

## PRACTICAL-1

**AIM:-** List and practice various web commands on dos Linux.

COMMANDS:

**1 .Ipconfig:-** IPCONFIG stands for Internet Protocol Configuration. This is a command-line application which displays all the current TCP/IP ,network configuration, refreshes the DHCP and DNS (Domain Name Server).

```
C:\Users\Aman Patel>ipconfig

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2402:3a80:1306:a677:d140:a483:bad8:6b
a1
```

**2.Netstat:-** netstat is a command-line network utility that displays network connections for Transmission Control Protocol , routing tables, and a number of network interface and network protocol statistics.

```
C:\Users\Aman Patel>netstat
```

### Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:1042	lepiy:56349	ESTABLISHED
TCP	127.0.0.1:1042	lepiy:56350	ESTABLISHED
TCP	127.0.0.1:9012	lepiy:56357	ESTABLISHED
TCP	127.0.0.1:13030	lepiy:49671	ESTABLISHED
TCP	127.0.0.1:17532	lepiy:56354	ESTABLISHED
TCP	127.0.0.1:49671	lepiy:13030	ESTABLISHED
TCP	127.0.0.1:56349	lepiy:1042	ESTABLISHED
TCP	127.0.0.1:56350	lepiy:1042	ESTABLISHED
TCP	127.0.0.1:56354	lepiy:17532	ESTABLISHED
TCP	127.0.0.1:56357	lepiy:9012	ESTABLISHED
TCP	192.168.45.152:56389	237:4070	ESTABLISHED

**3.Tarcert:-** Tracert is a diagnostic command-line interface commands for displaying possible routes (paths) and transit delays of packets across an Internet Protocol (IP) network.

```
C:\Users\Aman Patel>tracert
```

```
Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
              [-R] [-S srcaddr] [-4] [-6] target_name
```

### Options:

-d	Do not resolve addresses to hostnames.
-h maximum_hops	Maximum number of hops to search for target.
-j host-list	Loose source route along host-list (IPv4-only).
-w timeout	Wait timeout milliseconds for each reply.
-R	Trace round-trip path (IPv6-only).
-S srcaddr	Source address to use (IPv6-only).
-4	Force using IPv4.
-6	Force using IPv6.

**4.Ping:-** Ping (Packet Internet Groper) is a method for determining communication latency between two networks or ping is a method of determining the time it takes for data to travel between two devices or across a network.

```
C:\Users\Aman Patel>ping
```

```
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] | [-k host-list]]
          [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
          [-4] [-6] target_name
```

**Options:**

```
-t          Ping the specified host until stopped.
           To see statistics and continue - type Control-Break;
           To stop - type Control-C.
-a          Resolve addresses to hostnames.
-n count    Number of echo requests to send.
-l size     Send buffer size.
-f          Set Don't Fragment flag in packet (IPv4-only).
-i TTL      Time To Live.
-v TOS      Type Of Service (IPv4-only. This setting has been deprecated
and has no effect on the type of service field in the IP
Header).
-r count    Record route for count hops (IPv4-only).
-s count    Timestamp for count hops (IPv4-only).
-j host-list Loose source route along host-list (IPv4-only).
-k host-list Strict source route along host-list (IPv4-only).
-w timeout  Timeout in milliseconds to wait for each reply.
-R          Use routing header to test reverse route also (IPv6-only)
.
           Per RFC 5095 the use of this routing header has been
           deprecated. Some systems may drop echo requests if
           this header is used.
-S srcaddr  Source address to use.
-c compartment Routing compartment identifier.
-p          Ping a Hyper-V Network Virtualization provider address.
-4          Force using IPv4.
-6          Force using IPv6.
```

**5.Pathping:-** The PathPing command is a command-line network utility included in Windows NT operating systems since Windows 2000 that combines the functionality of ping with that of tracert. It is used to locate spots that have network latency and network loss.

```
C:\Users\Aman Patel>pathping
```

```
Usage: pathping [-g host-list] [-h maximum_hops] [-i address] [-n]  
               [-p period] [-q num_queries] [-w timeout]  
               [-4] [-6] target_name
```

**Options:**

-g host-list	Loose source route along host-list.
-h maximum_hops	Maximum number of hops to search for target.
-i address	Use the specified source address.
-n	Do not resolve addresses to hostnames.
-p period	Wait period milliseconds between pings.
-q num_queries	Number of queries per hop.
-w timeout	Wait timeout milliseconds for each reply.
-4	Force using IPv4.
-6	Force using IPv6.

**6.nslookup:-** Nslookup is the name of a program that lets users enter a host name and find out the corresponding IP address or domain name system (DNS) record. Users can also enter a command in nslookup to do a reverse DNS lookup and find the host name for a specified IP address.

```
C:\Users\Aman Patel>nslookup  
Default Server: UnKnown  
Address: 192.168.45.178
```

**7.Arp:**The Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address-

```
C:\Users\Aman Patel>arp
```

Displays and modifies the IP-to-Physical address translation tables used by address resolution protocol (ARP).

```
ARP -s inet_addr eth_addr [if_addr]
```

```
ARP -d inet_addr [if_addr]
```

```
ARP -a [inet_addr] [-N if_addr] [-v]
```

**-a** Displays current ARP entries by interrogating the current protocol data. If inet\_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.

**-g** Same as -a.

**-v** Displays current ARP entries in verbose mode. All invalid entries and entries on the loop-back interface will be shown.

**inet\_addr** Specifies an internet address.

**-N if\_addr** Displays the ARP entries for the network interface specified by if\_addr.

**-d** Deletes the host specified by inet\_addr. inet\_addr may be wildcarded with \* to delete all hosts.

**-s** Adds the host and associates the Internet address inet\_addr with the Physical address eth\_addr. The Physical address is given as 6 hexadecimal bytes separated by hyphens. The entry is permanent.

**eth\_addr** Specifies a physical address.

**if\_addr** If present, this specifies the Internet address of the interface whose address translation table should be modified.

If not present, the first applicable interface will be used.

Example:

```
> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry.
> arp -a .... Displays the arp table.
```

**8.Route:-**The function and syntax of windows route command is similar to the Linux route command. It uses the command to manually to configure the route in the routing table.

```
Command Prompt
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aman Patel>route

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
                                [MASK netmask] [gateway] [METRIC metric] [IF interface]

-f          Clears the routing tables of all gateway entries. If this is
            used in conjunction with one of the commands, the tables are
            cleared prior to running the command.

-p          When used with the ADD command, makes a route persistent across
            boots of the system. By default, routes are not preserved
            when the system is restarted. Ignored for all other commands,
            which always affect the appropriate persistent routes.

-4          Force using IPv4.

-6          Force using IPv6.

command     One of these:
            PRINT      Prints a route
            ADD        Adds a route
            DELETE     Deletes a route
            CHANGE     Modifies an existing route

destination Specifies the host.
MASK          Specifies that the next parameter is the 'netmask' value.
netmask       Specifies a subnet mask value for this route entry.
            If not specified, it defaults to 255.255.255.255.
gateway       Specifies gateway.
interface     the interface number for the specified route.
METRIC        specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
```

```

Command Prompt

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
and '?' matches any one char. Examples: 157.*.1, 157.*, 127.*, *224*.

Pattern match is only allowed in PRINT command.
Diagnostic Notes:
  Invalid MASK generates an error, that is when (DEST & MASK) != DEST.
  Example> route ADD 157.0.0.0 MASK 155.0.0.0 157.55.80.1 IF 1
           The route addition failed: The specified mask parameter is invalid. (Destination & Mask) != Destination.

Examples:

> route PRINT
> route PRINT -4
> route PRINT -6
> route PRINT 157*      .... Only prints those matching 157*

> route ADD 157.0.0.0 MASK 255.0.0.0 157.55.80.1 METRIC 3 IF 2
           destination^      ^mask      ^gateway      metric^      ^
                                   Interface^

  If IF is not given, it tries to find the best interface for a given
  gateway.
> route ADD 3ffe::/32 3ffe::1

> route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2

  CHANGE is used to modify gateway and/or metric only.

> route DELETE 157.0.0.0
> route DELETE 3ffe::/32

```

**9.getmac:** The getmac (short for **get MAC address**) is a simple Windows network command-line utility used to find the physical address of the network adapters (NIC) in a computer. This tool is typically used in troubleshooting network issues.

```

C:\Users\Aman Patel>getmac

Physical Address      Transport Name
=====
B4-8C-9D-8B-8D-A3    \Device\Tcpip_{65FCE4C6-741F-491D-A2FC-9519FB95BFE4}
58-11-22-3F-40-1D    Media disconnected
B4-8C-9D-8B-8D-A2    Media disconnected

```

**10.Telnet:**The telnet command is a network protocol that allows users to connect to remote devices or servers via a text - based interface, typically for testing and troubleshooting. It connects to the specified host and port, defaulting to port 23 if not specified. Since Telnet transmits data in plain text, it is not secure for sensitive information over untrusted networks.

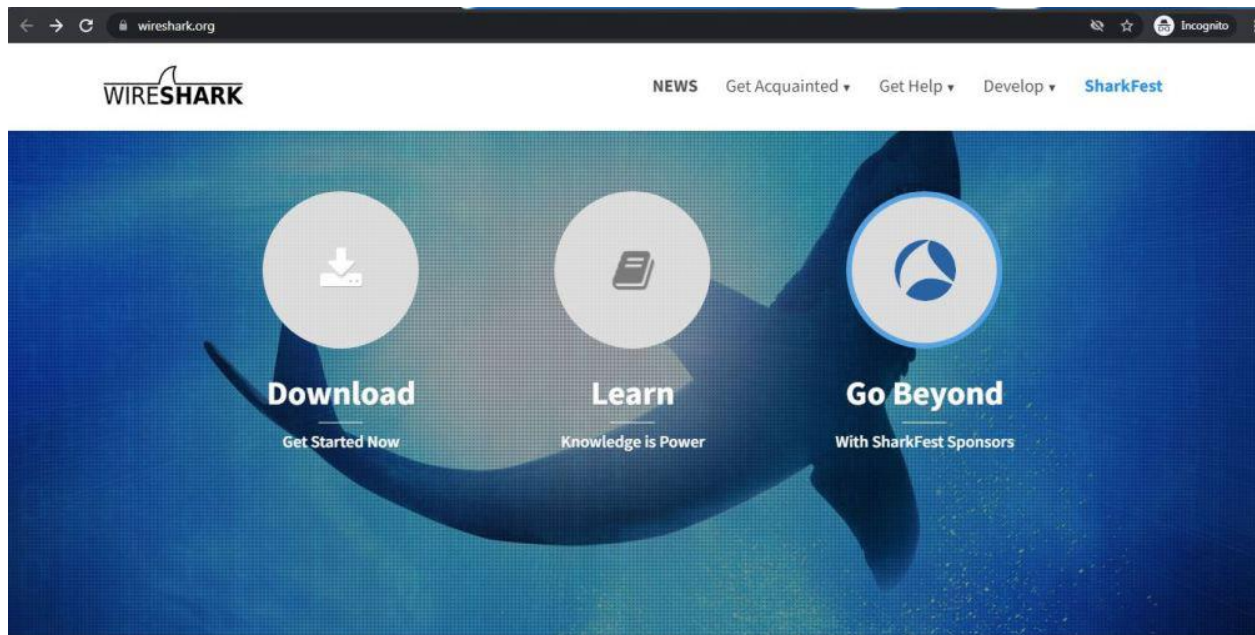
**11.FTP:** The ftp command is a network protocol used to transfer files between a local computer and a remote server. It establishes a connection to the specified host, allowing users to upload, download, and manage files on the server. FTP operates on default ports 20 and 21 and supports various commands for file manipulation and navigation.



