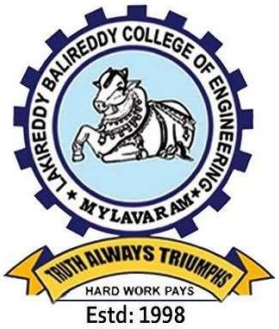
LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)



Department of Computer Science & Engineering

20CS61 – Information Security Lab Record

Name of the Student: ……

Registered Number: ……

Branch & Section: …… &……/Sec

Academic Year: 2023 – 24

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

CERTIFICATE

This is to certify that this is a bonafide record of the practical work done by Mr./Ms ……, bearing Regd. Num.: 21761A05……. of B.Tech ….. Semester, ……. Branch, …… Section in the 20CS61 – Information Security Laboratory during the Academic Year: 2023 – 24.

No. of Experiments/Modules held: 9

No. of Experiments Done: ……..

Date: ….. / …… / 2024 **Signature of the Faculty**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Vision of the Department

The Computer Science & Engineering aims at providing continuously stimulating educational environment to its students for attaining their professional goals and meet the global challenges.

Mission of the Department

- **DM1:** To develop a strong theoretical and practical background across the computer science discipline with an emphasis on problem solving.
- **DM2:** To inculcate professional behaviour with strong ethical values, leadership qualities, innovative thinking and analytical abilities into the student.
- **DM3:** Expose the students to cutting edge technologies which enhance their employability and knowledge.
- **DM4:** Facilitate the faculty to keep track of latest developments in their research areas and encourage the faculty to foster the healthy interaction with industry.

Program Educational Objectives (PEOs)

- **PEO1:** Pursue higher education, entrepreneurship and research to compete at global level.
- **PEO2:** Design and develop products innovatively in computer science and engineering and in other allied fields.
- **PEO3:** Function effectively as individuals and as members of a team in the conduct of interdisciplinary projects; and even at all the levels with ethics and necessary attitude.
- **PEO4:** Serve ever-changing needs of society with a pragmatic perception.

PROGRAMME OUTCOMES (POs):

PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO 6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend

	and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAMME SPECIFIC OUTCOMES (PSOs):

PSO 1	The ability to apply Software Engineering practices and strategies in software project development using open-source programming environment for the success of organization.
PSO 2	The ability to design and develop computer programs in networking, web applications andIoT as per the society needs.
PSO 3	To inculcate an ability to analyze, design and implement database applications.

INDEX

6	**Study about Wireshark packet sniffer tool in promiscuous and non-promiscuous mode.**				
7	**Download and install nmap. Use it with different options to scan open ports, do a ping scan, tcp port scan, udp port scan**				
8	**Iptables in linux.**				
9	**Demonstrate intrusion detection system (ids) using any tool (snort or any other s/w).**				

1. Implement any two Substitution Techniques.

AIM: Write a Java program to perform encryption and decryption using the following algorithms:

a) **Ceaser Cipher**

ALGORITHM

1. In Ceaser Cipher each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
2. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
3. The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,Z = 25.
4. Encryption of a letter x by a shift n can be described mathematically as,

En(x) = (x + n) mod26

5. Decryption is performed similarly,

Dn (x)=(x - n) mod26

PROGRAM:

```
class caesarCipher {
 public static String encode(String enc, int offset) {
offset = offset % 26 + 26;
 StringBuilder encoded = new StringBuilder();
 for (char i : enc.toCharArray())
{ if (Character.isLetter(i)) { if
(Character.isUpperCase(i)) {
 encoded.append((char) ('A' + (i - 'A' + offset) % 26));
 } else {
 encoded.append((char) ('a' + (i - 'a' + offset) % 26));
 }
 } else {
 encoded.append(i);
 }
 }
 return encoded.toString();
 }
 public static String decode(String enc, int offset) {
return encode(enc, 26 - offset);
 }
 public static void main(String[] args) throws java.lang.Exception {
 String msg = "LBRCE CSE";
 System.out.println("Simulating Caesar Cipher\n ---------------------------- ");
 System.out.println("Input : " + msg);
 System.out.printf("Encrypted Message : ");
 System.out.println(caesarCipher.encode(msg, 3));
 System.out.printf("Decrypted Message : ");
 System.out.println(caesarCipher.decode(caesarCipher.encode(msg, 3), 3));
 } }
```

Output:

Simulating Caesar Cipher

Input : LBRCE CSE

Encrypted Message : OEUFH FVH

Decrypted Message : LBRCE CSE

RESULT:

Thus the program for ceaser cipher encryption and decryption algorithm has been implemented and the output verified successfully.

b) Playfair Cipher

AIM:

To implement a program to encrypt a plain text and decrypt a cipher text using play fair Cipher substitution technique.

ALGORITHM:

1. To encrypt a message, one would break the message into digrams (groups of 2 letters)
2. For example, "HelloWorld" becomes "HE LL OW OR LD".
3. These digrams will be substituted using the key table.
4. Since encryption requires pairs of letters, messages with an odd number of characters usually append an uncommon letter, such as "X", to complete the final digram.
5. The two letters of the digram are considered opposite corners of a rectangle in the key table. To perform the substitution, apply the following 4 rules, in order, to each pair of letters in the plaintext:

PROGRAM:

playfairCipher.java import java.awt.Point; class playfairCipher { private static char[][] charTable; private static Point[] positions; private static String prepareText(String s, boolean chgJtoI) { s = s.toUpperCase().replaceAll("[^A-Z]", "");

return chgJtoI ? s.replace("J", "I") : s.replace("Q", "");

}

```
 private static void createTbl(String key, boolean chgJtoI)
{ charTable = new char[5][5];  positions = new Point[26];
 String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
chgJtoI); int len = s.length(); for (int i = 0,
k = 0; i < len; i++) { char c = s.charAt(i); if
(positions[c - 'A'] == null) { charTable[k /
5][k % 5] = c; positions[c - 'A'] = new
Point(k % 5, k / 5); k++;
 }
 }
 }
 private static String codec(StringBuilder txt, int dir)
{ int len = txt.length(); for (int i = 0; i < len; i += 2) {
char a = txt.charAt(i); char b = txt.charAt(i + 1); int
row1 = positions[a - 'A'].y; int row2 = positions[b -
'A'].y; int col1 = positions[a - 'A'].x; int col2 =
positions[b - 'A'].x; if (row1 == row2) { col1 = (col1 +
dir) % 5; col2 = (col2 + dir) % 5; } else if (col1 ==
col2) { row1 = (row1 + dir) % 5; row2 = (row2 + dir)
% 5;
 } else { int tmp
 = col1; col1 =
 col2; col2 =
 tmp;
 }
 txt.setCharAt(i, charTable[row1][col1]);
txt.setCharAt(i + 1, charTable[row2][col2]);
 }
 return txt.toString();
 }
```

```
private static String encode(String s) {
StringBuilder sb = new StringBuilder(s);
for (int i = 0; i < sb.length(); i += 2) {
 if (i == sb.length() - 1) {
          sb.append(sb.length() % 2 == 1 ? 'X' : "");
 }
 else if (sb.charAt(i) == sb.charAt(i + 1)) {
          sb.insert(i + 1, 'X');
 }
 }
 return codec(sb, 1);
 }
 private static String decode(String s) {
return codec(new StringBuilder(s), 4);
 }
 public static void main(String[] args) throws java.lang.Exception {
 String key = "CSE";
 String txt = "Security Lab"; /* make sure string length is even */
/* change J to I */
  boolean chgJtoI = true;  createTbl(key, chgJtoI);
 String enc = encode(prepareText(txt, chgJtoI));
 System.out.println("Simulating Playfair Cipher\n ---------- ");
System.out.println("Input Message : " + txt);
 System.out.println("Encrypted Message : " + enc);
 System.out.println("Decrypted Message : " + decode(enc));
 }
}
```

OUTPUT:

Simulating Playfair Cipher

Input Message : Security Lab

Encrypted Message : EABPUGYANSEZ

Decrypted Message : SECURITYLABX

RESULT:

Thus the program for playfair cipher encryption and decryption algorithm has been implemented and the output verified successfully.

