255 192.168.1.0

0.0.0.255

R2 (config)#crypto isakmp policy 10

R2 (config-isakmp)#encryption aes 256

R2 (config-isakmp)#authentication pre-share R2
(config-isakmp)#group 5

R2 (config-isakmp)#exit

R2(config)#crypto isakmp key ismile address 20.0.0.1

R2(config)#crypto ipsec transform-set R2->R1 esp-aes 256 esp-sha-hamc

Part 4 : Create the crypto map and apply to the required interface. Verify the output by pinging one PC from other

R1(config)#Crypto map IPSEC-MAP 10 ipsec-isakmp
R1(Config-crypto-map)#set peer 30.0.0.1

R1(Config-crypto-map)#set pfs group5

R1(Config-crypto-map)#set security-association lifetime 86400
R1(Config-crypto-map)#set transform-set R1->R2

R1(Config-crypto-map)#match address 100
R1(Config-crypto-map)#exit R1(Config)#interface g0/0

R1(Config-if)#crypto map IPSEC-MAP

Repeat the same for router 2.
R2(Config-crypto-map)#set peer 20.0.0.1 R1(Config-crypto-map)#set transform-set R2->R1

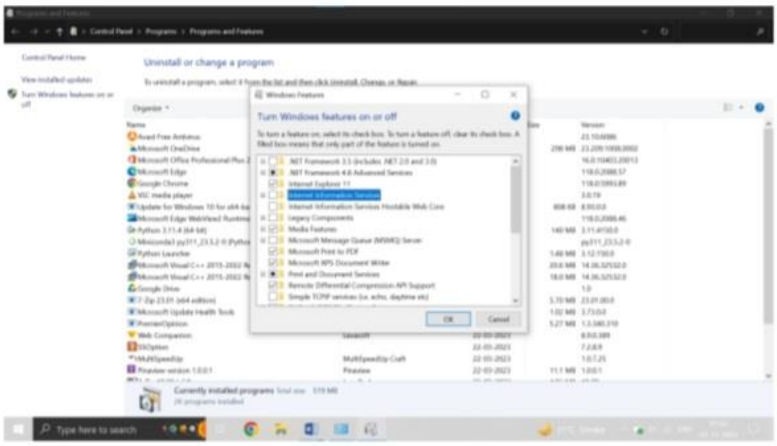
Now ping the system again we will get the output

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time<1ms TTL=126
Reply from 192.168.2.2: bytes=32 time<1ms TTL=126
Reply from 192.168.2.2: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

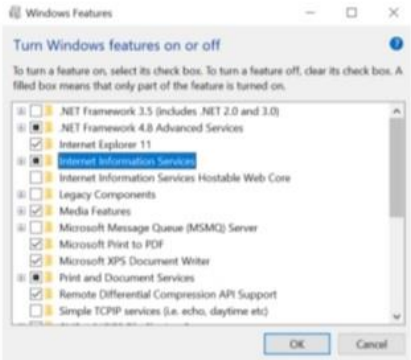
Practical – 7

Aim: Configure and implement secure web communication using SSL/TLS protocols, including certificate management and secure session establishment.

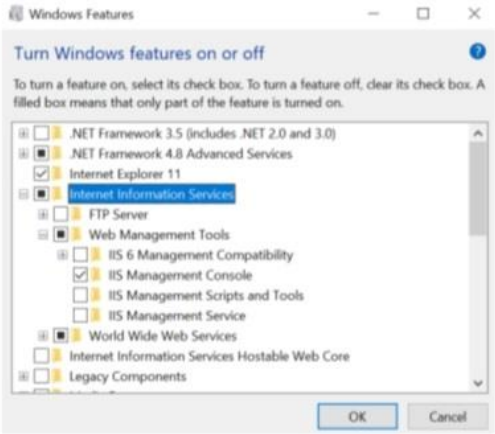
Steps: Go to Control Panel->Program and Features->Turn Windows Feature on