## PRACTICAL 2

**AIM:- To install the WIRESHARK in computer.**

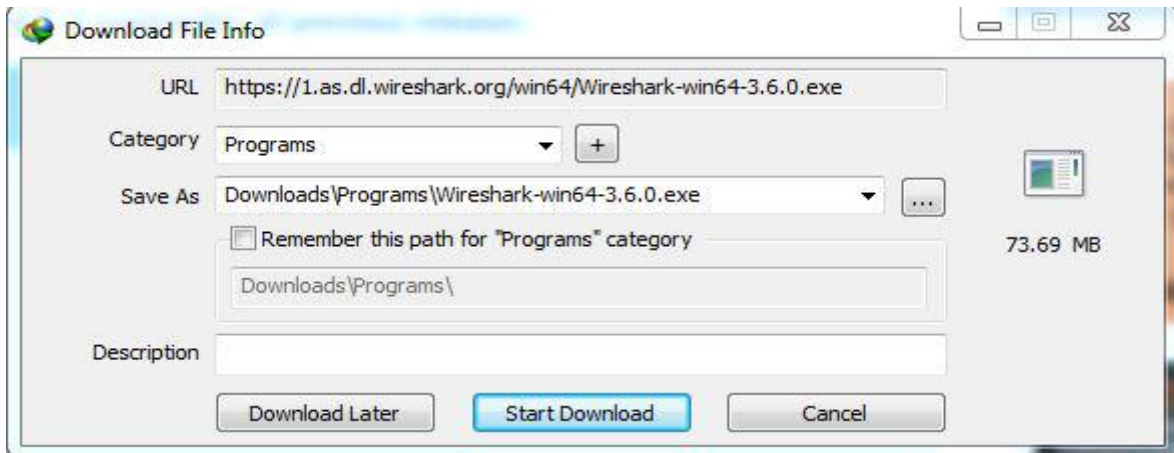
**Step 1:** Visit the official Wireshark website using any web browser.



**Step 2:** Click on Download, a new webpage will open with different installers of Wireshark.



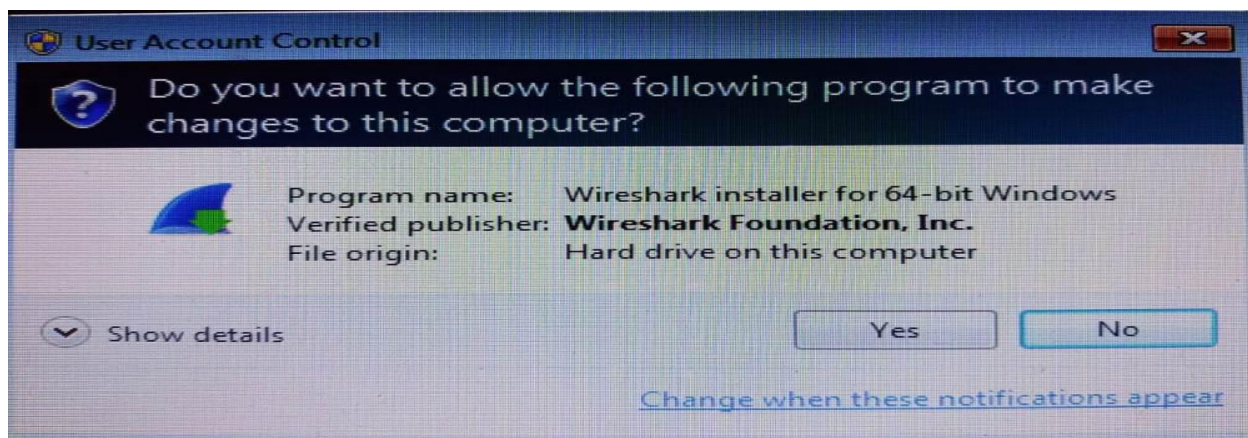
**Step 3:** Downloading of the executable file will start shortly. It is a small 73.69 MB file that will take some time.



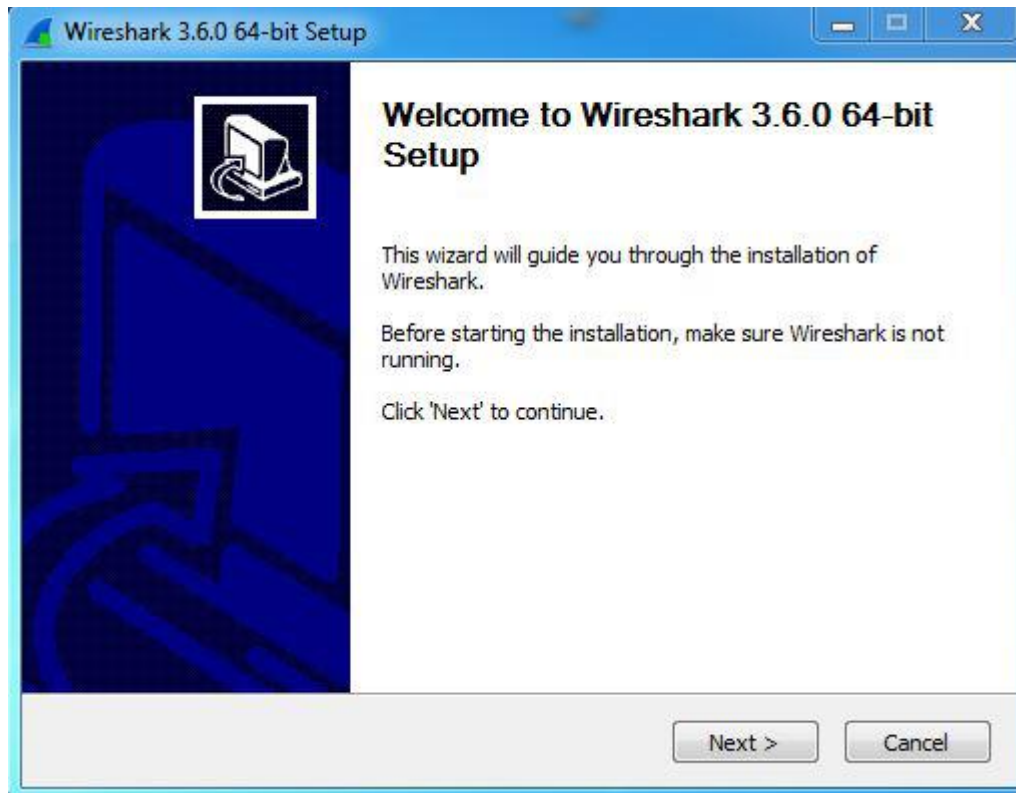
**Step 4:** Now check for the executable file in downloads in your system and run it.



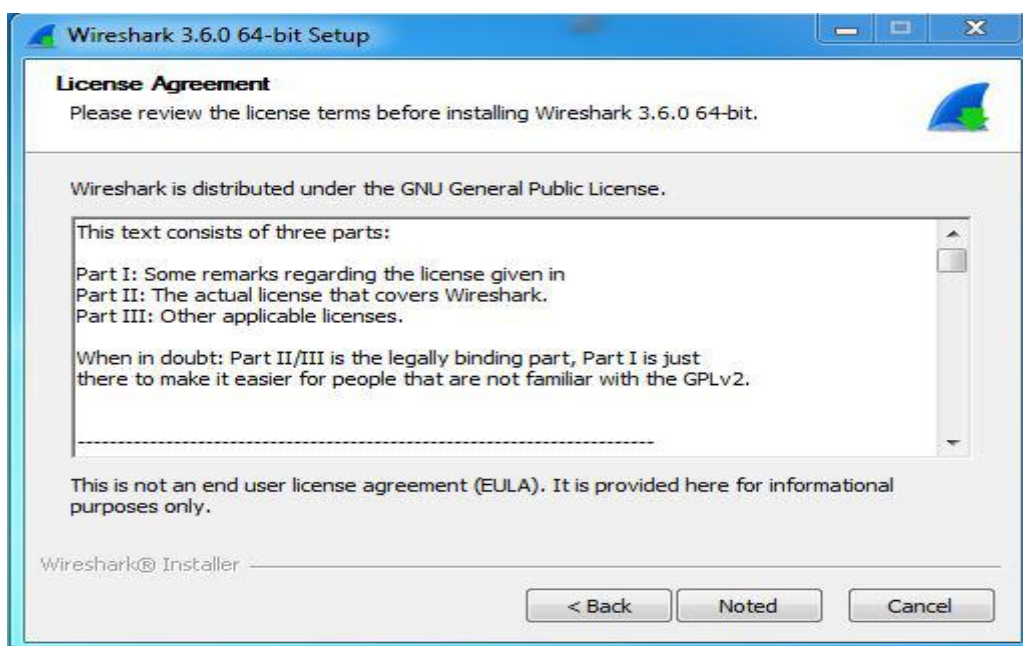
**Step 5:** It will prompt confirmation to make changes to your system. Click on Yes.



**Step 6:** Setup screen will appear, click on Next.

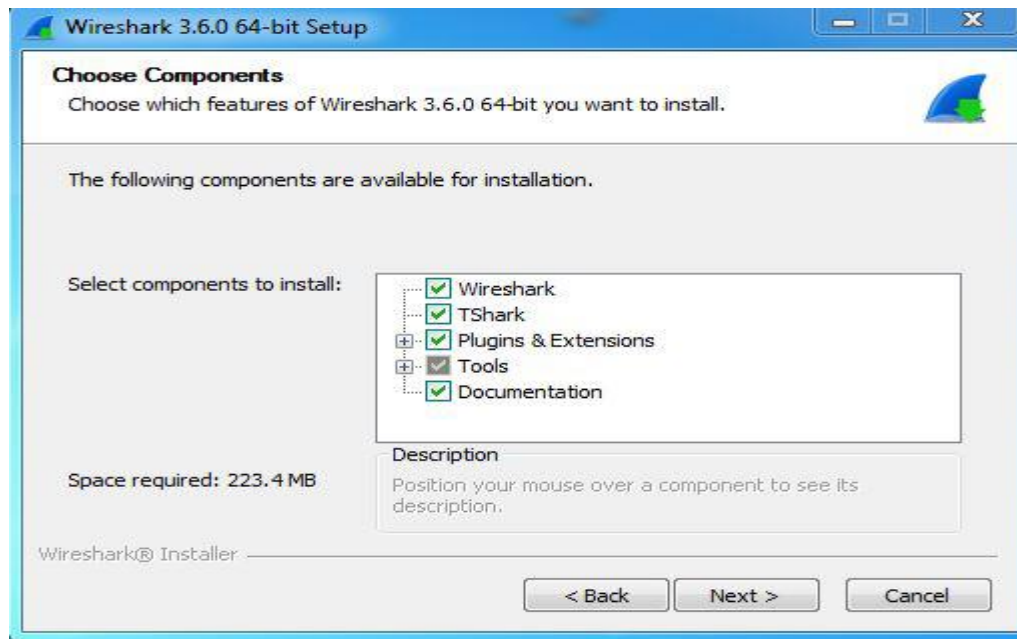


**Step 7:** The next screen will be of License Agreement, click on Noted.

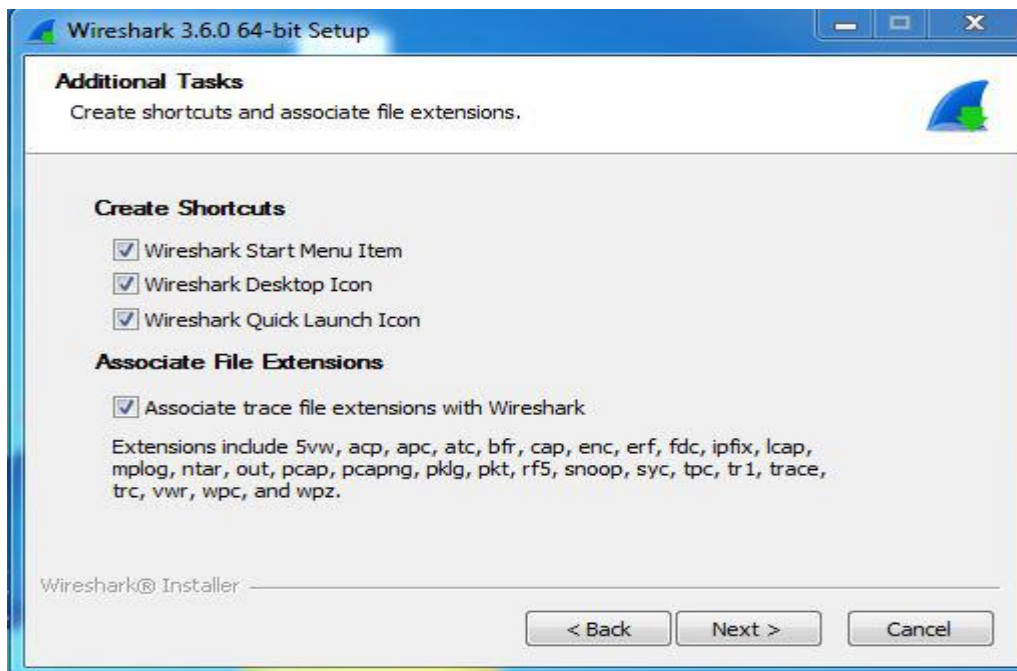




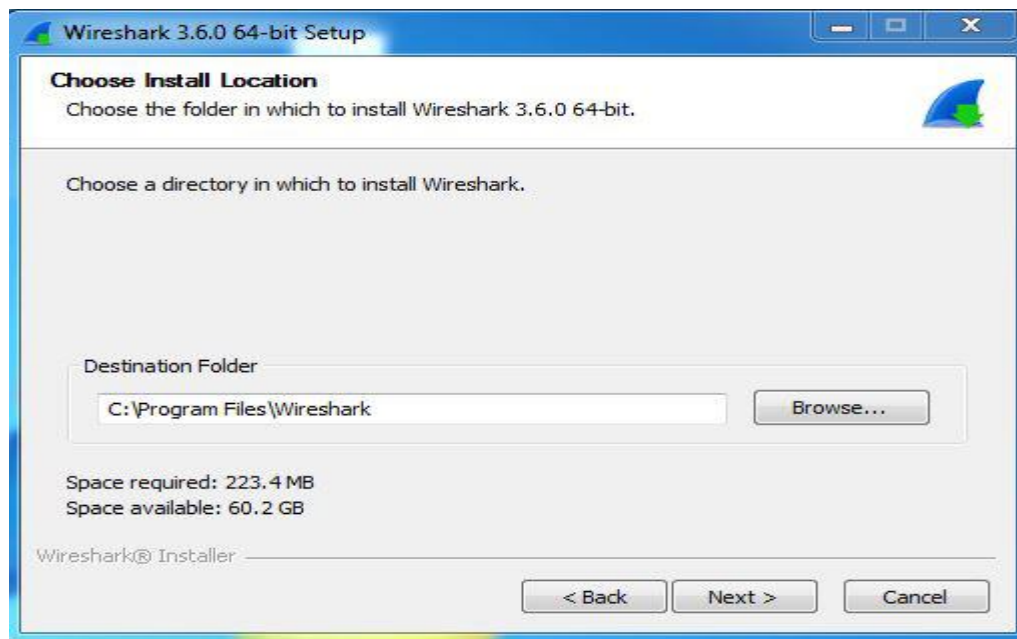
**Step 8:** This screen is for choosing components, all components are already marked so don't change anything just click on the Next button.