2. Implement any two Transposition Techniques

a) Rail Fence Cipher Transposition Technique

AIM:

To implement a program for encryption and decryption using rail fence transposition technique.

ALGORITHM:

1. In the rail fence cipher, the plaintext is written downwards and diagonally on successive "rails" of an imaginary fence, then moving up when we reach the bottom rail.
2. When we reach the top rail, the message is written downwards again until the whole plaintext is written out.
3. The message is then read off in rows.

PROGRAM:

railFenceCipher.java class

```
railfenceCipherHelper {
 int depth;
String encode(String msg, int depth) throws Exception {
int r = depth;
int l = msg.length();
int c = l / depth;
int k = 0;
char mat[][] = new char[r][c];
 String enc = "";
 for (int i = 0; i < c; i++)
 { for (int j = 0; j < r; j++) {
if (k != l) {
mat[j][i] = msg.charAt(k++);
} else {
mat[j][i] = 'X';
}
```

```
}
for (int i = 0; i < r; i++)
{
for (int j = 0; j < c; j++)
{
enc += mat[i][j];
}
}
return enc;
}
String decode(String encmsg, int depth) throws Exception {
int r = depth; int l = encmsg.length(); int c = l / depth; int
k = 0; char mat[][] = new char[r][c]; String dec = ""; for (int i
= 0; i < r; i++) { for (int j = 0; j < c; j++) { mat[i][j] =
encmsg.charAt(k++);
}
}
for (int i = 0; i < c; i++) { for
(int j = 0; j < r; j++) { dec
+= mat[j][i];
}
}
return dec;
}
}
class railFenceCipher { public static void main(String[] args)
throws java.lang.Exception { railfenceCipherHelper rf = new
railfenceCipherHelper();
String msg, enc, dec;
msg = "INFORMATION SECURITY";
```

```
int depth = 2; enc =
rf.encode(msg, depth); dec =
rf.decode(enc, depth);
System.out.println("Simulating Railfence Cipher\n ----------");
System.out.println("Input Message : " + msg);
System.out.println("Encrypted Message : " + enc);
System.out.printf("Decrypted Message : " + dec);
}
}
```

OUTPUT:

```
Simulating Railfence Cipher
----------
Input Message : INFORMATION SECURITY
Encrypted Message : IFRAINSCRTNOMTO EUIY
Decrypted Message : INFORMATION SECURITY
```

RESULT:

Thus the java program for Rail Fence Transposition Technique has been implemented and the output verified successfully.

b) Columnar Transformation Technique

AIM:

To implement a program for encryption and decryption by using row and column transformation technique.

ALGORITHM:

1. Consider the plain text hello world, and let us apply the simple columnar transposition technique as shown below

H	e	l	l
O	w	o	r
L	d		

2. The plain text characters are placed horizontally and the cipher text is created with vertical format as: holewdlo lr.

3. Now, the receiver has to use the same table to decrypt the cipher text to plain text.

PROGRAM:

TransCipher.java

```
import java.util.*; class TransCipher {
public static void main(String args[]) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the plain text");
String pl = sc.nextLine(); sc.close();
String s = ""; int start = 0; for
(int i = 0; i < pl.length(); i++) { if
(pl.charAt(i) == ' ') { s = s +
pl.substring(start, i); start = i +
1;
}
}
s = s + pl.substring(start);

System.out.print(s);
System.out.println(); // end of
space deletion int k =
s.length(); int l = 0; int col = 4;
int row = s.length() / col; char
ch[][] = new char[row][col]; for
(int i = 0; i < row; i++) { for (int
j = 0; j < col; j++) { if (l < k) {
ch[i][j] = s.charAt(l); l++;
} else { ch[i][j]
= '#';
}
}
}
```

```
// arranged in matrix char
trans[][] = new char[col][row]; for
(int i = 0; i < row; i++) { for (int j =
0; j < col; j++) { trans[j][i] = ch[i][j];
}
}
for (int i = 0; i < col; i++) { for
(int j = 0; j < row; j++) {
System.out.print(trans[i][j]);
}
}
// display
System.out.println();
}
}
```

OUTPUT:

Enter the plain text information

security informationsecurity

irienmocfanuotsr

RESULT:

Thus the java program for Row and Column Transposition Technique has been implemented and the output verified successfully.

3. **Implement any two Symmetric algorithms.**

PROGRAM:

a) **Aes.java**

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
public class AES {
private static SecretKeySpec secretKey;
private static byte[] key;
public static void setKey(String myKey) {
MessageDigest sha = null;
try {
key = myKey.getBytes("UTF-8");
sha = MessageDigest.getInstance("SHA-1");
key = sha.digest(key);
key = Arrays.copyOf(key, 16);
secretKey = new SecretKeySpec(key, "AES");
} catch (NoSuchAlgorithmException e) {
e.printStackTrace();
} catch (UnsupportedEncodingException e) {
e.printStackTrace();
}
}
public static String encrypt(String strToEncrypt, String secret) {
try {
setKey(secret);
Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE, secretKey);
```

```
Return
Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-
8")));
 } catch (Exception e) {
 System.out.println("Error while encrypting: " + e.toString());
 }
 return null;
 }
 public static String decrypt(String strToDecrypt, String secret) {
 try {
setKey(secret);
 Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
cipher.init(Cipher.DECRYPT_MODE, secretKey); return new
String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
 } catch (Exception e) {
 System.out.println("Error while decrypting: " + e.toString());
 }
 return null;
 }
 public static void main(String[] args) {
final String secretKey = "lbrce csection";
String originalString = "www.lbrce.edu";
 String encryptedString = AES.encrypt(originalString, secretKey);
 String decryptedString = AES.decrypt(encryptedString, secretKey);
System.out.println("URL Encryption Using AES Algorithm\n---------- ");
 System.out.println("Original URL : " + originalString);
 System.out.println("Encrypted URL : " + encryptedString);
 System.out.println("Decrypted URL : " + decryptedString);
 }
}
```

OUTPUT:

URL Encryption Using AES Algorithm

Original URL : www.lbrce.edu

Encrypted URL : 77eTRluGI5G3/VxvVlhc7A==

Decrypted URL : www.lbrce.edu

RESULT:

Thus the java program for AES has been implemented and the output verified successfully.

b) des.java

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
public class DES
{
public static void main(String[] argv) {
 try{
 System.out.println("Message Encryption Using DES Algorithm\n --------");
 KeyGenerator keygenerator = KeyGenerator.getInstance("DES");
 SecretKey myDesKey = keygenerator.generateKey();
Cipher desCipher;
desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
desCipher.init(Cipher.ENCRYPT_MODE, myDesKey);
 byte[] text = "Secret Information ".getBytes();
System.out.println("Message [Byte Format] : " + text);
System.out.println("Message : " + new String(text));
```

```
byte[] textEncrypted = desCipher.doFinal(text);
System.out.println("EncryptedMessage: "+textEncrypted);
desCipher.init(Cipher.DECRYPT_MODE,myDesKey);
byte[] textDecrypted = desCipher.doFinal(textEncrypted);
 System.out.println("Decrypted Message: " + new String(textDecrypted));
}catch(NoSuchAlgorithmException e){
e.printStackTrace();
}catch(NoSuchPaddingException e){
e.printStackTrace();
}catch(InvalidKeyException e){
e.printStackTrace();
}catch(IllegalBlockSizeException e){
e.printStackTrace();
}catch(BadPaddingException e){
e.printStackTrace();
}
}
}
```

OUTPUT

Message Encryption Using DES Algorithm

Message [Byte Format] : [B@604ed9f0
Message : Secret Information
Encrypted Message: [B@6a41eaa2
Decrypted Message: Secret Information

RESULT:

Thus the java program for DES has been implemented and the output verified successfully.