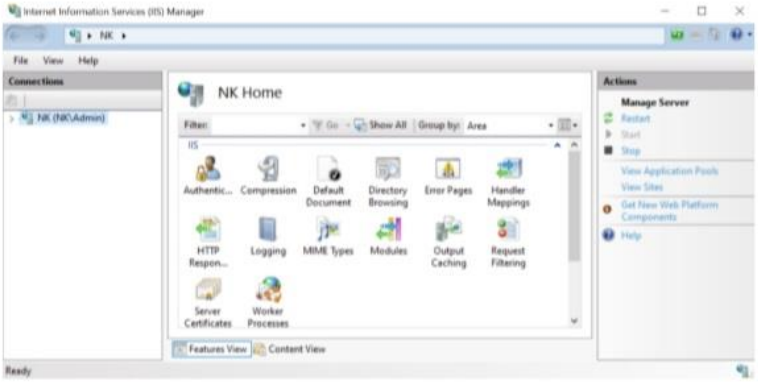
Turn On Internet Information Service



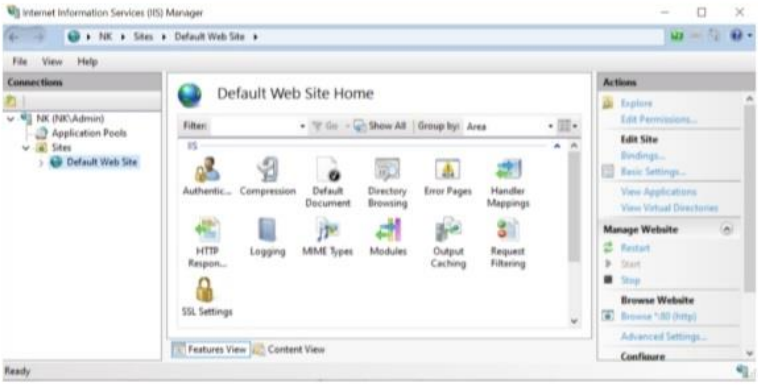
Turn on the IIS Management Console and World Wide Web services



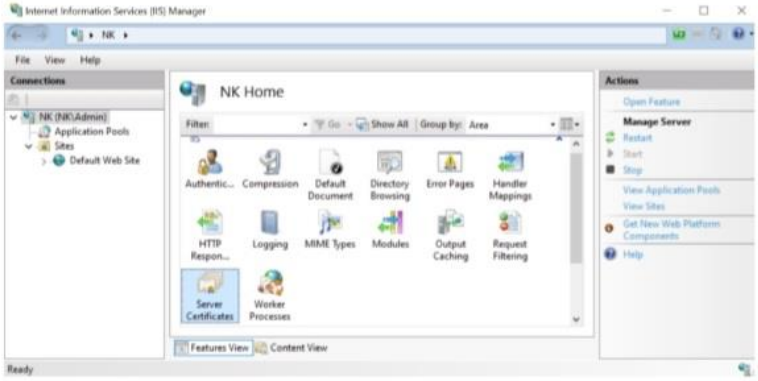
Now from window explorer, open IIS Manager



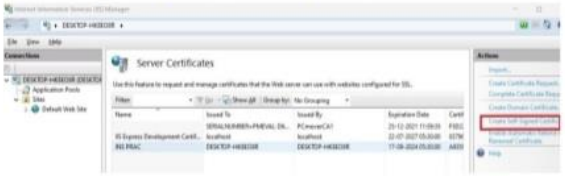
Click on server in the left pane and Choose default web site.



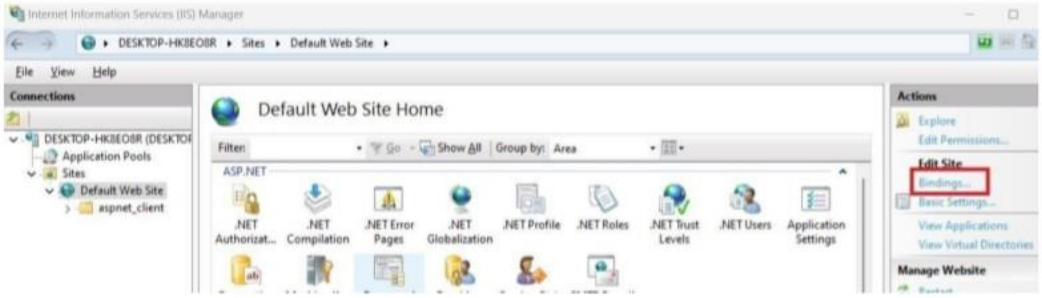
Click on Desktop and click on server Certificate

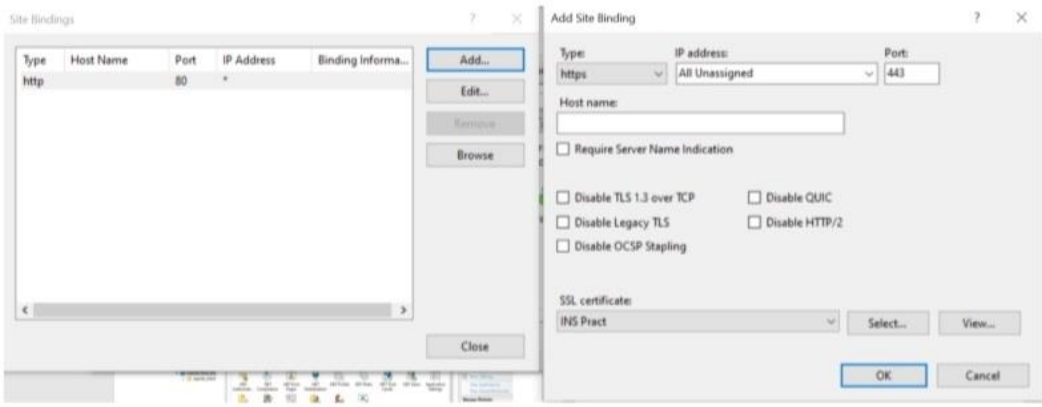


Create Self Signed Certificate, give a proper name and certificate store as Web hosting