**Step 9:** This screen is of choosing shortcuts like start menu or desktop icon along with file extensions which can be intercepted by Wireshark, tick all boxes and click on Next button.



**Step 10:** The next screen will be of installing location so choose the drive which will have sufficient memory space for installation. It needed only a memory space of 223.4 MB.



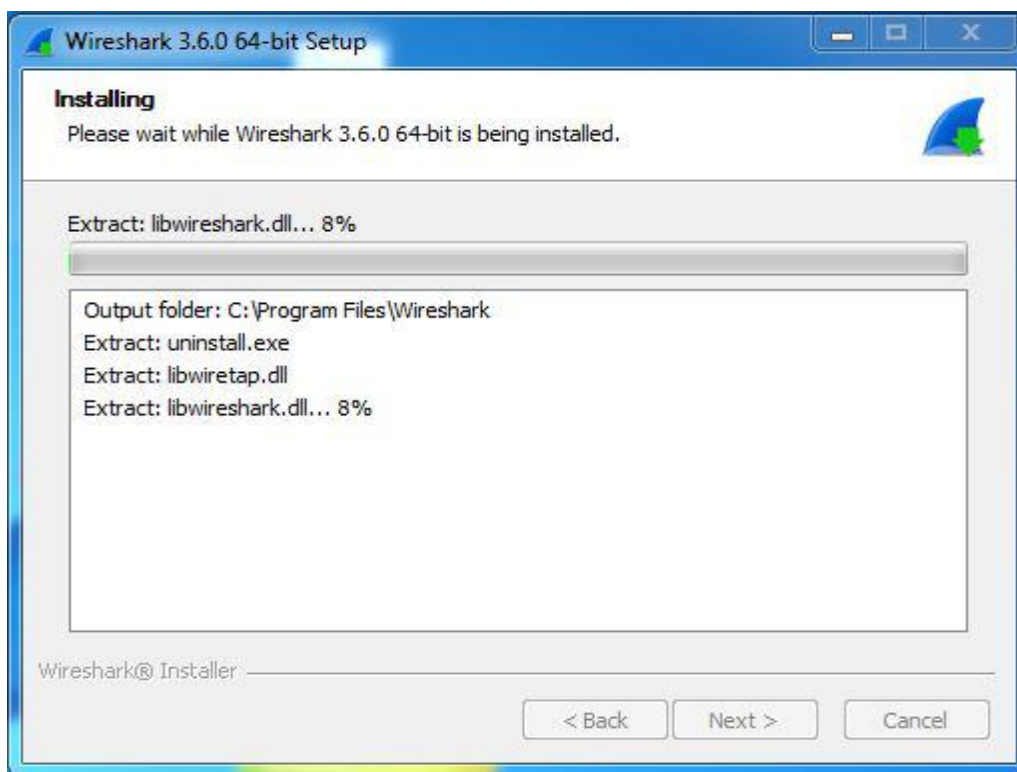
**Step 11:** Next screen has an option to install Npcap which is used with Wireshark to capture packets *pcap* means packet capture so the install option is already checked don't change anything and click the next button.



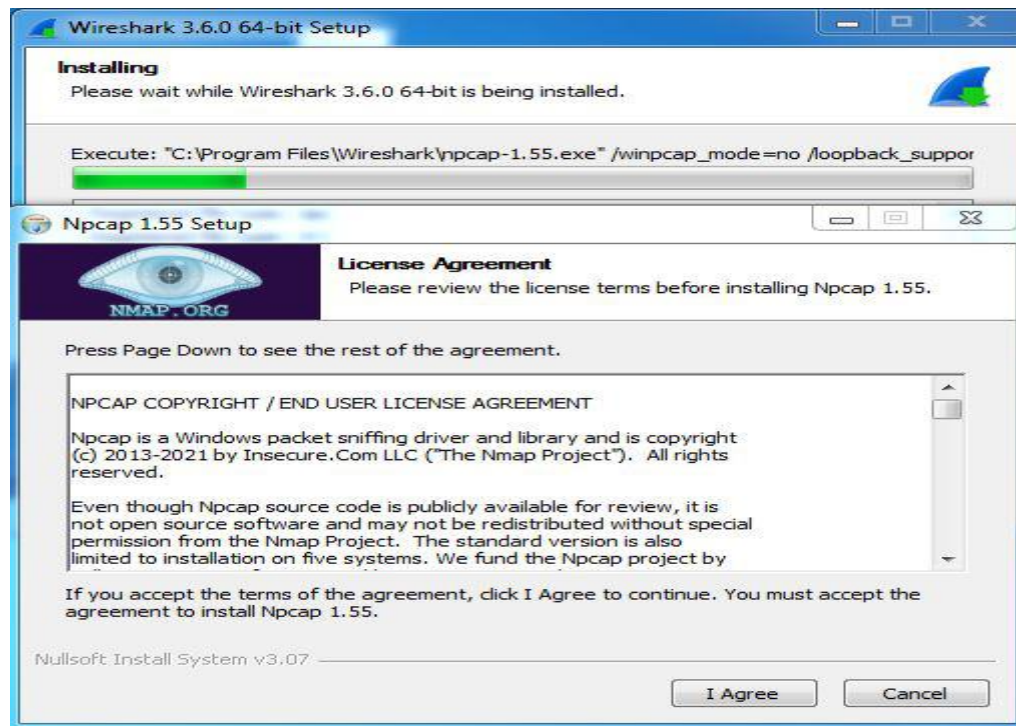
**Step 12:** Next screen is about USB network capturing so it is one's choice to use it or not, click on Install.



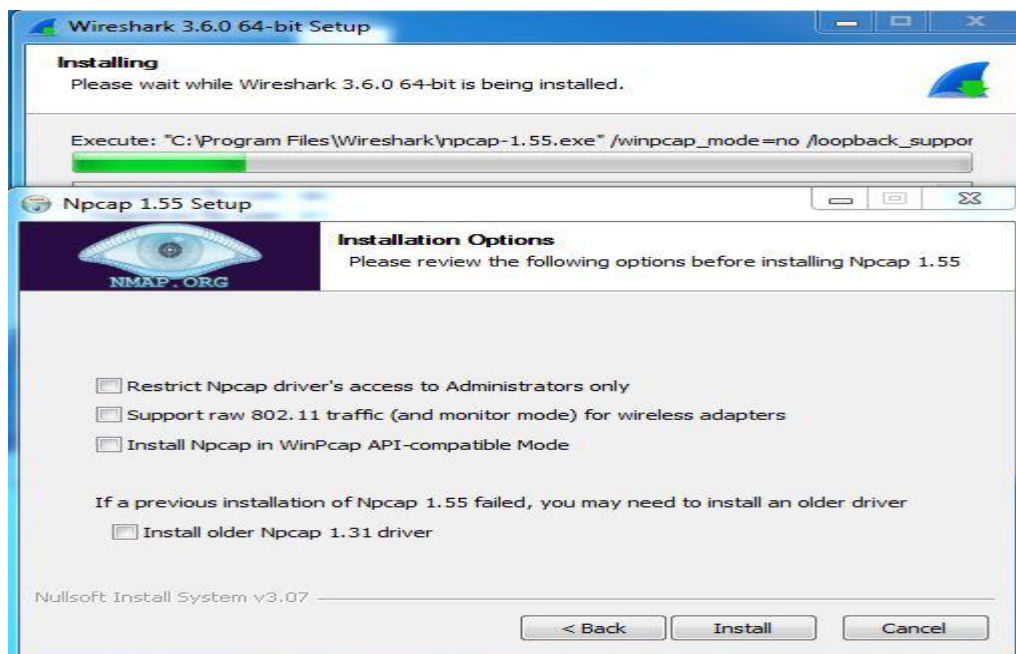
**Step 13:** After this installation process will start.



**Step 14:** This installation will prompt for Npcap installation as already checked so the license agreement of Npcap will appear to click on the *I Agree* button.