4. Implement any two Private -Key based algorithms

a) Diffie-Hellman Key Exchange algorithm

AIM:

To implement the Diffie-Hellman Key Exchange algorithm for a given problem

.

ALGORITHM:

1. Alice and Bob publicly agree to use a modulus p = 23 and base g = 5 (which is a primitive root modulo 23).
2. Alice chooses a secret integer a = 4, then sends Bob A = g^a mod p
   - A = 5^4 mod 23 = 4
3. Bob chooses a secret integer b = 3, then sends Alice B = g^b mod p
   - B = 5^3 mod 23 = 10
4. Alice computes s = B^a mod p
   - s = 10^4 mod 23 = 18
5. Bob computes s = A^b mod p
   - s = 4^3 mod 23 = 18
6. Alice and Bob now share a secret (the number 18).

PROGRAM:

```
DiffieHellman.java class DiffieHellman {
public static void main(String args[]) {
int p = 23; /* publicly known (prime number) */
 int g = 5; /* publicly known (primitive root) */
 int x = 4; /* only Alice knows this secret */
 int y = 3; /* only Bob knows this secret */
 double aliceSends = (Math.pow(g, x)) % p;
 double bobComputes = (Math.pow(aliceSends, y)) % p;
double bobSends = (Math.pow(g, y)) % p; double
aliceComputes = (Math.pow(bobSends, x)) % p; double
sharedSecret = (Math.pow(g, (x * y))) % p;
 System.out.println("simulation of Diffie-Hellman key exchange algorithm\n---------
--------------------------------");
```

```
 System.out.println("Alice Sends : " + aliceSends);
 System.out.println("Bob Computes : " + bobComputes);
 System.out.println("Bob Sends : " + bobSends);
 System.out.println("Alice Computes : " + aliceComputes);
 System.out.println("Shared Secret : " + sharedSecret);  /* shared secrets
should match and equality is transitive */
if ((aliceComputes == sharedSecret) && (aliceComputes ==
bobComputes)){
System.out.println("Success: Shared Secrets Matches! " + sharedSecret);
else {
 System.out.println("Error: Shared Secrets does not Match");
 }
}
```

OUTPUT:

```
simulation of Diffie-Hellman key exchange algorithm
---------------------------------------------------------
Alice Sends : 4.0
Bob Computes : 18.0
Bob Sends : 10.0
Alice Computes : 18.0
Shared Secret : 18.0
Success: Shared Secrets Matches! 18.0
```

RESULT:

Thus the Diffie-Hellman key exchange algorithm has been implemented using Java Program and the output has been verified successfully.

b) RSA Algorithm

AIM: To implement RSA (Rivest–Shamir–Adleman) algorithm

ALGORITHM:

1. Choose two prime number p and q
2. Compute the value of n and p
3. Find the value of e (public key)
4. Compute the value of d (private key) using gcd()

5. Do the encryption and decryption

a. Encryption is given as,

$c = t^e \bmod n$

b. Decryption is given as,

$t = c^d \bmod n$

PROGRAM:

rsa.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.math.*;
import java.util.Random;
import java.util.Scanner;
public class RSA {
static Scanner sc = new Scanner(System.in);
public static void main(String[] args) {
// TODO code application logic here
System.out.print("Enter a Prime number: ");
BigInteger p = sc.nextBigInteger(); // Here's one prime number..
System.out.print("Enter another prime number:
"); BigInteger q = sc.nextBigInteger(); // ..and another.
BigInteger n = p.multiply(q);
BigInteger n2 = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
BigInteger e = generateE(n2);
BigInteger d = e.modInverse(n2); // Here's the multiplicative inverse
System.out.println("Encryption keys are: " + e + ", " + n);
System.out.println("Decryption keys are: " + d + ", " + n);
}
public static BigIntegergenerateE(BigIntegerfiofn) {
int y, intGCD; BigInteger e;
BigInteger gcd;
Random x = new Random();
do {
```