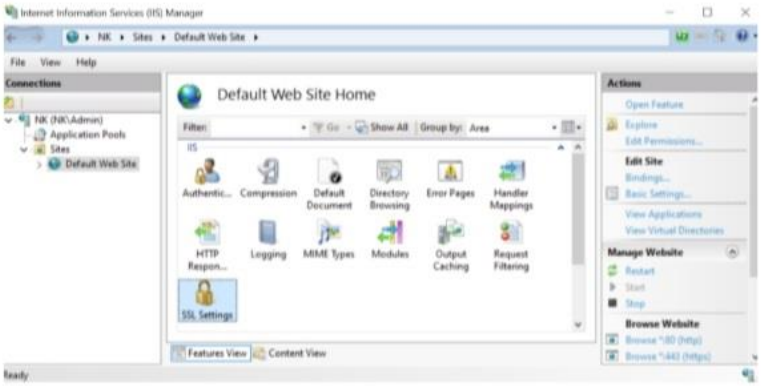


Click on Default Website & Click on bindings and bind with type **https** port **443** by clicking Add btn.

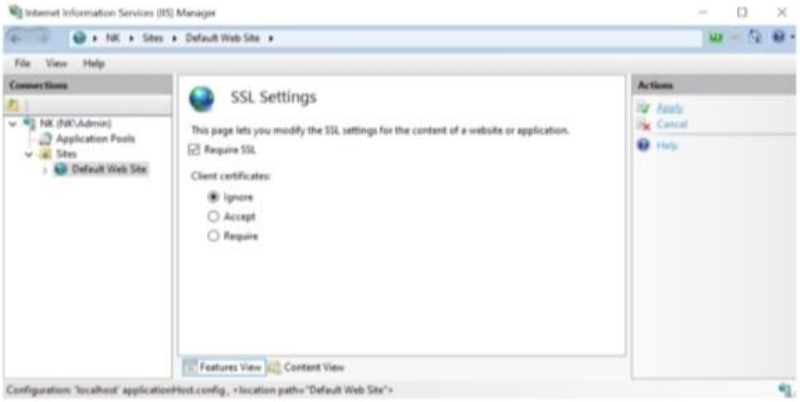




Click on Default Website and click SSL Settings

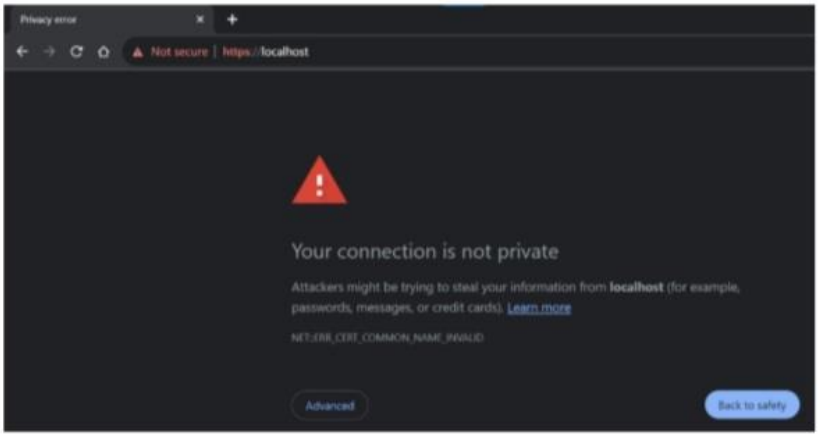


Click on Require SSL Checkbox and click Apply in the right pane , it is successfully changed.

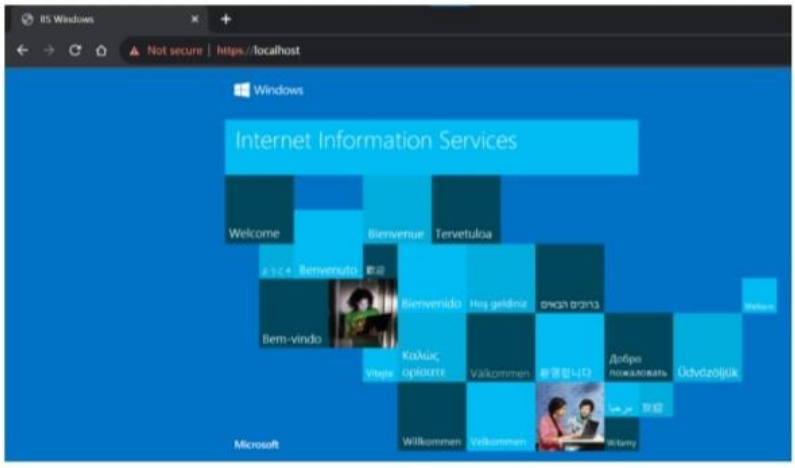


Type **localhost** in google URL and you would get the following page

Now, Type **https://localhost** in google URL and you would get the following page



Now, click on **Advanced** and then click on proceed to localhost and you will get the following screen.



Practical – 9

Aim: To do Detect and Analyse Malware

Steps:

Open the website www.virusshare.com to download the clean sample of Malware.

Create account by sending a mail to Melissa at melissa97@virusshare.com with 'access' in the subject. She will review your request and hopefully send you an invitation link.

VirusShare.com - Because Sharing is Caring

Home • Hashes • Research • About • Swag Shop

Please login to search and download.

System currently contains 71,601,954 malware samples.