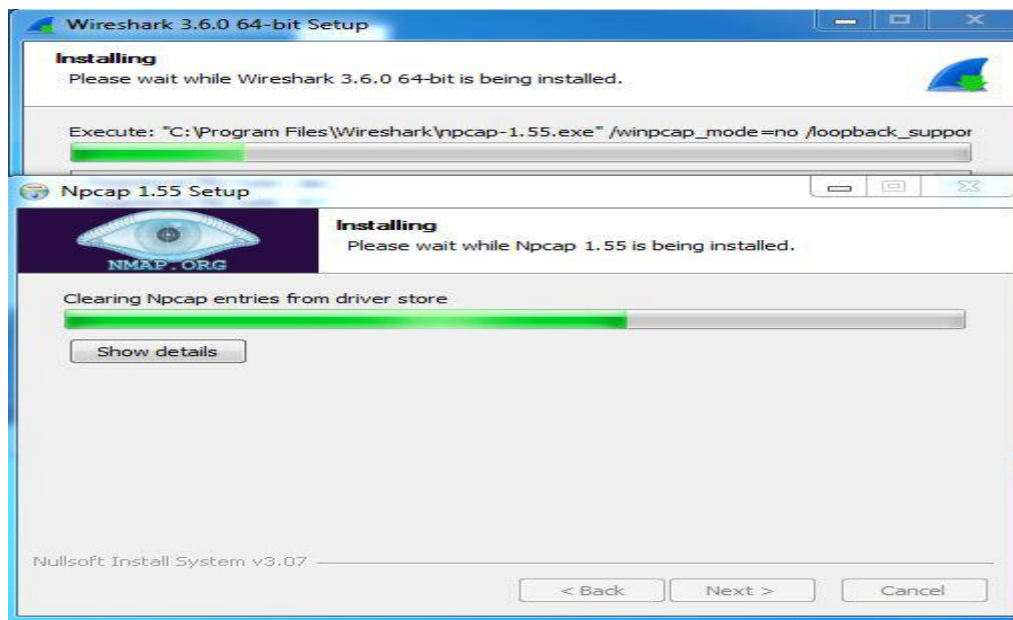


**Step 15:** Next screen is about different installing options of *npcap*, don't do anything click on Install.

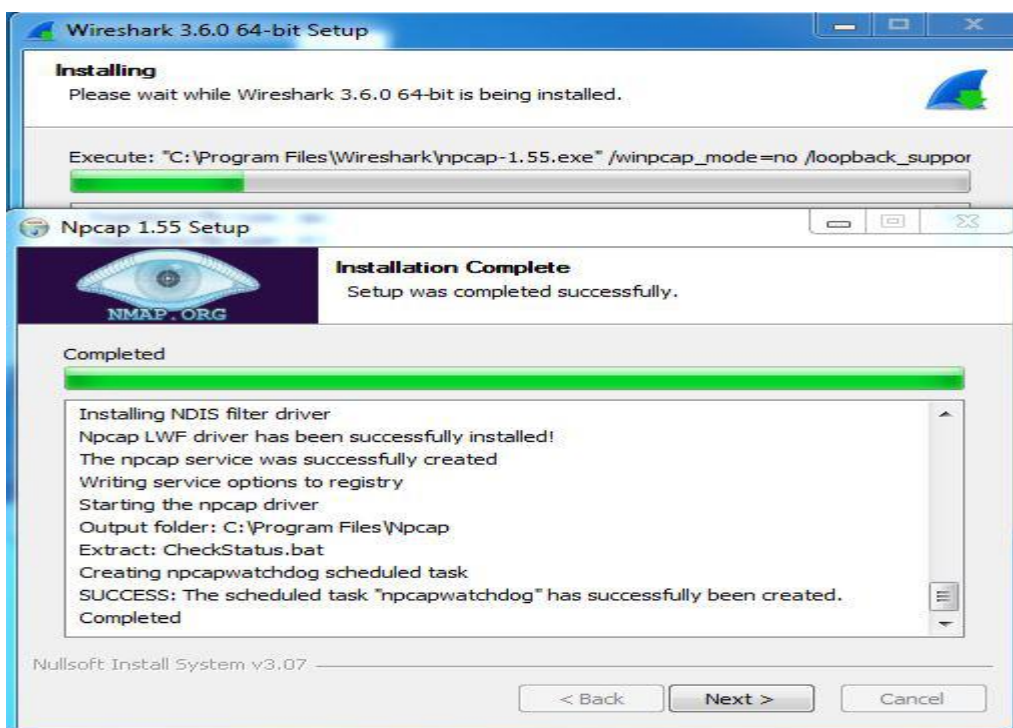




**Step 16:** After this installation process will start which will take only a minute.

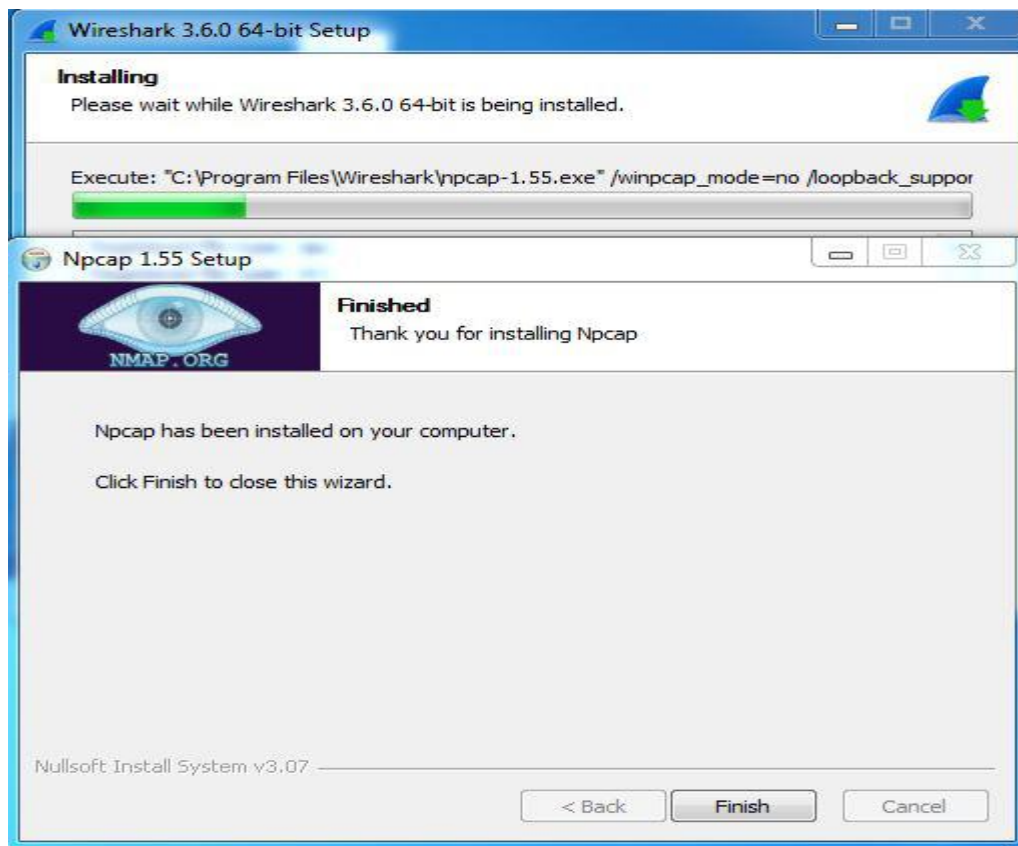


**Step 17:** After this installation process will complete click on the Next button.

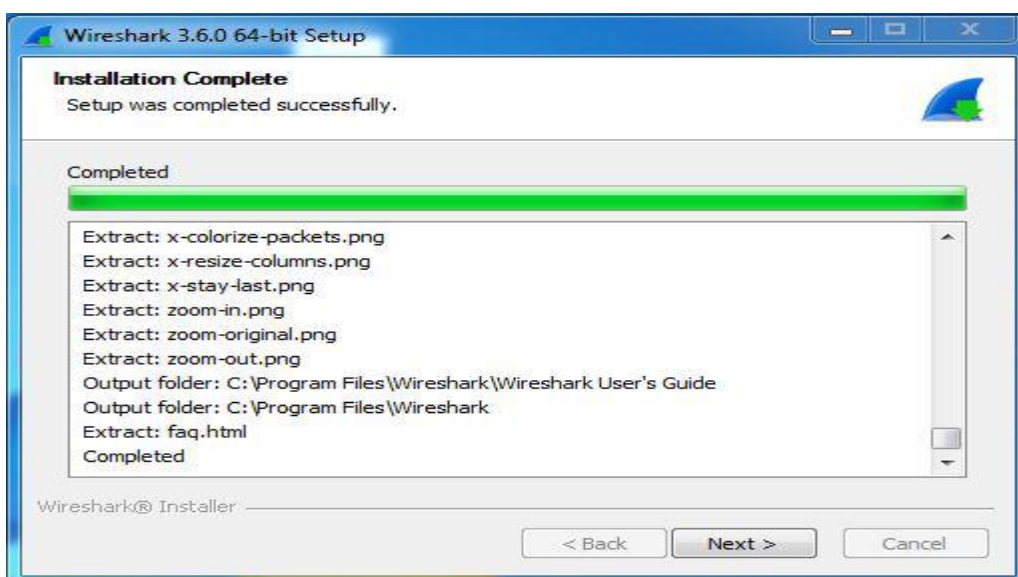




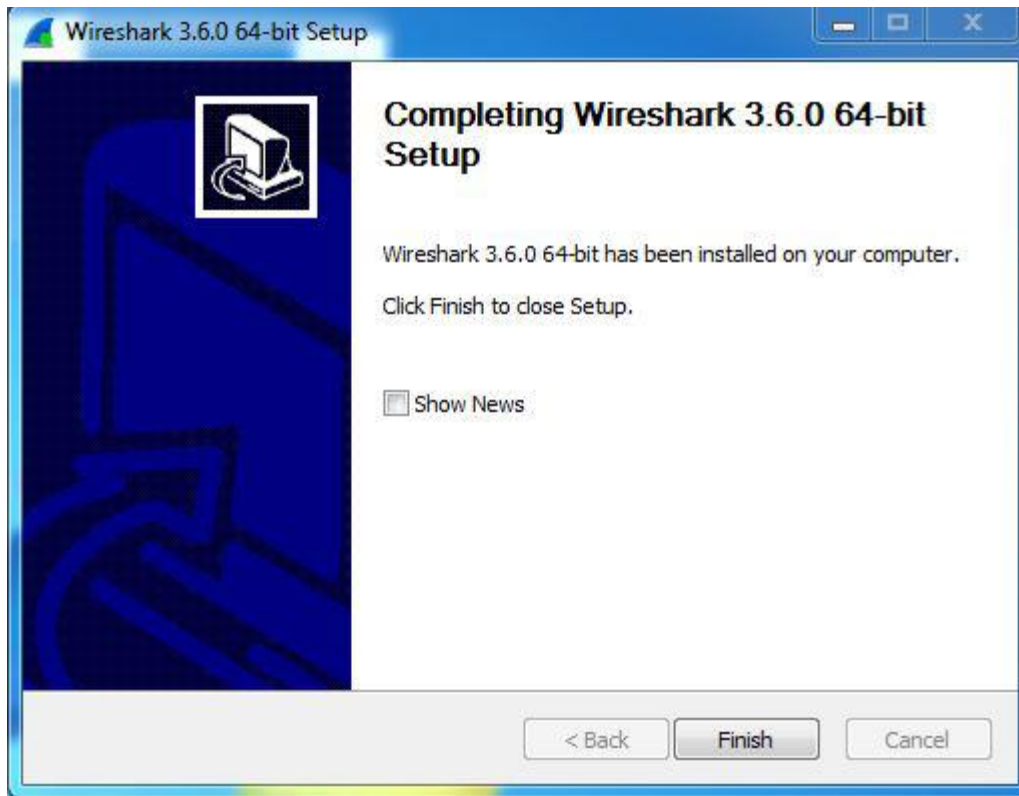
**Step 18:** Click on Finish after the installation process is complete.



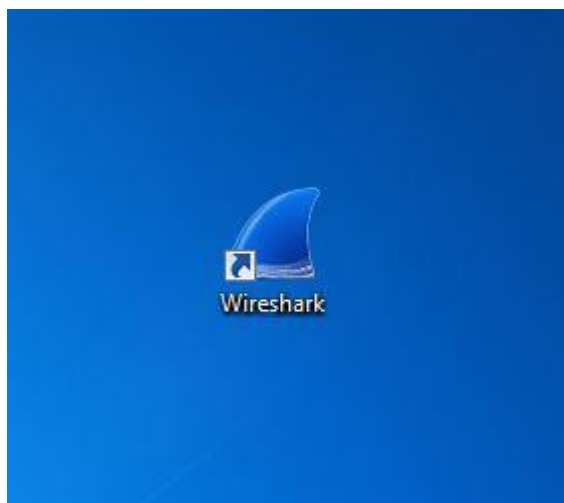
**Step 19:** After this installation process of Wireshark will complete click on the Next button.



**Step 20:** Click on Finish after the installation process of Wireshark is complete.



Wireshark is successfully installed on the system and an icon is created on the desktop as shown below:



Now run the software and see the interface.

### **PRACTICAL-3**

**Aim: write test & debug ceaser cipher algorithm.**

```
#include <stdio.h>

void encrypt(char message[], int shift) {
    int i;
    char ch;
    for (i = 0; message[i] != '\0'; ++i) {
        ch = message[i];
        // Encrypt uppercase letters
        if (ch >= 'A' && ch <= 'Z')
            message[i] = (char)((((ch-'A' + shift) % 26) + 'A'));
        // Encrypt lowercase letters
        else if (ch >= 'a' && ch <= 'z')
            message[i] = (char)((((ch-'a' + shift) % 26) + 'a'));
    }
}

void decrypt(char message[ ], int shift) {
    // Decryption is the same as encryption with a negative shift
    encrypt(message, -shift);
}

int main() {
    char message[100];
```

```
int shift;
// Input message and shift value
printf ("Enter a message: ");
fgets (message, sizeof (message), stdin); // Use fgets instead of gets
printf ("Enter the shift value: ");
scanf ("%d", &shift);
getchar ( ); // Consume the newline character left in the input buffer
// Encrypt and display the encrypted message
encrypt(message, shift);
printf("Encrypted message: %s\n", message);
// Decrypt and display the decrypted message
decrypt(message, shift);
printf("Decrypted message: %s\n", message);
return 0;
}
```

**output:**

```
Output
/tmp/zSyr2w4QXL.o
Enter a message: parul university
Enter the shift value: 2
Encrypted message: rctwn wpkxgtukva

Decrypted message: parul universit_

=== Code Execution Successful ===
```