```
y = x.nextInt(fiofn.intValue()-1);
String z = Integer.toString(y); e
= new BigInteger(z); gcd =
fiofn.gcd(e); intGCD =
gcd.intValue();
}
while(y <= 2 || intGCD != 1); return
e;
}
}
```

OUTPUT:

Enter a Prime number: 79

Enter another prime number: 83

Encryption keys are: 3163, 6557

Decryption keys are: 2467, 6557

RESULT:

Thus the RSA has been implemented and the output has been verified successfully.

5. Explore any four network diagnosis tools

WHOIS

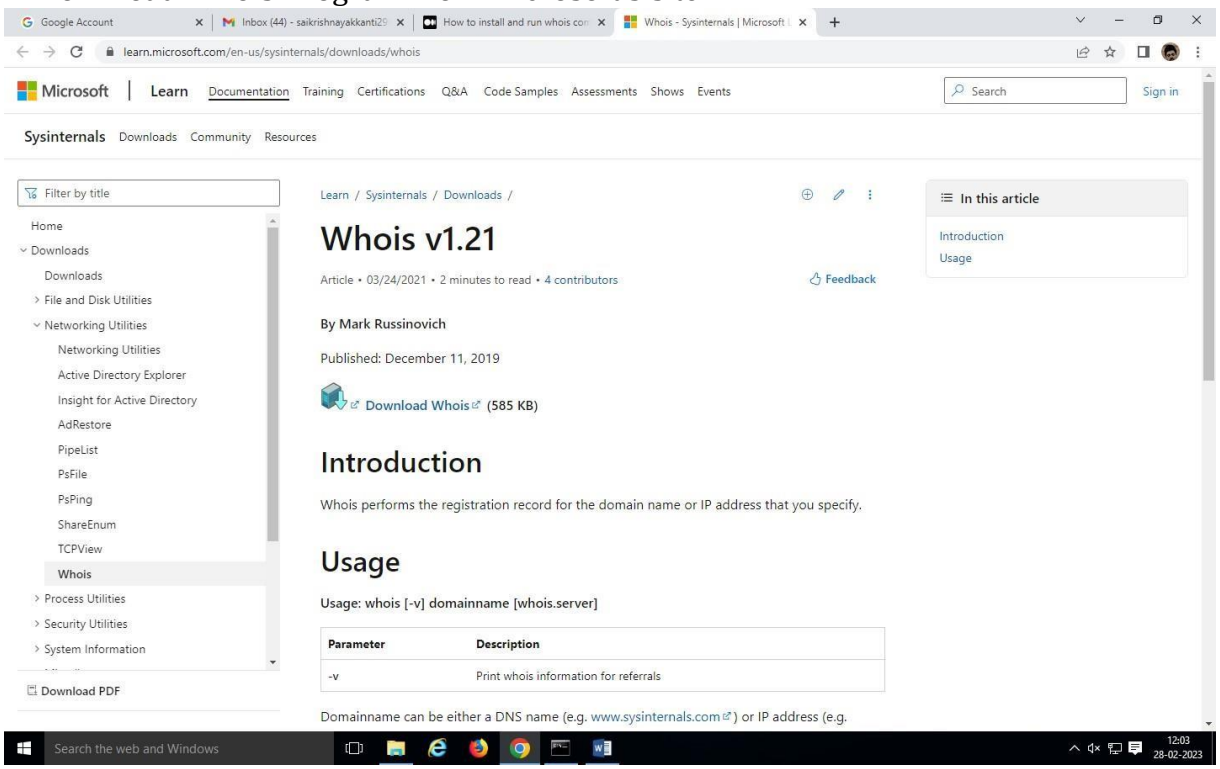
A Whois domain lookup allows you to trace the ownership and tenure of a domain name.

Similar to how all houses are registered with a governing authority, all domain name registries maintain a record of information about every domain name purchased through them, along with who owns it, and the date till which it has been purchased.

How to install and run whois command in Windows 10

Steps

1. Download Whois Program from Microsoft's site.



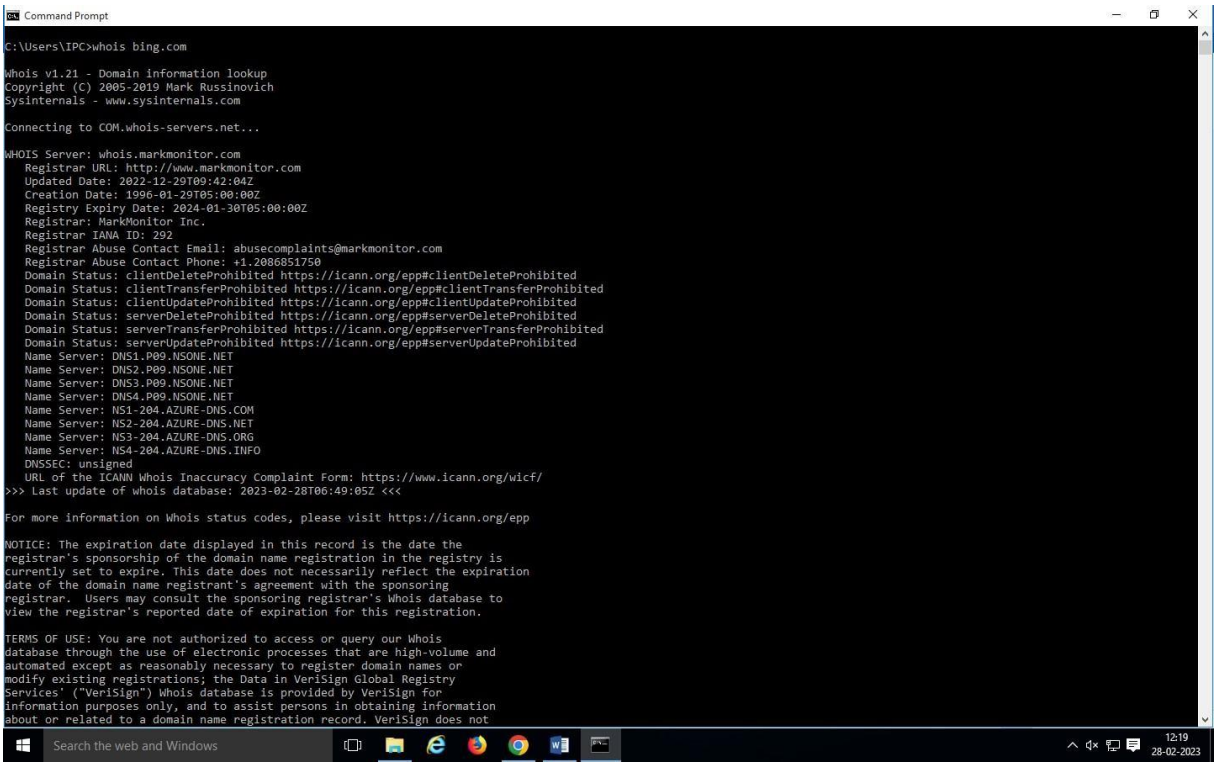
2. Now copy the zip file and paste in the following folder C drive windows system32
3. Extract the contents from the zip file

Now we will get 3 more files

Move to command prompt

Enter the command : whois domain.com

Example : whois bing.com



Dig

The dig (domain information groper) command is **a flexible tool for interrogating DNS servers** **name**