Report for a sample recently added to the system:
2cdf012c81bdc294cbfdb48cf04acbd5e84461f4069943eb434886b5806247b9

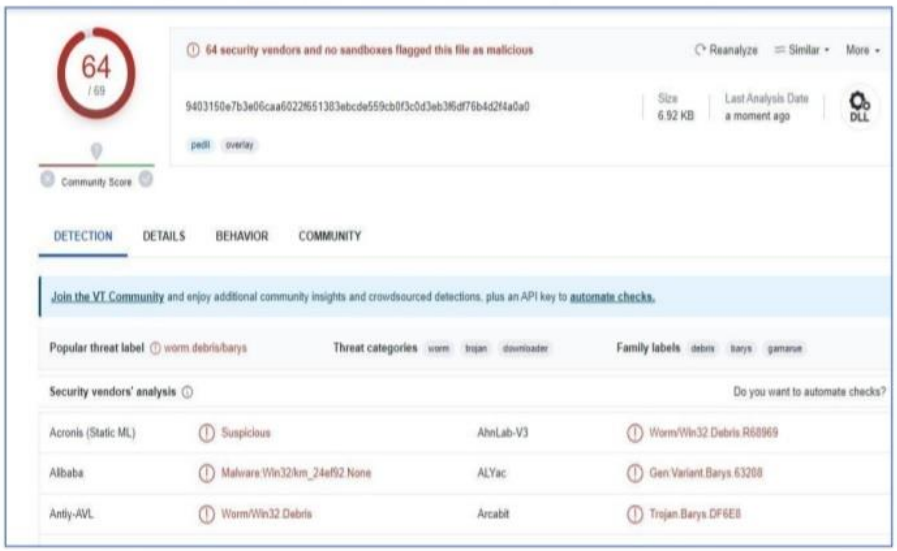
VirusShare info last updated 2023-11-04 00:00:00 UTC

MD5	a1482914b0b7ab9459c239486d62e74b
SHA1	ef419c5f83ce5543196207d95b89fff2fbdc555a
SHA256	2cdf012c81bdc294cbfdb48cf04acbd5e84461f4069943eb434886b5806247b9
SSDeep	12288:yfgJM2QgAIX1nnH7Wj9sYfwpTJIOZT2mj53qjnNh3UF:yoJM8hb2fCIOZTPj5WNh3UF
Authentihash	9c71bd61ba35b0a1691b7ea8b85bea2900d2744f0cc1574b23ee4168eec6dfe7
Size	612,076 bytes
File Type	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows
Mime Type	application/x-dosexec
Extension	exe

By clicking the above download icon the Malware gets downloaded in ZIP format.

Do not unzip the file, we create a folder "Malware" on desktop and save the file in the folder. In order to analyse the Malware, we select the website www.virustotal.com

Click on “Choose File” and select the file from the location (ZIP file will do, if asks for password enter infected). We get the following after the upload is completed



We interpret the following findings

a) 64 security vendors out of 69 flagged this file as malicious

b) The detection tab shows the threats-type which were flagged by the vendors for e.g