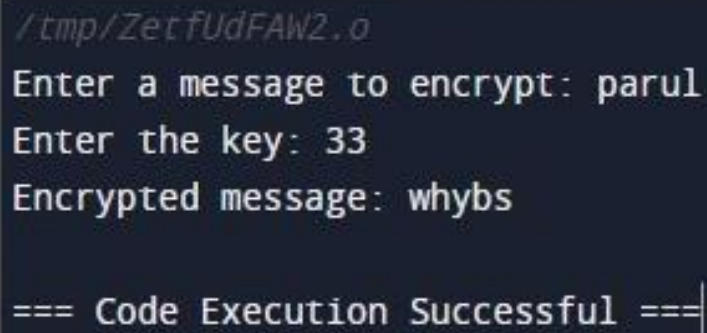
## PRACTICAL – 4

**Aim:-Write algorithm for shiftsfer and solve various example on it.**

### **SOLUTION:**

```
#include<stdio.h>
#include<ctype.h>
int main()
{
    char text[500], ch; int key;
    // Taking user input.
    printf("Enter a message to encrypt: ");
    scanf("%s", text);
    printf("Enter the key: ");
    scanf("%d", & key);
    // Visiting character by character.
    for (int i = 0; text[i] != '\0'; ++i)
        ch = text[i];
    // Check for valid characters.
    if (isalnum(ch))
    {
        //Lowercase characters.
        if (islower(ch))
        {
            ch = (ch - 'a' + key) % 26 + 'a';
        }
    }
    // Invalid character.
```

```
else
{
printf("Invalid Message");
}
// Adding encoded answer.
text[i] = ch;
}
printf("Encrypted message: %s", text);
return 0;
}
```

**Output :**A terminal window with a dark background and light-colored text. The text shows the execution of a program. It starts with a file path, followed by prompts for a message and a key, the resulting encrypted message, and a success confirmation.

```
/tmp/ZetfUdFAW2.o
Enter a message to encrypt: parul
Enter the key: 33
Encrypted message: whybs
=== Code Execution Successful ===
```

## **Practical: 5**

**Aim: Write test and debug for Play fair cipher & solve various examples on it.**

### **SOLUTION:**

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
// Size of the Playfair matrix
#define SIZE 5

// Function to initialize the Playfair matrix with the given
key void initializeMatrix (char key[], char matrix[SIZE][SIZE])
{
    int i, j, k;
    char keyTable[26] = {0};

    // Initialize keyTable to check duplicate characters
    for (k = 0; k < strlen(key); k++)
    {
        keyTable[key[k] - 'A'] = 1;
    }
    char currentChar = 'A';
    // Fill the matrix with the key
    for (i = 0; i < SIZE; i++)
    {
        for (j = 0; j < SIZE; )
        {
            if (keyTable[currentChar - 'A'] != 1)
```

```
{
matrix[i][j] = currentChar;
keyTable[currentChar - 'A'] = 1; j++;
}
currentChar++;
}
}
}
// Function to find the position of a character in the matrix
void findPosition(char matrix[SIZE][SIZE], char ch, int *row, int *col)
{
int i, j;
for (i = 0; i < SIZE; i++)
{
for (j = 0; j < SIZE; j++)
{
if (matrix[i][j] == ch)
{
*row = i; *col = j; return;
}
}
}
}
// Function to apply Playfair encryption
void encrypt(char matrix[SIZE][SIZE], char plaintext[])
{
int i, len = strlen(plaintext);
char encryptedText[len];
for (i = 0; i < len; i += 2)
```



```
{
int row1, col1, row2, col2;
// Find positions of the characters in the matrix
findPosition(matrix, plaintext[i], &row1, &col1);
findPosition(matrix, plaintext[i + 1], &row2, &col2);
// Same row, shift columns to the right
if (row1 == row2)
{
encryptedText[i] = matrix[row1][(col1 + 1) % SIZE];
encryptedText[i + 1] = matrix[row2][(col2 + 1) % SIZE];
}
// Same column, shift rows down
else if (col1 == col2)
{
encryptedText[i] = matrix[(row1 + 1) % SIZE][col1];
encryptedText[i + 1] = matrix[(row2 + 1) % SIZE][col2];
}
// Form a rectangle, swap columns
else
{
encryptedText[i] = matrix[row1][col2];
encryptedText[i + 1] = matrix[row2][col1];
}
}
// Null-terminate the encrypted text
encryptedText[len] = '\0';
// Print the encrypted text
```

```
printf("Encrypted Text: %s\n", encryptedText);
}
int main()
{
char key[25], plaintext[100];
// Input key from the user
printf("Enter the key (uppercase letters): ");
scanf("%s", key);
// Input plaintext from the user
printf("Enter the plaintext (uppercase letters): ");
scanf("%s", plaintext);
// Remove spaces from the plaintext
char sanitizedPlaintext[100];
int j = 0;
for (int i = 0; plaintext[i] != '\0'; i++)
{
if (isalpha(plaintext[i]))
{
sanitizedPlaintext[j++] = toupper(plaintext[i]);
}
}
sanitizedPlaintext[j] = '\0';
charmatrix[SIZE][SIZE];
initializeMatrix(key, matrix);
// Encrypt the plaintext using the Playfair Cipher
encrypt(matrix, sanitizedPlaintext);
```

```
return 0;  
}
```

**Output:**

Output

```
/tmp/U5cW9iELKb.o  
Enter the key (uppercase letters): UNIVERSITY  
Enter the plaintext (uppercase letters): PARUL  
Encrypted Text: MCMCG
```

## **Practical: 6**

**Aim: Write test and debug Hill cipher & solve various examples on it.**

### **SOLUTION:**

```
#include <stdio.h>

#include <stdlib.h>
#define MOD 26
// Function to calculate the determinant of a 2x2 matrix

int calculateDeterminant(int a, int b, int c, int d)

{
return (a * d - b * c + MOD) % MOD;
}
// Function to calculate the modular inverse of a number

int modInverse(int a)

{
int m;
for (m = 1; m < MOD; m++)
{
if ((a * m) % MOD == 1)
{
return m;
}
}
return -1;
}
// If the modular inverse does not exist
```

```
}  
// Function to encrypt a message using Hill Cipher  
void encryptHillCipher(int key[2][2], char message[])  
{  
    int len = strlen(message);  
    // Pad the message with 'X' if its length is odd  
    if (len % 2 != 0)  
    {  
        message[len] = 'X'; len++;  
    }  
    // Convert characters to numeric values (A=0, B=1, ..., Z=25)  
    int numericMessage[len];  
    for (int i = 0; i < len; i++)  
    {  
        numericMessage[i] = message[i] - 'A';  
    }  
    // Encrypt the message  
    for (int i = 0; i < len; i += 2)  
    {  
        int x = numericMessage[i];  
        int y = numericMessage[i + 1];  
        numericMessage[i] = (key[0][0] * x + key[0][1] * y) % MOD; numericMessage[i  
+ 1] = (key[1][0] * x + key[1][1] * y) % MOD;  
    }  
    // Convert numeric values back to characters  
    for (int i = 0; i < len; i++)  
    {
```

```
message[i] = numericMessage[i] + 'A';
}
// Null-terminate the encrypted message
message[len] = '\0';
}

int main() {
// Define the key matrix (2x2)
int key[2][2] = {{6, 24}, {13, 16}};
// Input message (all uppercase letters)
char message[100];
printf("Enter the message (uppercase letters
only):");
scanf("%s", message);
// Encrypt the message using Hill Cipher
encryptHillCipher(key, message);
// Display the encrypted message
printf("Encrypted Message: %s\n", message);
return 0;
}
```

**Output:-****Output**