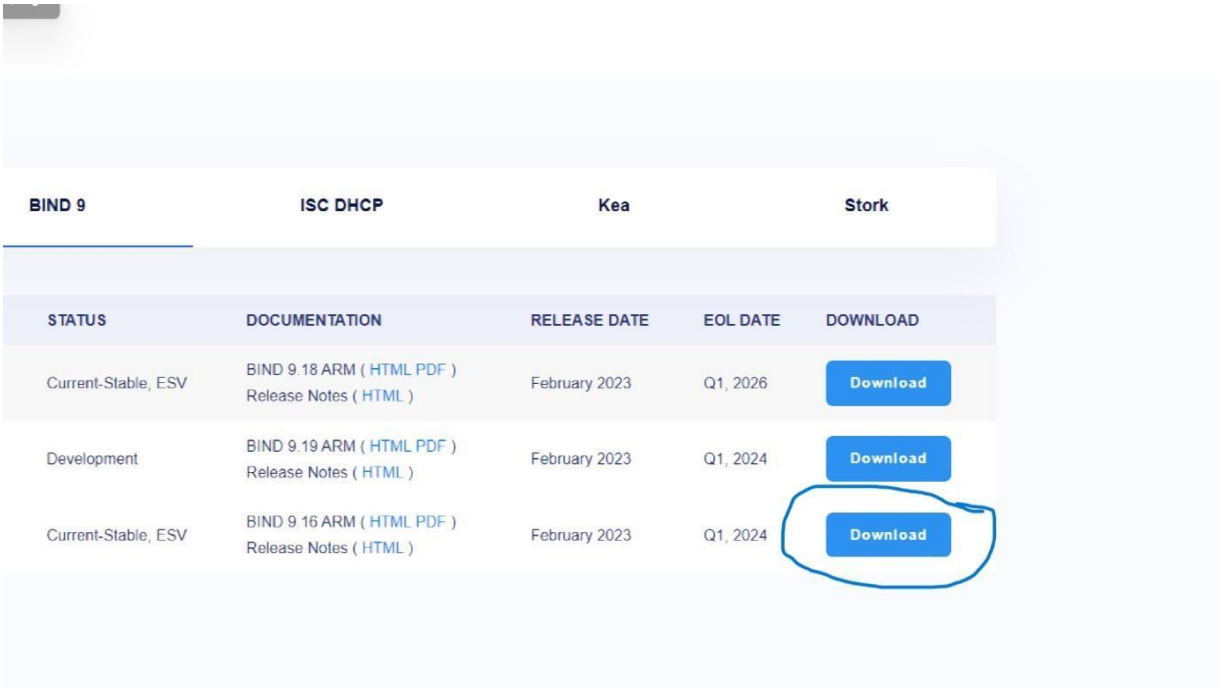
How to install the Dig on Windows.

1. Download BIND.
2. Install BIND.
3. Create Windows Path Variable.
4. Using dig command in windows. Using dig without command options.
5. List all records for a hostname.
6. Get a hostname IP address.
7. Check DNS Resolution.
8. Query a nameserver directly.
9. Do a reverse dns lookup.

Download BIND

Steps :-

1. Visit the BIND download page using your preferred web browser. https://www.isc.org/download/
2. Click the **Download** button to select the latest stable version of BIND. In this tutorial, we are using version 9.16.23.



3. Click the link to download the BIND installation zip file.

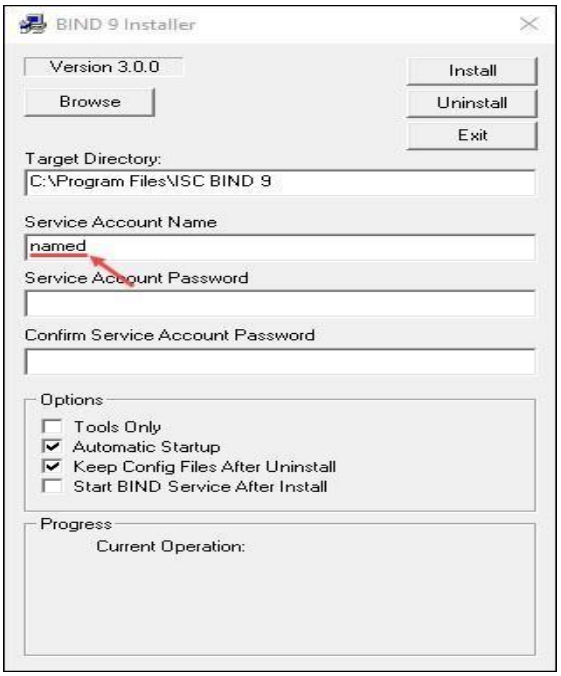
Now copy the zip file and paste in the following folder

D drive----------> MyFolder.

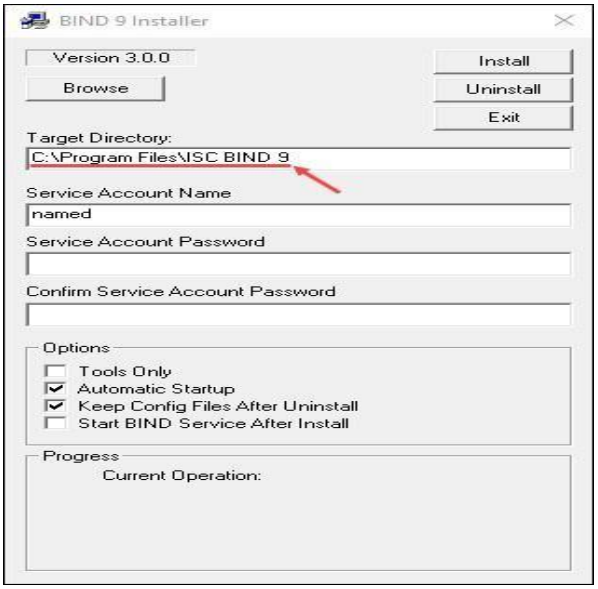
Install BIND

1. Extract the BIND installation file.

2. Open the BINDInstall.exe file as an administrator to start the installation wizard.

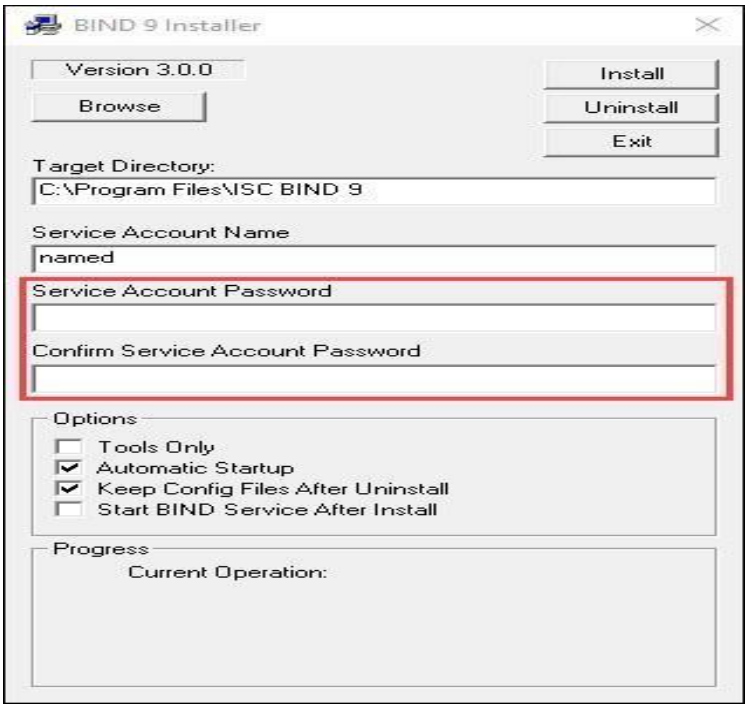


3. In the Target Directory field, set the path to where you want to install BIND on your system.

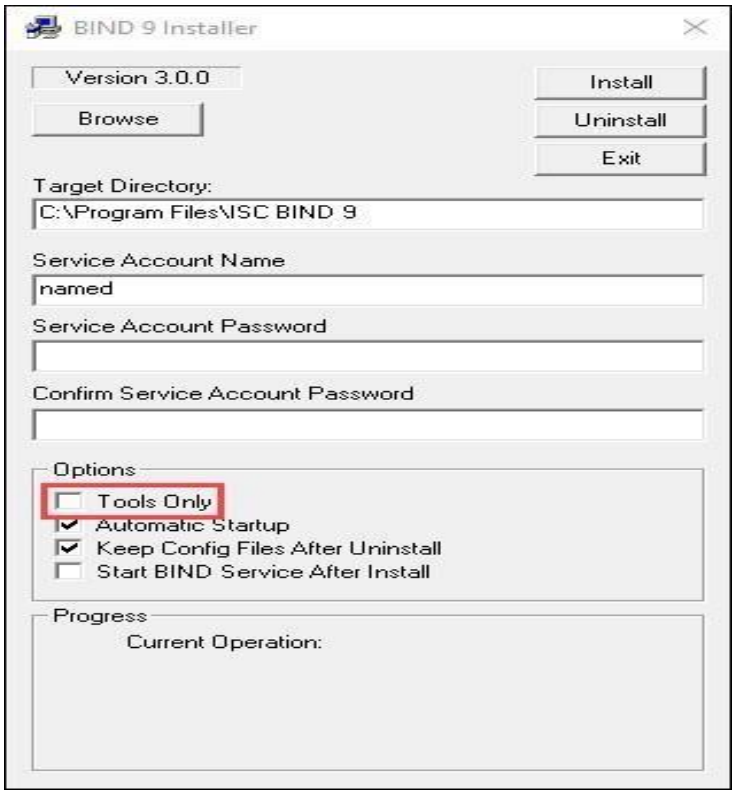


4. Set a name for your service account in the Service Account Name field.

5. Set and confirm a password for the service account.



6. In the Options section, check the Tools Only box.