c) The details tab gives the following information

   i. Basic properties

   ii. History

   iii. Compiler products

   iv. Header

   v. Sections

   vi. Imports

   vii. Exports

   viii. Overlays

d) The Behavior tab gives the following information

   i. Activity summary

   ii. MITRE ATT&CK Tactics and Techniques

   iii. Behavior Similarity Hashes

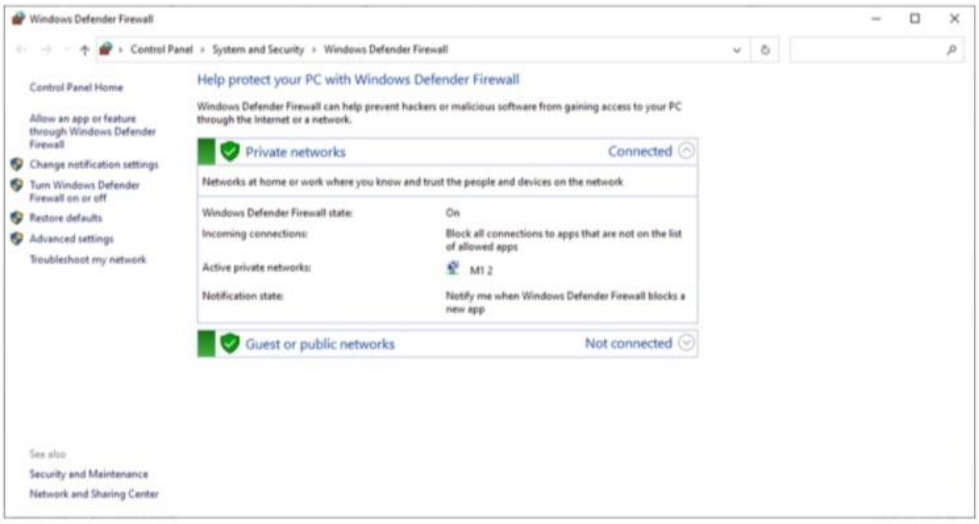
   iv. Process and service actions

Practical - 10

Aim: Firewall Configuration and Rule- based Filtering

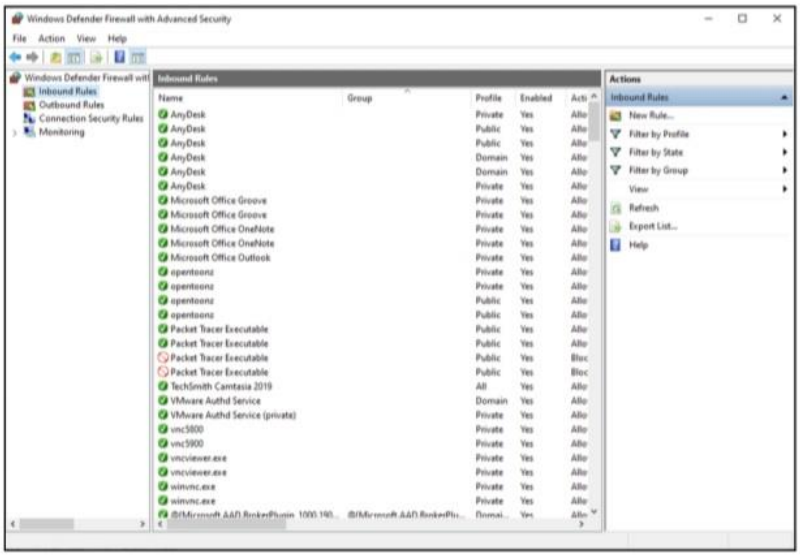
Part 1: Blocking the Port

We access any website through the browser and confirm that the HTTP/HTTPS protocols are working.

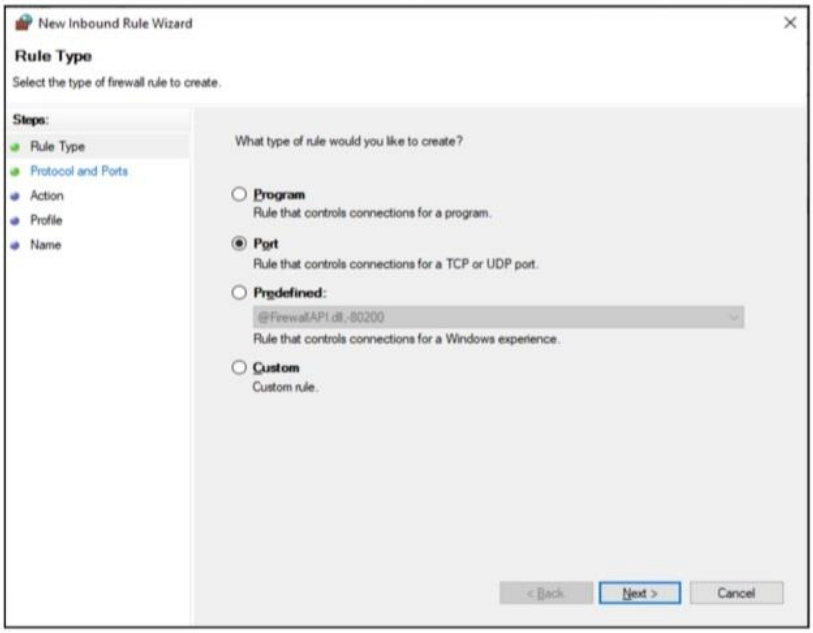


Step 2: We open 'Windows Defender Firewall'

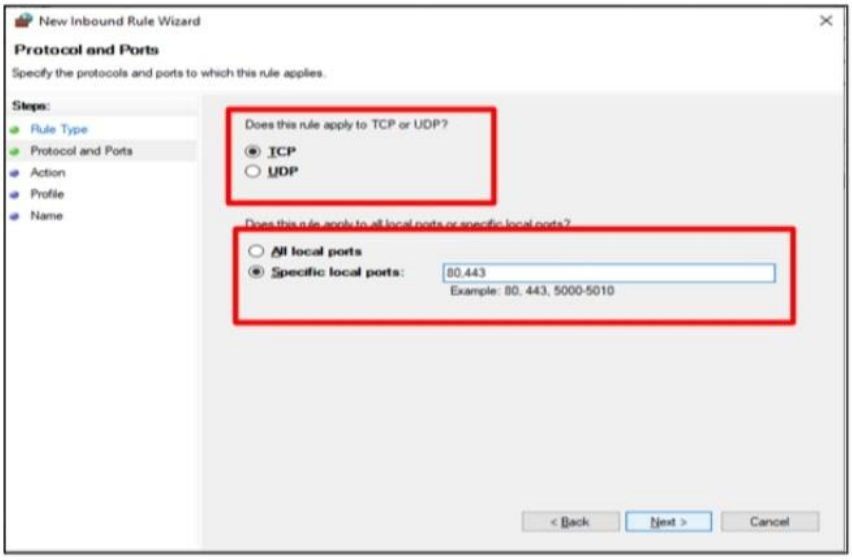
Next we click on ‘Inbound Rules’