```
/tmp/F8ZfaUQb65.o
Enter the message (uppercase letters only): HELLO
Encrypted Message: IZSHME
```

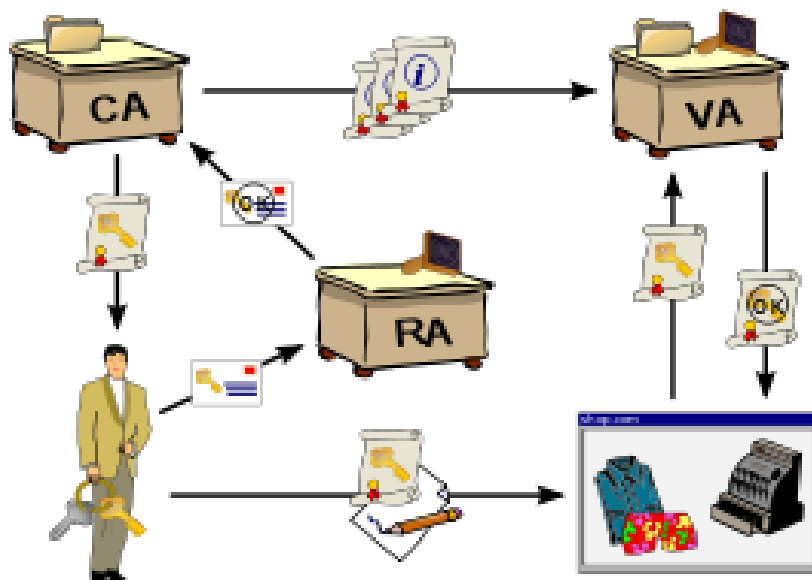
## Practical: 7

### **Aim: Draw diagram of public key infrastructure.**

A public key infrastructure (PKI) supports the distribution, revocation and verification of public keys used for public key encryption, and enables linking of identities with public key certificates. A PKI enables users and systems to securely exchange data over the internet and verify the legitimacy of certificate-holding entities, such as web servers, other authenticated servers and individuals.

PKI certificates include a public key used for encryption and cryptographic authentication of data sent to or from the entity that was issued the certificate. Other information included in a PKI certificate includes identifying information about the certificate holder, about the PKI that issued the certificate, and other data including the certificate's creation date and validity period.

Without PKI, sensitive information can still be encrypted, ensuring confidentiality, and exchanged between two entities, but there would be no assurance of the identity of the other party. Any form of sensitive data exchanged over the internet is reliant on the PKI for enabling the use of public key cryptography because the PKI enables the authenticated exchange of public keys.



## **Elements of PKI**

A typical PKI includes the following key elements:

A trusted party provides the root of trust for all PKI certificates and provides services that can be used to authenticate the identity of individuals, computers and other entities. Usually known as certificate authorities (CA), these entities provide assurance about the parties identified in a PKI certificate. Each CA maintains its own root CA, for use only by the CA.

- A registration authority (RA), often called a subordinate CA, issues PKI certificates. The RA is certified by a root CA and authorized to issue certificates for specific uses permitted by the root.
- A certificate database stores information about issued certificates. In addition to the certificate itself, the database includes validity period and status of each PKI certificate. Certificate revocation is done by updating this database, which must be queried to authenticate any data digitally signed or encrypted with the secret key of the certificate holder.
- A certificate store, which is usually permanently stored on a computer, can also be maintained in memory for applications that do not require that certificates be stored permanently. The certificate store enables programs running on the system to access stored certificates, certificate revocation lists and certificate trust lists.

A CA issues digital certificates to entities and individuals; applicants may be required to verify their identity with increasing degrees of assurance for certificates with increasing levels of validation. The issuing CA digitally signs certificates using its secret key; its public key and digital signature are made available for authentication to all interested parties in a self-signed CA certificate. CAs use the trusted root certificate to create a "chain of trust;" many root certificates are embedded in web browsers so they have builtin trust of those CAs. Web servers, email clients, smartphones and many other types of hardware and software -- including IoT devices -- also support PKI and contain trusted root certificates from the major CAs.

## **PKI certificates**

Along with an entity's or individual's public key, digital certificates contain information about the algorithm used to create the signature, the person or entity identified, the digital signature of the



CA that verified the subject data and issued the certificate, the purpose of the public key encryption, signature and certificate signing, as well as a date range during which the certificate can be considered valid.

While PKI certificates are used for implementing cryptography over web and other internet connections, they are also used for other applications, including individual certification for code signing applications, for authenticating digital transactions and more.

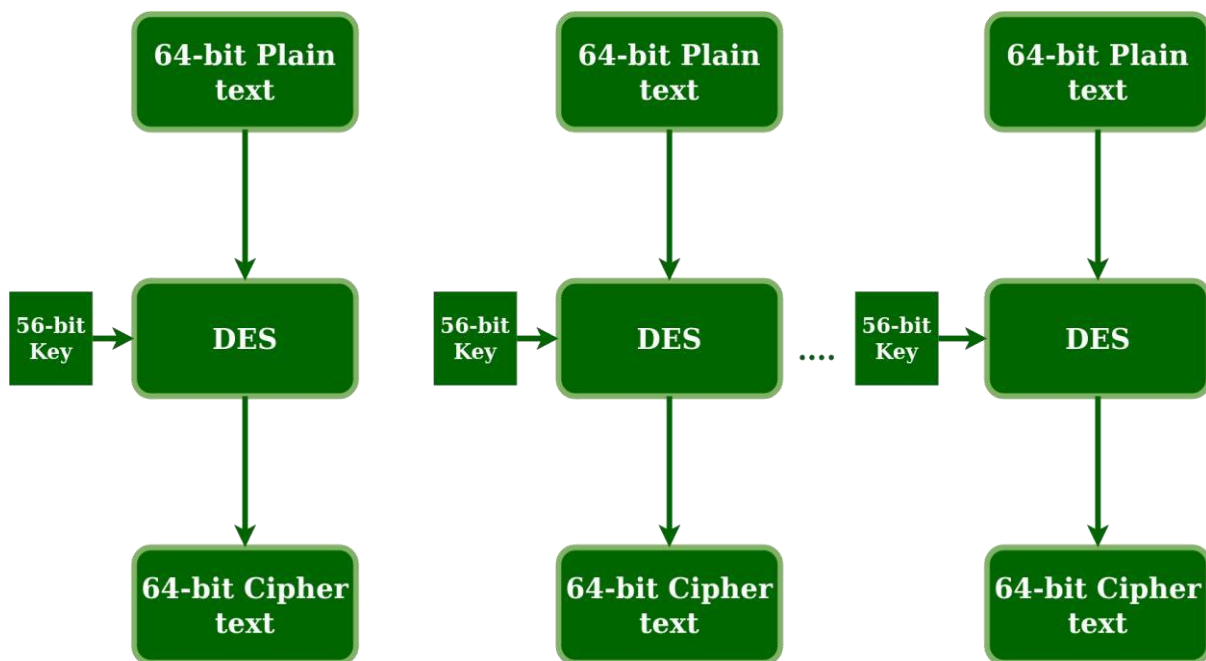
## Practical: 8

### **Aim: Prepare case study on single round of DES.**

Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline.

DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext.

The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits.



We have mentioned that DES uses a 56-bit key. Actually, the initial key consists of 64 bits.

However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Figure - discarding of every 8<sup>th</sup> bit of original key

Thus, the discarding of every 8th bit of the key produces a 56-bit key from the original 64-bit key. DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion).

DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition.

### Single Round of DES:

#### Initial Permutation (IP):

1. **Input:** A 64-bit block of plaintext.
2. **Permutation:** The bits of the plaintext are permuted according to a fixed table, known as the Initial Permutation (IP) table.

#### Key Mixing (Key Schedule):

1. **Input:** A 56-bit key (selected from the original 64-bit key).
2. **Key Compression:** The 56-bit key is compressed and transformed into a 48bit subkey specific to the current round.

#### Expansion Permutation (E):

1. **Input:** A 32-bit half-block (from the previous step).
2. **Expansion:** The 32-bit half-block is expanded to 48 bits using a fixed table.

#### Subkey Mixing (XOR):

1. **Input:** The expanded 48-bit block and the 48-bit subkey for this round.
2. **Operation:** The expanded block is XORed with the round-specific subkey.

#### Substitution (S-boxes):

1. **Input:** The XOR result (48 bits).

2. **Operation:** The 48-bit block is divided into eight 6-bit blocks, each passed through a different S-box (substitution box) producing 32 output bits.

**Permutation (P):**

1. **Input:** The 32-bit output of the S-boxes.
2. **Permutation:** The bits are permuted according to a fixed table called the Permutation (P) table.

**Feistel Function:**

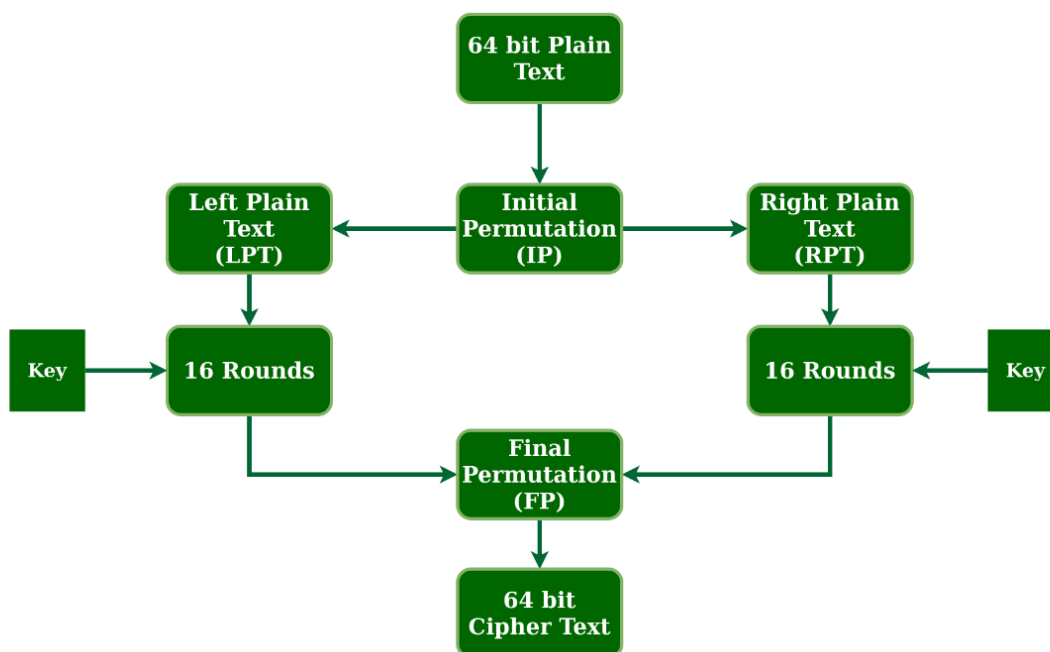
1. **Input:** The 32-bit half-block from the previous round and the 48-bit subkey for this round.
2. **Operation:** The half-block undergoes expansion, XOR with the subkey, substitution through S-boxes, and permutation.

**Final XOR:**

1. **Input:** The output of the Feistel function and the other half of the 64-bit block.
2. **Operation:** The output of the Feistel function is XORed with the other halfblock.

**Round Output:**

The output of the single round consists of two 32-bit halves, which are swapped to prepare for the next round.



steps in DES.

1. In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.
2. The initial permutation is performed on plain text.
3. Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).
4. Now each LPT and RPT go through 16 rounds of the encryption process.
5. In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block.
6. The result of this process produces 64-bit ciphertext.

### **Conclusion:**

This single round process is repeated 16 times for encryption (with different subkeys for each round) or decryption (with subkeys applied in reverse order). DES provides a relatively simple yet effective method of symmetric encryption, though its key length has become too short by modern standards, rendering it vulnerable to bruteforce attacks.

## **Practical: 9**

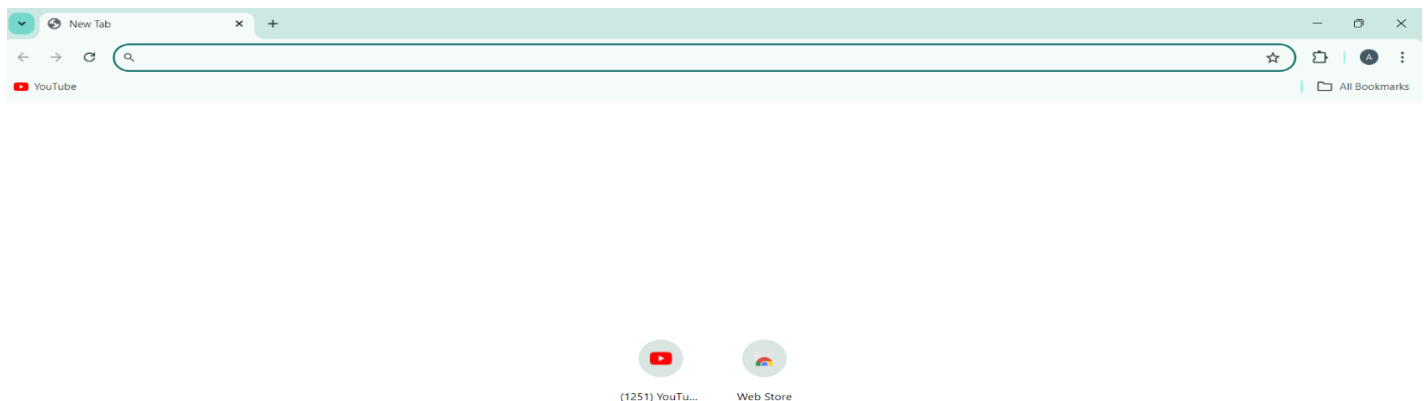
### **Aim: Configure Web browser security settings**

Optimizing your browser's settings is a critical step in using the Internet securely and privately. Today's popular browsers include built-in security features, but users often fail to optimize their browser's security settings on installation. Failing to correctly set up your browser's security features can put you at a higher risk for malware infections and malicious attacks. This installation of our "Cybersecurity 101" series provides our tips for securing several of today's most popular browsers, including Google Chrome, Mozilla Firefox, and Microsoft Internet Explorer. While it is impossible to guarantee complete protection from cyber threats, following these tips will greatly increase the security of your web browser.