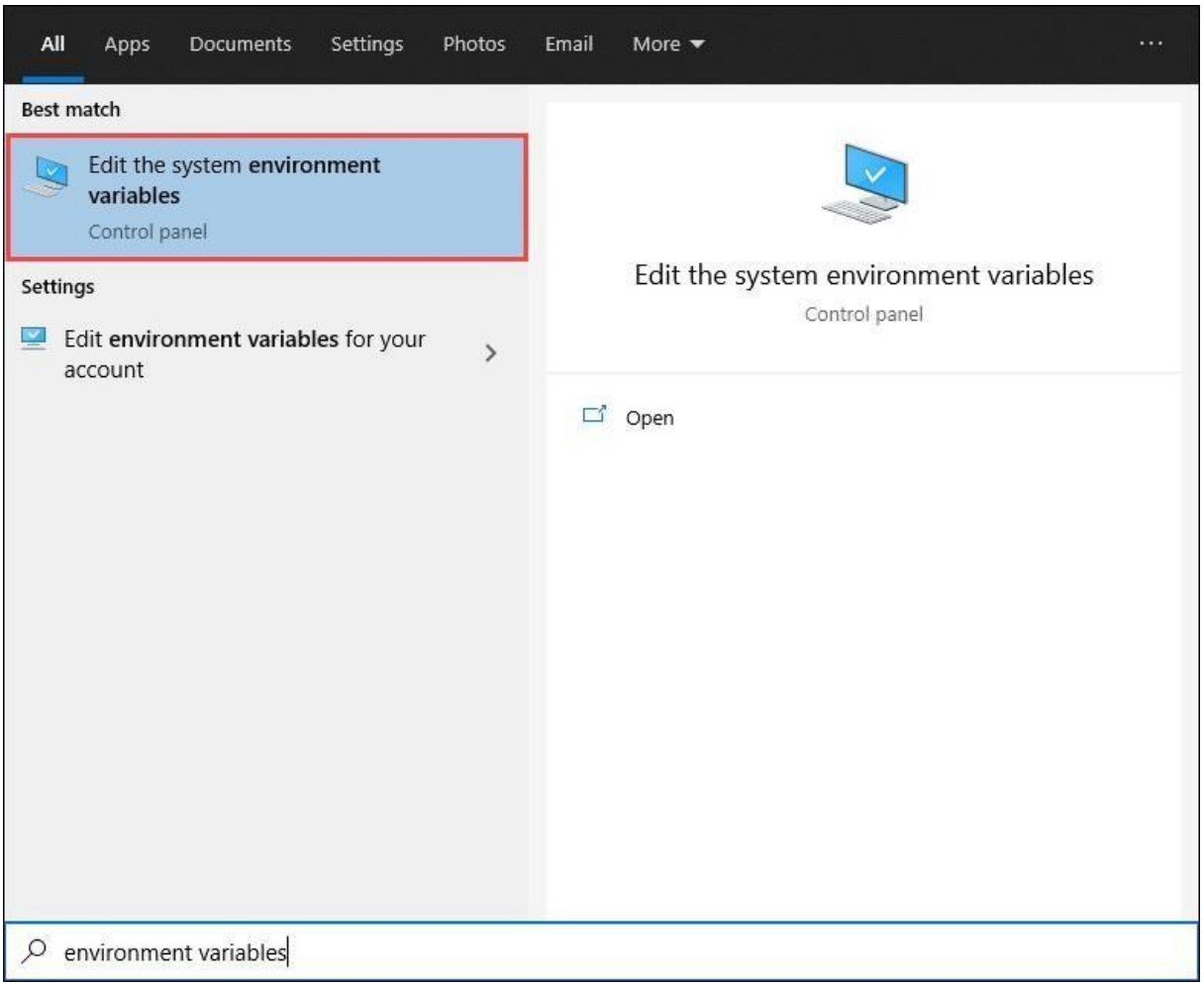
7. Click the Install button to start the installation process.

Note: Installing BIND may automatically install Microsoft Visual C++ Redistributable.
In this case, click the Install button to confirm.

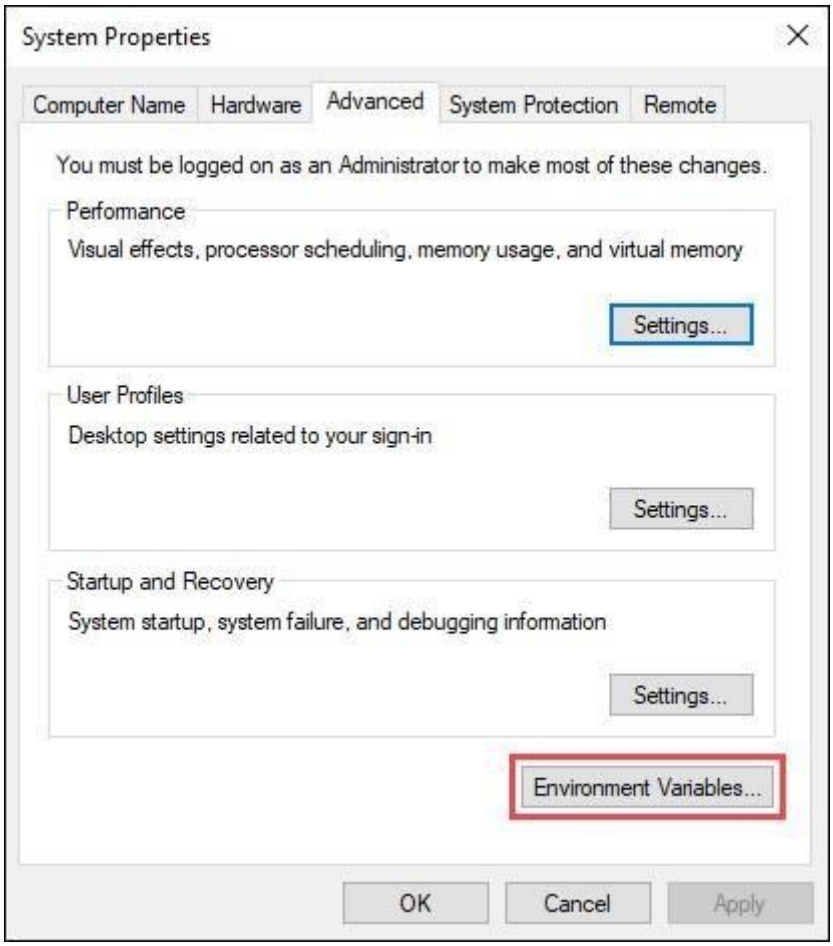
8. Once the installation is complete, click OK to confirm and Exit to close the installer.

Create Windows Path Variable

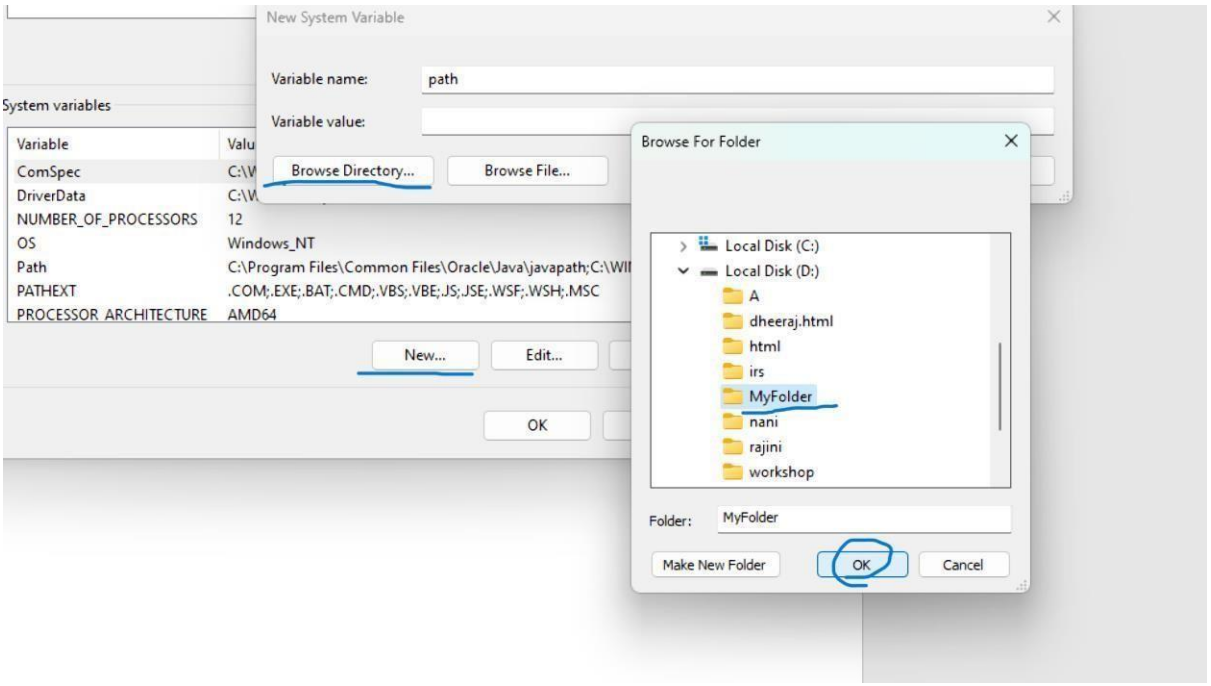
1. Open the Start menu and search for "environment variables".
2. Select the Edit the system environment variables option.



2. In the System Properties window, under the Advanced tab, click the Environment Variables… button.



3. Under System variables, perform the following order :-
Move to new - - - ->set variable name as “path” and ---------- >Browse Directory - - - -> select your destination folder - - - -
-> ok.



4. Click OK to confirm the edits to the Path variable.
5. Click OK to confirm the changes and exit the Environment Variables window.

Using dig Command in Windows

After installing BIND, open the Windows command prompt to start using the dig command. The dig command uses the following syntax:

dig [hostname] [options]

Using dig Without Command Options

Using the dig command without any options returns DNS data on the provided hostname.

For instance:

```
C:\Users\akova>dig google.com

; <<>> DiG 9.16.23 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14807
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             39      IN      A       142.250.180.238

;; Query time: 16 msec
;; SERVER: 10.240.30.10#53(10.240.30.10)
;; WHEN: Tue Dec 07 10:54:31 Central Europe Standard Time 2021
;; MSG SIZE  rcvd: 55


C:\Users\akova>
```

The dig command also allows you to specify the type of record you want to query by using:

dig [hostname] [record type]

List All Records for a Hostname

To return all records for the provided hostname, use the any option:

dig [hostname] any

Get a Hostname's IP Address

Using the +short option with the dig command provides a shortened output (usually just the IP address):

dig [hostname] +short

For example: dig google.com +short

```
C:\Users\akova>dig google.com +short
142.250.217.142

C:\Users\akova>
```

Check DNS Resolution

Adding the +trace option resolves the query starting from the root nameserver and working its way down, reporting the results from each step:

dig [hostname] +trace For instance:

dig google.com +trace

Query a Nameserver Directly

The dig command also allows you to query a nameserver directly:
dig @[nameserver address] [hostname]
Do a Reverse DNS Lookup
Another use for the dig command is performing reverse DNS lookups: **dig -X [IP address]**

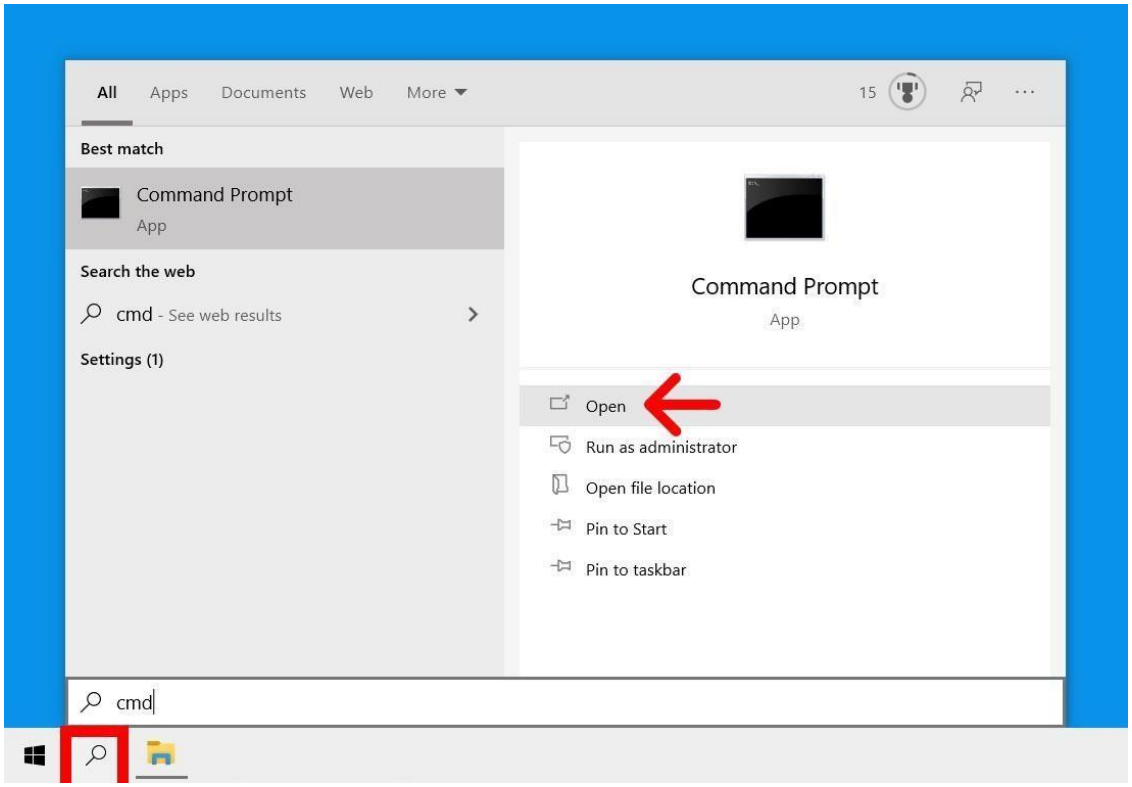
TRACEROUTE:

- Traceroute is a network diagnostic tool that tracks the path of a packet of data as it travels from your computer to a destination over the internet.
- Traceroute prints the route that packets take to a network host.
- Traceroute utility uses the TTL field in the IP header to achieve its operation.

How to Run a Traceroute on a Windows 10 Computer

To run a traceroute on a Windows 10 computer, open the Windows search box and type CMD into the search bar. Then open the Command Prompt app and type in tracert followed by a space and then the destination URL or IP address. Finally, hit Enter.

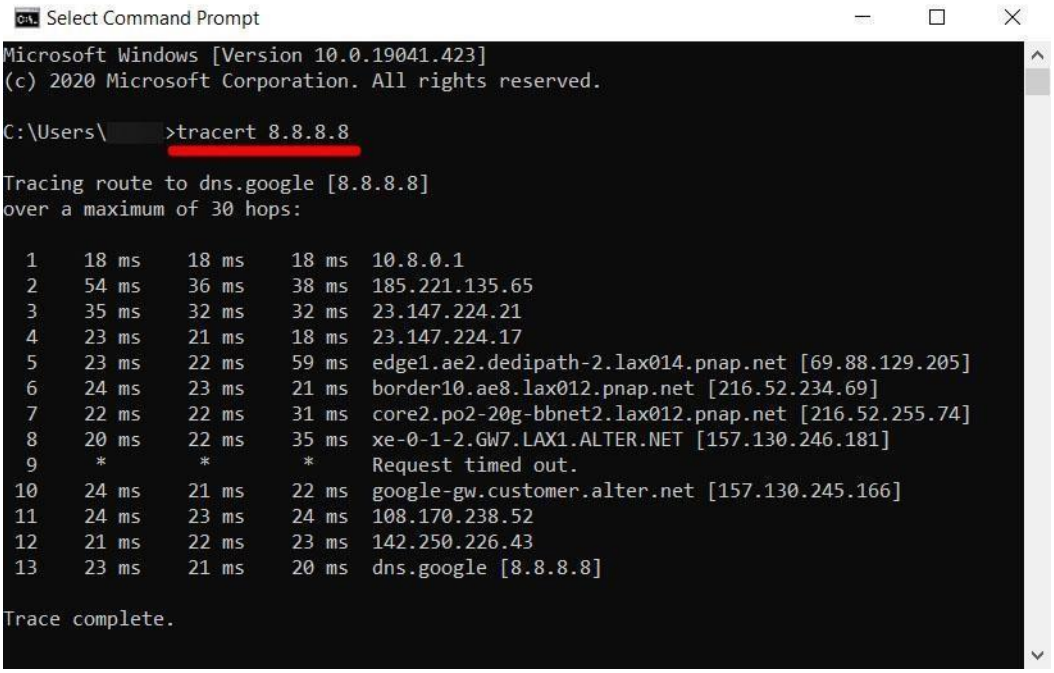
1. Open the Windows search box.
2. Then type CMD in the search bar and click Open.



3. Next, type ***tracert*** followed by a space and then an IP address or URL.

If you just want to test your internet connection, it is a good idea to run a traceroute to 8.8.8.8