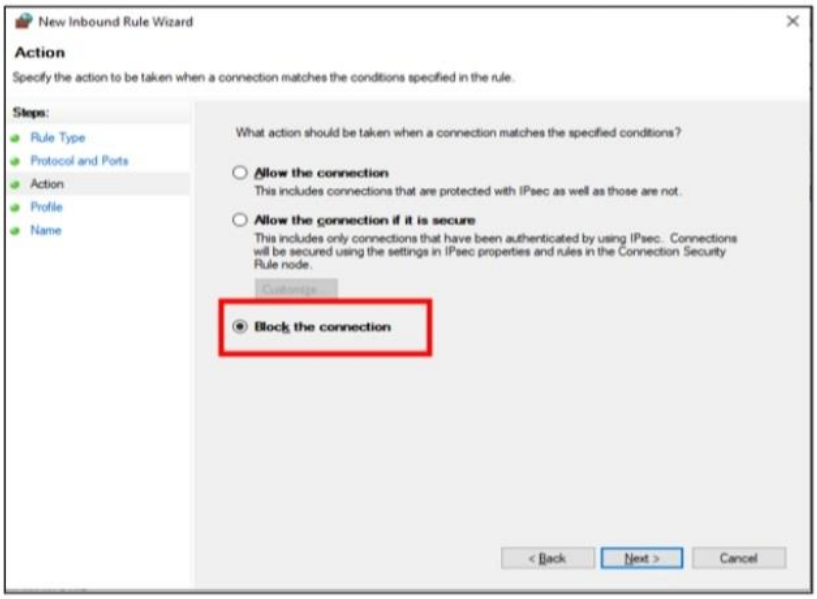
Then click on ‘New Rule’



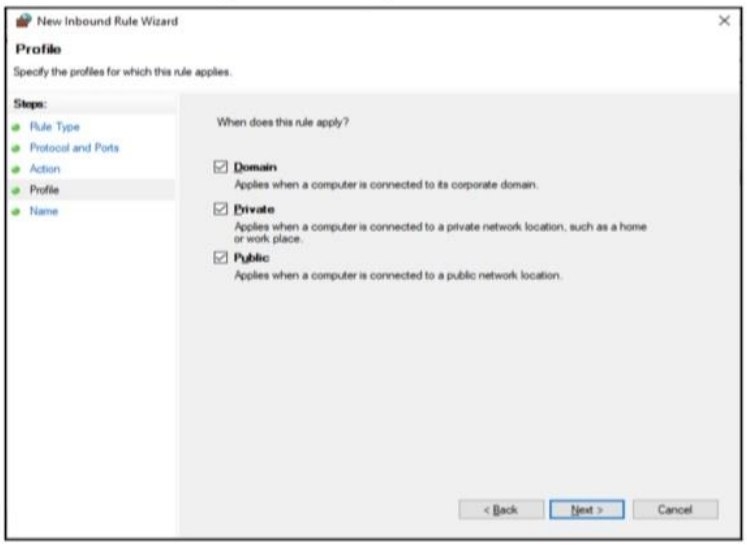
Select the radio button 'Port' and click 'Next' and enter the following



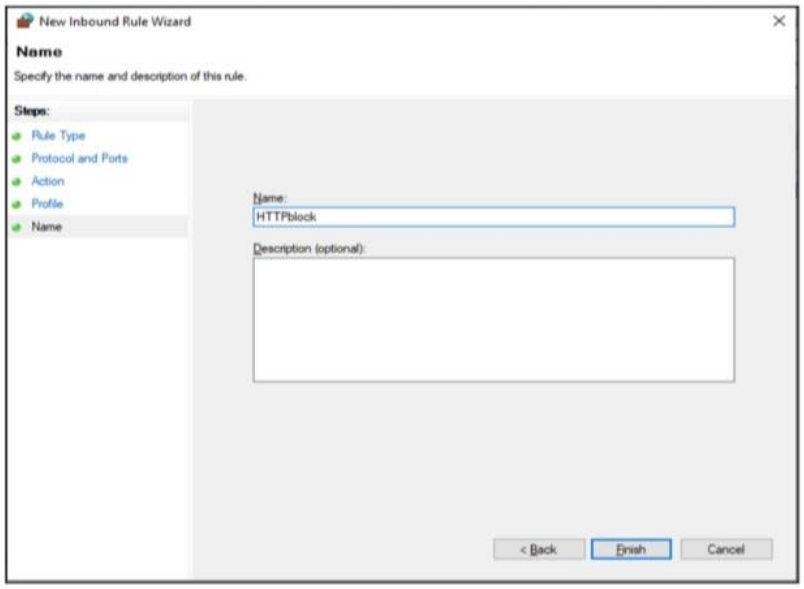
After next, we need to finalise the rule