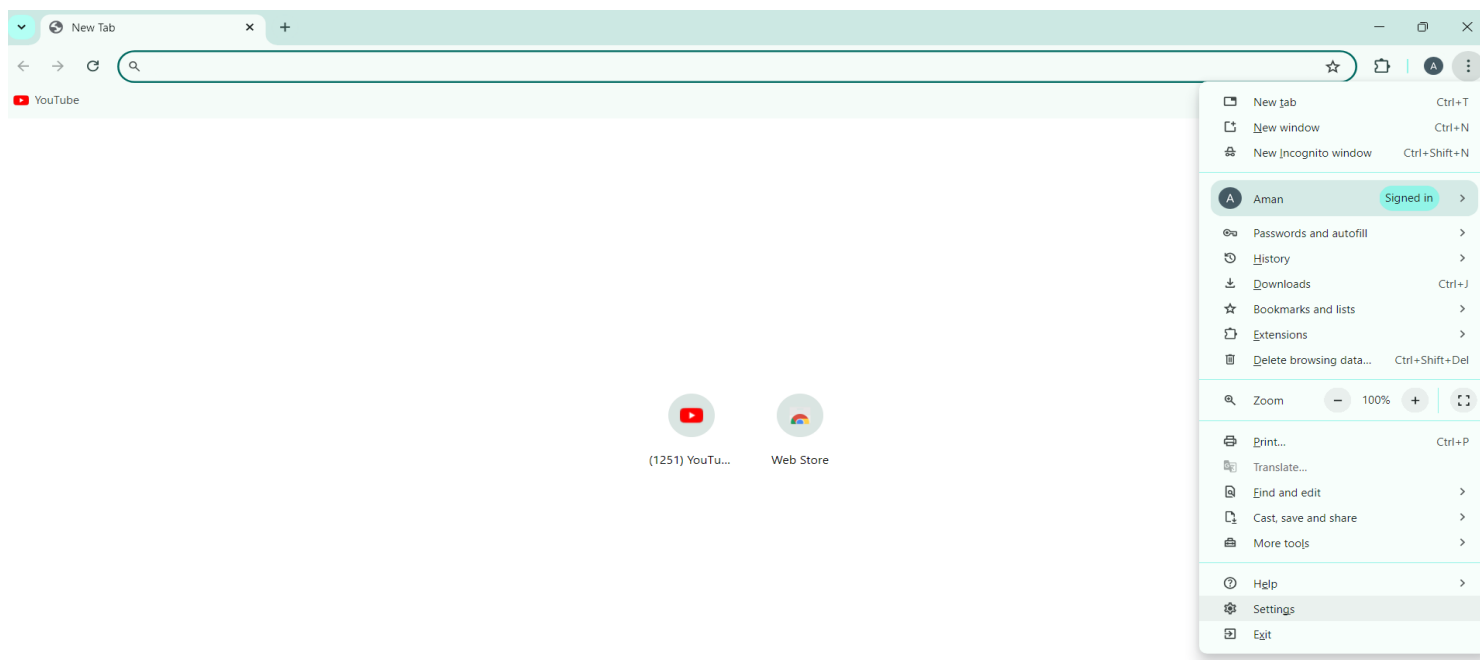
#### **Secure Browsing with Google Chrome**

These settings can be accessed through Chrome's "Advanced Settings" menu or by navigating to "chrome://settings/."

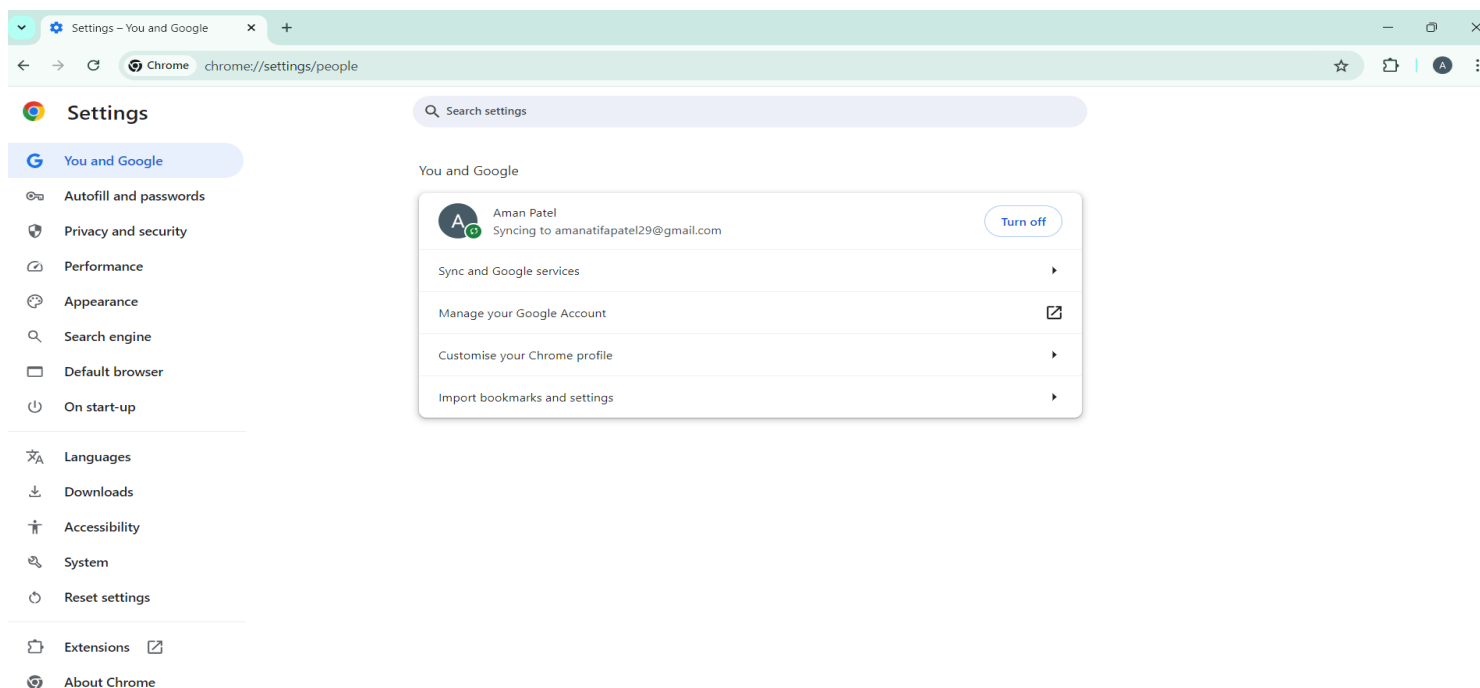
- Open Chrome and click on the three dots in the top-right corner to access the menu.



- Select "Settings" from the dropdown menu.

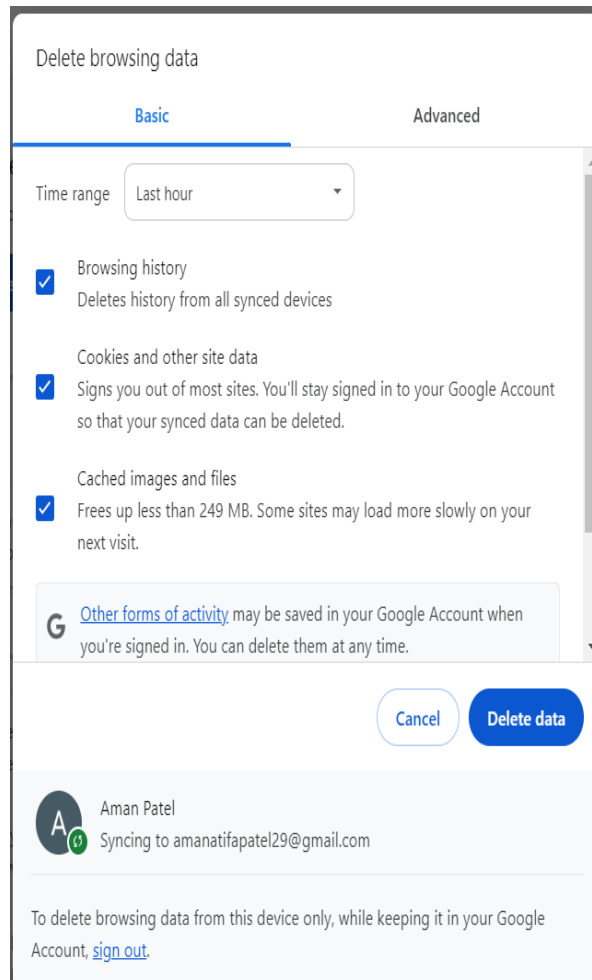


- Scroll down and click on "Privacy and security" in the left-hand menu.











- Here, you can configure settings such as:

- **Clear browsing data:** Regularly clear your browsing history, cookies, and cache to remove any potentially harmful data.




- **Site settings:** Control permissions for websites, such as camera, microphone, and location access.



Permissions	
	Location Sites can ask for your location
	Camera Sites can ask to use your camera
	Microphone Sites can ask to use your microphone
	Notifications Collapse unwanted requests (recommended)
	Embedded content Sites can ask to use info they've saved about you
Additional permissions	
Content	
	Third-party cookies Third-party cookies are allowed
	JavaScript Sites can use JavaScript
	Images Sites can show images

- Security: Enable features like Safe Browsing to protect against phishing and malware.



Safe Browsing	
<input type="radio"/>	Enhanced protection Real-time, proactive protection against dangerous sites, downloads and extensions that's based on your browsing data being sent to Google
<input checked="" type="radio"/>	Standard protection Protects against sites, downloads and extensions that are known to be dangerous. When you visit a site, Chrome sends an obfuscated portion of the URL to Google through a privacy server that hides your IP address. If a site does something suspicious, full URLs and bits of page content are also sent.
	Help improve security on the web for everyone Sends URLs of some pages that you visit, limited system information and some page content to Google, to help discover new threats and protect everyone on the web.
	Warn you if a password was compromised in a data breach When you use a password, Chrome warns you if it has been published online. When doing this, your passwords and usernames are encrypted, so they can't be read by anyone, including Google.
	No protection (not recommended)

- **Enable phishing and malware protection:** Make sure that Chrome's phishing and malware protection feature is enabled under the "Privacy" section. This feature will warn you if a site you're trying to visit may be phishing or contain malware.
- **Turn off instant search:** The Instant search feature should be turned off for optimal security. While it offers some convenience in searching, having this feature enabled means that anything you type in the address bar is instantly sent to Google.
- **Don't sync:** Disconnect your email account from your browser under the "Personal Stuff" tab. Syncing your email account with your Chrome browser means that personal information such as passwords, autofill data, preferences, and more is stored on Google's servers. If you must use sync, select the "Encrypt all synced data" option and create a unique passphrase for encryption.

## **Practical: 10**

### **Aim: Configure your e-mail account against various Threats.**

Malware, short for malicious software, is frequently spread via e-mail on home networks. This type of security threat to home networks — and computers in general — may even appear to come from someone you know and trust. E-mail also has some original threats of its own, including spam, spoofing, and phishing attacks.

### **E-MAIL SECURITY THREAT: SPAM**

Spam is the scourge of e-mail around the world. At times, it makes up as much as 95 percent of all e-mail on the Internet! Spammers get e-mail addresses from newsgroups, unscrupulous Web site operators who sell e-mail addresses to them, and malware that harvests e-mail addresses from hacked e-mail accounts. Spammers also guess e-mail addresses and sometimes just get lucky.

Spam causes a number of issues, including these:

- **Network congestion:** Spam clogs your network pipes. Although e-mail is relatively small in size, receiving enough of it will cause congestion on your network. Worse yet, if your computer has become part of a botnet, you will definitely see a negative effect on your network as you could be unknowingly *sending* thousands of spam e-mails to others!
- **Distraction and clutter:** Because spam can account for a large volume of e-mail, legitimate e-mails may get buried in your inbox or inadvertently deleted along with all the spam.
- **Malware:** A large proportion of spam contains malware, or links to Web sites that contain malware.

The best protection against spam (other than not using e-mail at all) is to use a spam filter. Of course, this may not be an option on your home network (although some Internet service providers offer spam filtering as an additional service). If you don't have a spam filter, you should also use any junk mail filtering options available in your e-mail software.

## **E-MAIL SECURITY THREAT: SPOOFING**

E-mail spoofing occurs when an attacker sends you an e-mail pretending to be someone you know. Spoofing is analogous to sending a letter to someone and forging the return address on the envelope. Unfortunately, e-mail spoofing is easy to do, and very difficult to trace to its real sender.

## **E-MAIL SECURITY THREAT: PHISHING**

Phishing (pronounced like *fishing*) e-mails have become a favorite weapon of identity thieves, and they are becoming increasingly difficult to spot. Most phishing e-mails purport to be from a banking or other financial institution (as well as Web sites such as PayPal), and every once in a while they get lucky and actually send an e-mail pretending to be from your bank.

**Following features are provided for e-mail account against various Threats with example:**

### **1. Strong Password:**

- Use a complex password that includes a combination of letters, numbers, and special characters.
- Avoid using easily guessable information like birthdays or common words.

Example: Instead of using "password123," use something like "P@ssw0rd!23".

### **2. Two-Factor Authentication (2FA):**

- Enable two-factor authentication for an extra layer of security.
- This typically involves receiving a code on your phone or email to verify your identity during login attempts.

Example: Link your phone number to your email account so that a verification code is sent via SMS whenever you log in from a new device.

**3. Email Encryption:**

- Use email encryption tools to ensure that your messages are secure and cannot be intercepted by unauthorized parties.

Example: Use PGP (Pretty Good Privacy) or S/MIME (Secure/Multipurpose Internet Mail Extensions) to encrypt your emails before sending them.

**4. Avoid Clicking Suspicious Links or Attachments:**

- Exercise caution when clicking on links or downloading attachments from unknown or suspicious senders.
- Example: If you receive an unexpected email with a link or attachment from an unknown sender, verify the sender's identity before clicking or downloading anything.

## **Practical: 11**

**Aim: Study of the features of firewall in providing network security and to set Firewall Security in windows.**

### **1. Threat Prevention**

- The longer that a cyber threat has access to an organization's network, the more expensive it will be to remediate it. Cyberattacks can cause damage and additional expense in a number of different ways.
- Exfiltration of sensitive data can result in legal and regulatory penalties, ransomware can decrease productivity and cause a loss of profits, and even simple malware often has persistence mechanisms designed to make it difficult and timeconsuming to remove from a system.

### **2. Unified Security Management**

- Organizations must cope with rapidly increasing network security complexity. Most companies' networks are growing larger and more complex as mobile devices, cloud deployments, and Internet of Things (IoT) devices join traditional user workstations and on-premises servers on the corporate network.
- At the same time, cyber threats are becoming more sophisticated and numerous. As a result, companies must deploy, monitor, and maintain a growing array of security solutions to manage their cyber risk.

### **3.Application and Identity-Based Inspection**

- Digital transformation efforts mean that an organization's network landscape is constantly evolving. New applications are deployed on the corporate network to accomplish certain goals, and others are phased out when they become obsolete.
- Different applications require different policies. Some applications may be highpriority traffic, while others should be blocked, throttled, or otherwise managed on the network.
- An organization's next-generation firewall should be capable of identifying the application that generates a particular stream of traffic and applying applicationspecific policies to that traffic.

### **4.Hybrid Cloud Support**

- Almost all organizations are using cloud computing, and the vast majority are using a

hybrid cloud deployment. Private and public cloud deployments have different security requirements, and it is necessary for an organization to be able to enforce consistent security policies across cloud-based environments hosted by multiple vendors.

## 5. Scalable Performance

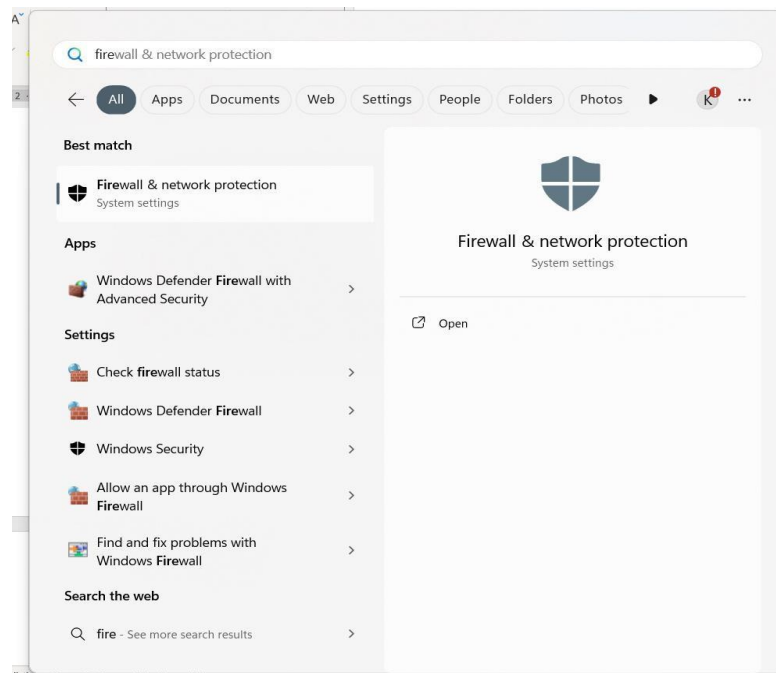
- Many organizations have transitioned to cloud-based infrastructure due to its increased scalability and flexibility. Ultimately, we want the benefits of the cloud, in the cloud and on-premises.
- In the cloud this simply means choosing a NGFW template. In regards to onpremises, this means looking beyond legacy HA clustering solutions.

## Configure Windows Firewall

You can customize most settings of your Windows Firewall through the left pane of the Firewall applet in Control Panel.

### 1. Turn on Windows Firewall

This setting is selected by default. When Windows Firewall is On, most programs are blocked from communicating through the firewall. Clicking on the **Turn Firewall On or Off** will let you enable or disable the Windows Firewall on your computer.





## **2. Block all incoming firewall connections, including those in the list of allowed programs**

This setting blocks all unsolicited attempts to connect to your computer. Use this setting when you need maximum protection for your computer, such as when you connect to a public network in a hotel or airport, or when a computer worm is spreading over the Internet. With this setting, you are not notified when Windows Firewall blocks programs, and programs in the list of allowed programs are ignored. When you block all incoming connections, you can still view most web pages, send and receive an e-mail, and send and receive instant messages.

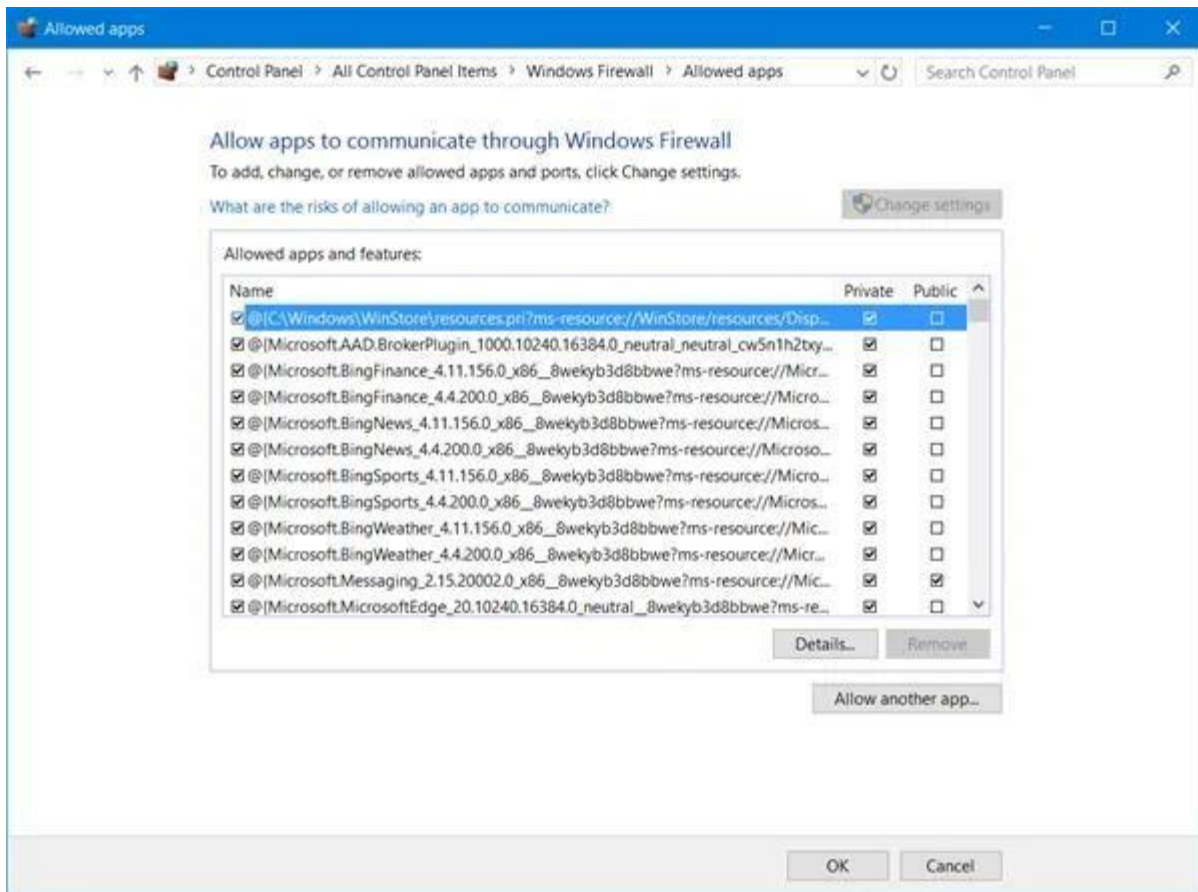
## **3. Turn off Windows Firewall**

Avoid using this setting unless you have another firewall running on your computer. Turning off Windows Firewall might make your computer more vulnerable to damage from hackers and malicious software. Clicking on the **Turn Firewall On or Off** will let you enable or disable the Windows Firewall on your computer.

## **4. Block or Allow Programs through the Windows Firewall**

By default, most programs are blocked by Windows Firewall to help make your computer more secure. To work properly, some programs might require you to allow them to communicate through the firewall. Here's how to do that:

Click **Allow an app or feature through Windows Firewall**. If you are prompted for an administrator password or confirmation, type the password or provide confirmation.



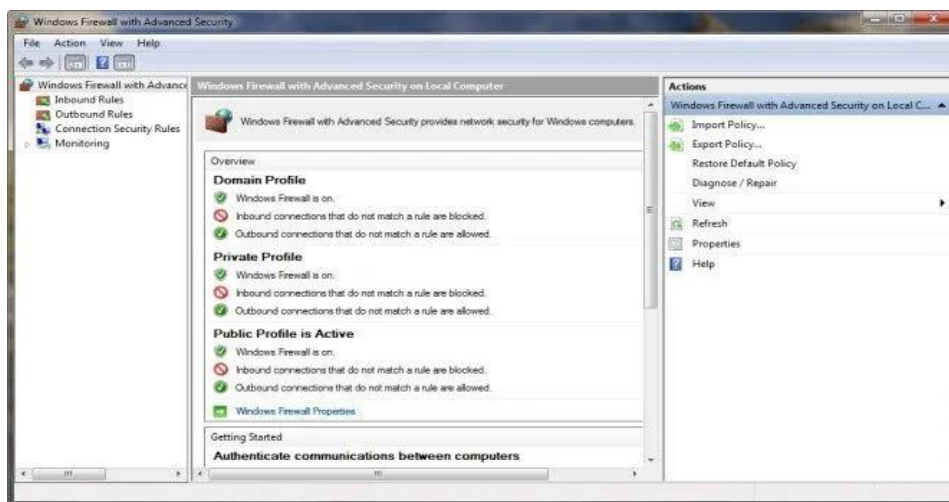
Select the check box next to the program you want to allow, select the network location types you want to allow communication on, and then click OK.

If you want to allow a program to communicate through the firewall, you can add it to the list of allowed programs. For example, you might not be able to send photos in an instant message until you add the instant messaging program to the list of allowed programs. To add or remove a program to the list, click on the Allow an app or feature through Windows Firewall link to open the following panel, where you will be able to get more details about allowed programs and allow another app to communicate through the firewall.

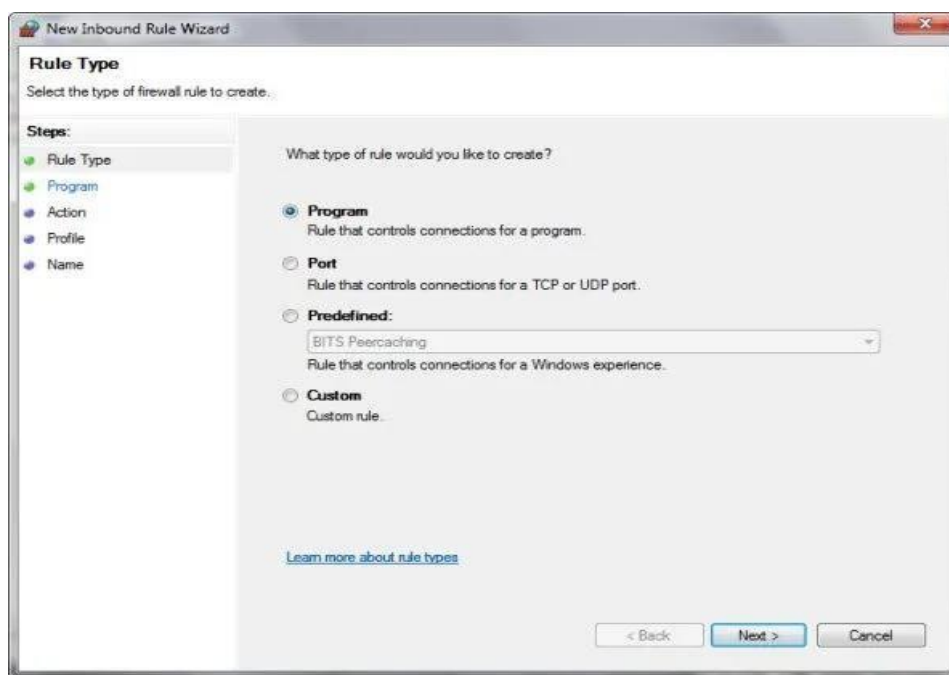
## 5. How to open a port in Windows Firewall

You can also block or open a Port in Windows Firewall. If Windows Firewall is blocking a program and you want to allow that program to communicate through the firewall, you can usually do that by selecting the program in the list of allowed programs (also called the exceptions list) in the Windows Firewall. To learn how to do this, see Allow a program to communicate through Windows Firewall.

Click to open Windows Firewall. In the left pane, click Advanced settings.



In the Windows Firewall with Advanced Security dialog box, in the left pane, click **Inbound Rules**, and then, in the right pane, click **New Rule**.



follow the instructions on your screen to its logical conclusion.

## **Practical: 12**

**Aim: Analysis the Security Vulnerabilities of E-commerce services.**

### **Phishing and Malware**

Phishing scams remain popular with hackers despite companies' educational and awareness efforts. In this method, a hacker sends an email to an employee, often posing as a colleague and trying to persuade the employee to click on a malicious link or reveal sensitive information like passwords or credit card numbers.

Phishing is a common technique for installing malware on a device; once an employee has clicked the link, the malware can infect your system and begin accessing your sensitive data.

### **Bad Bots**

Bots can make your life easier by automating simple tasks, but they can make hackers' jobs easier too. Cybercriminals increasingly use bots to harvest data, engage in price scraping, or perform other malicious actions that could harm your security and your business.

Such attacks include distributed denial of service (DDOS) attacks, where bots are deployed to overwhelm your site's capacity with traffic, allowing hackers to access your site while your focus is elsewhere.

Bots are also frequently used in brute force attacks, where algorithms methodically guess every possible password until they find the one that finally succeeds.

### **Cross-Site Scripting and SQL Injections**

Both SQL injections and Cross-Site Scripting (XSS attacks) seek to exploit existing vulnerabilities in your site with an injection of malicious code.

SQL injection occurs when hackers use a site's entry forms (such as email or password fields) to inject malicious code into your online store. This code is then used to access and manipulate sensitive databases with information like phone numbers and credit card information.

Cross-Site Scripting works in a similar way but targets web applications rather than website forms.

### **Resolving E-Commerce Security Issues**

In the face of these threats, it's important to perform due diligence and protect your company (and your customers' information) to the best of your ability.

The following techniques can help you improve your website security and mitigate losses that may occur.

### **Security Hygiene**

Basic security measures can go a long way toward protecting your company from cyberattacks. All accounts should be secured with strong, unique passwords that are changed frequently.

Multi-factor authentication is another popular way of adding another layer of security to your accounts.

You can also arrange to receive notifications every time your system is accessed from an unknown IP address, which can alert you to potential breaches.

## **Software Solutions**

There are a number of software solutions and defenses you can implement to prevent security issues.

Hosting your site on an e-commerce platform is often helpful, since the security is included by the server and the risk is therefore shared.

Firewalls and anti-virus software can help keep malicious actors away from your network, and SSL certificates will allow you to encrypt sensitive data in motion.