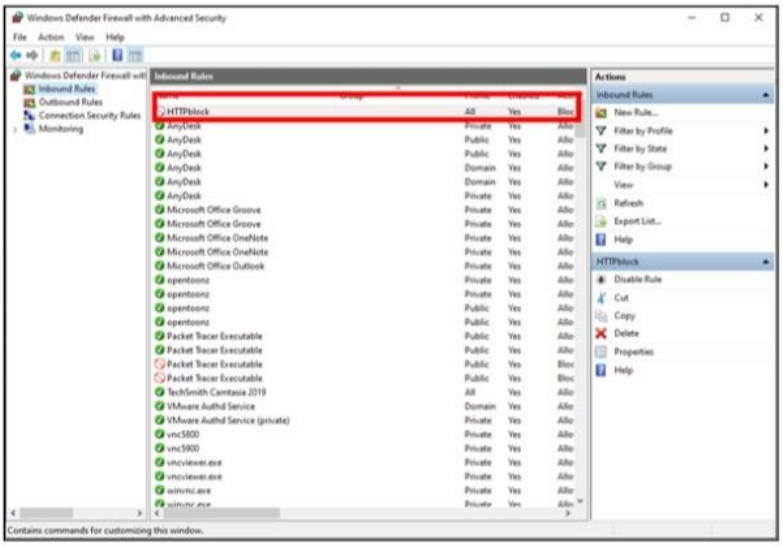
Click 'Next' and we get the following



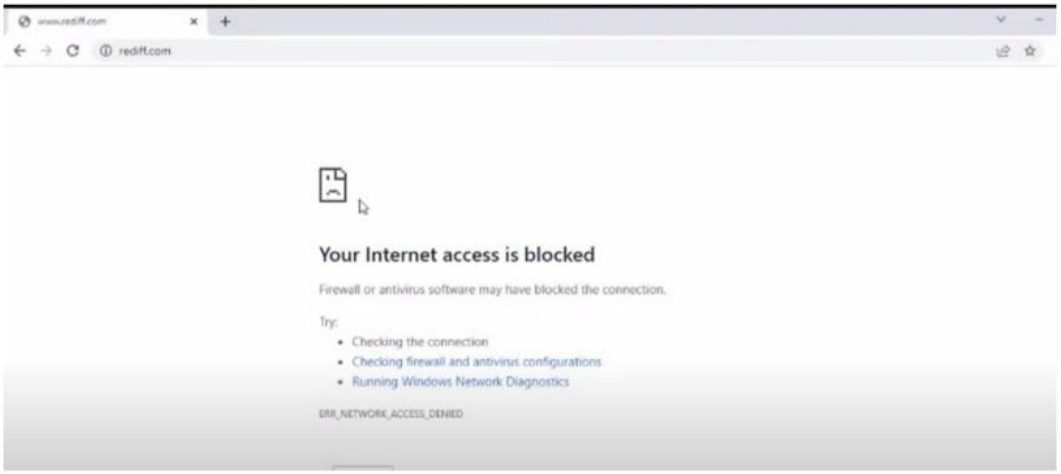
After clicking the 'Next' button we need to name the rule and click finish



The Inbound rule is added

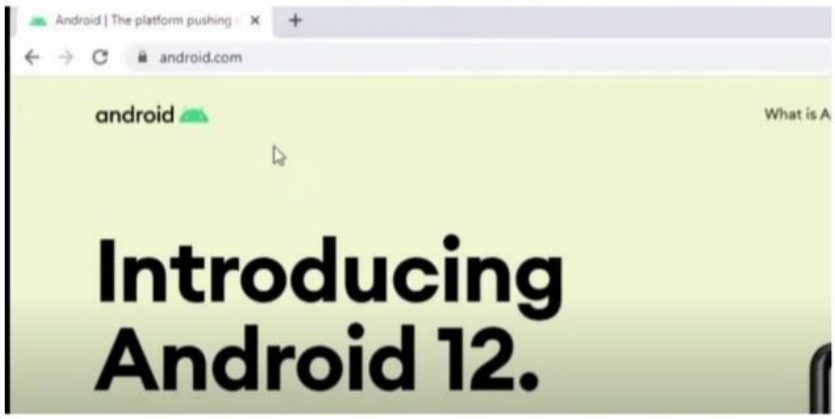


We repeat all the above steps for creating 'Outbound Rules', and then try to access the internet. We see that the accessed is blocked

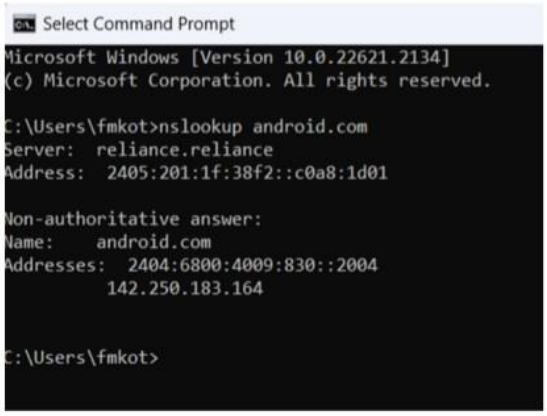


Part 2: Blocking the website www.android.com

We open the browser and access the website, which is now accessible



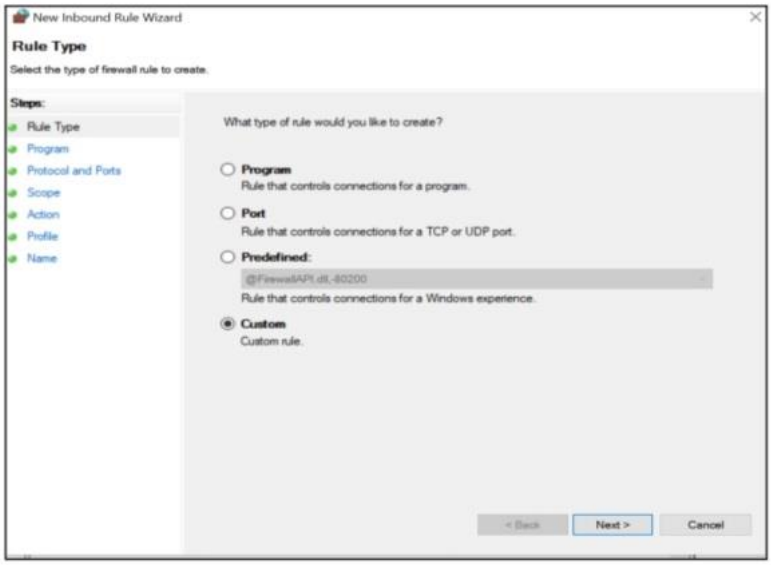
We find the IP addresses of the website using the following command

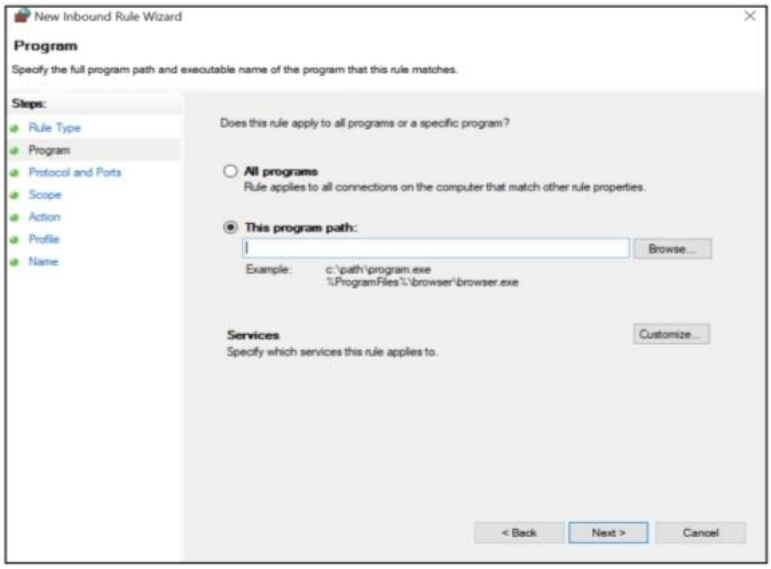


We save the IP addresses

IPv4	216.58.196.68
IPv6	2404:6800:4009:809::2004

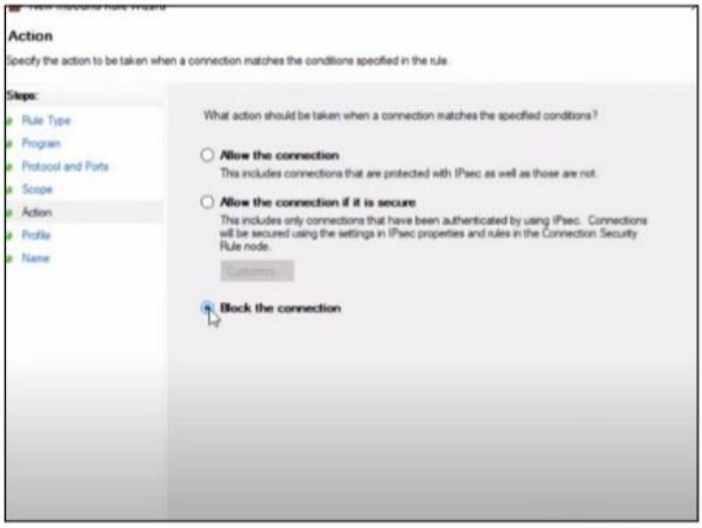
We open the windows Firewall settings and apply the Inbound Rule



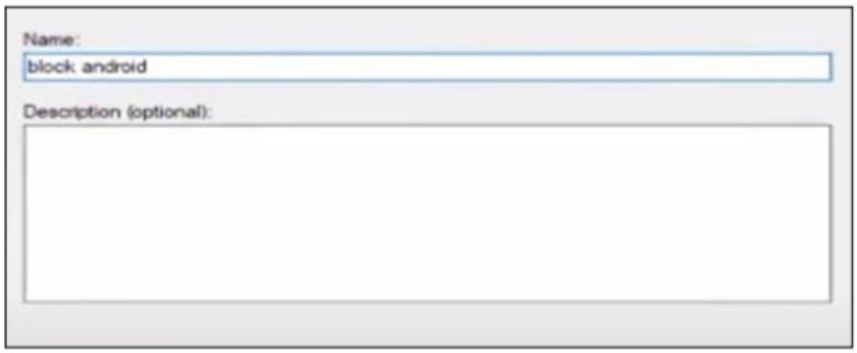


Insert the IP addresses both IPv4 and IPv6

Select Block connection



Provide a suitable name and finish



Repeat the above for Outbound Rules now!

if we try to access the website www.android.com , it would be blocked