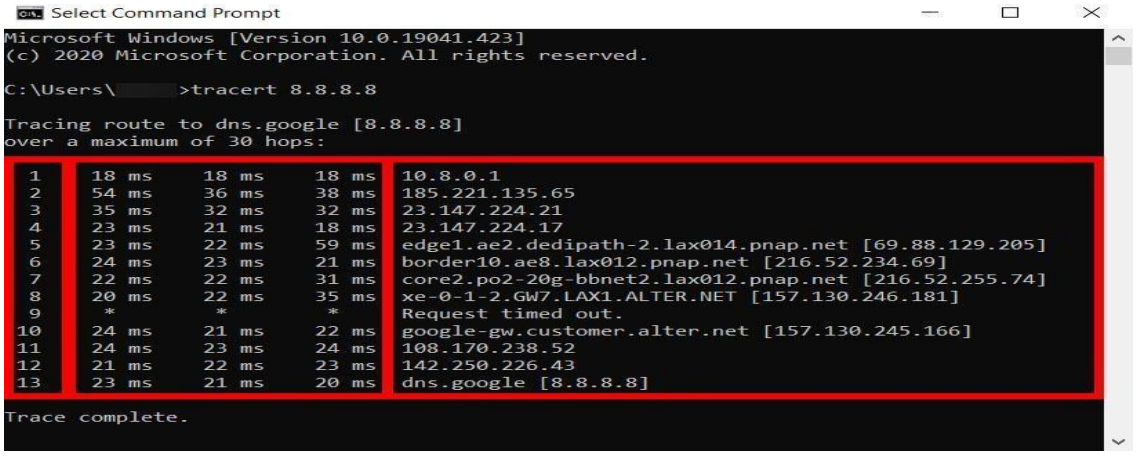
4. Finally, press Enter on your keyboard and wait for the traceroute to finish.



How to Read the Traceroute Columns:

Column 1: This represents the hop number, or the number of hops that the three data packets were pushed through to reach the destination. Columns 2-4: These show the round trip time measured in milliseconds. RTT represents the time it took for a data packet to travel from the source to the destination and back again. To check for the consistency of the response times, the traceroute command sends three packets to each hop, which is why there are three time values listed per row. RTT values below 100 milliseconds are acceptable. However, if you see RTT values consistently increasing from the middle hop to the destination, it could be due to a network problem.

Column 5: This column shows the name or IP address of the routers on every hop from your computer to the destination. It will also list the domain name of the router, if that information is available.



Traceroute Command Variations:

The following example of command syntax shows all of the possible options:

tracert-d -h maximum_hops -j host-list -w timeout target_host

What the parameters do:

-d
Specifies to not resolve addresses to host names

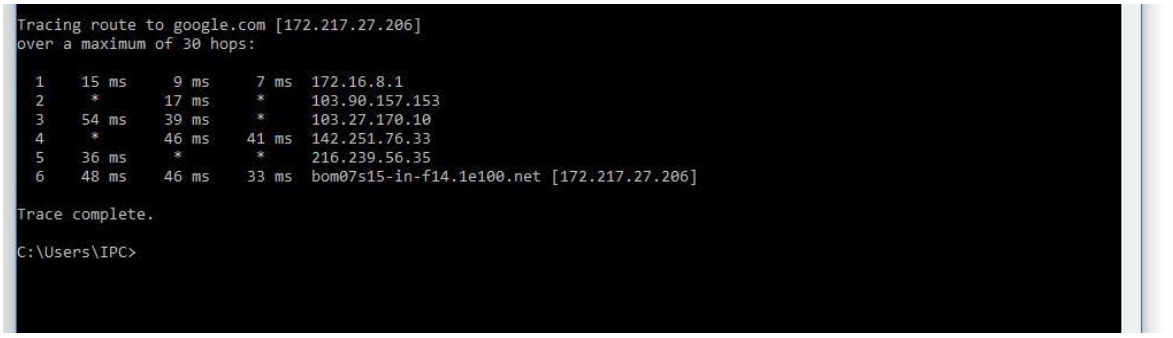
-h maximum_hops
Specifies the maximum number of hops to search for the target

-j host-list
Specifies loose source route along the host-list

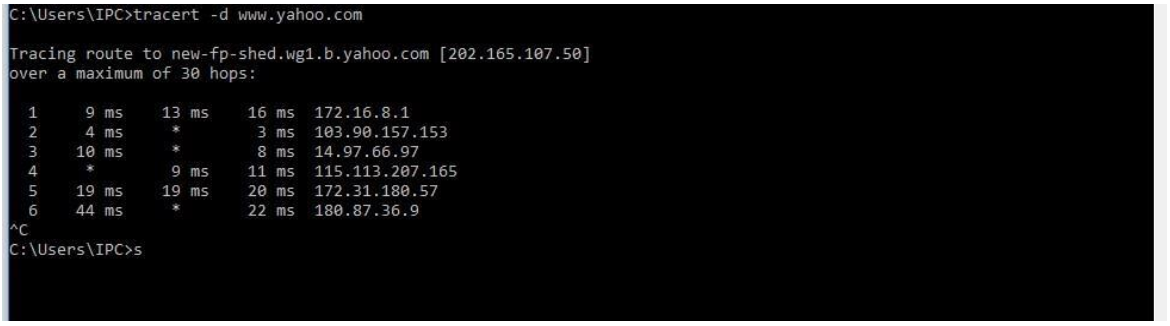
-w timeout
Waits the number of milliseconds specified by timeout for each reply target_host Specifies the name or IP address of the target host

commands:

1. tracert google.com



2. tracert –d *www.yahoo.com*



3. tracert –h 3 lifewire.com

```
Tracing route to lifewire.com [151.101.2.137]
over a maximum of 3 hops:

  1    12 ms    18 ms     *     172.16.8.1
  2     3 ms     4 ms     2 ms  103.90.157.153
  3     *        8 ms    12 ms  static-97.66.97.14-tataidc.co.in [14.97.66.97]

Trace complete.

C:\Users\IPC>
```

4. tracert –d twitter.com

```
Tracing route to twitter.com [104.244.42.129]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  172.16.8.1
  2     3 ms     *       21 ms  103.90.157.153
  3   100 ms     *       16 ms  14.97.66.97
  4     9 ms     8 ms     9 ms  115.113.207.165
^C
C:\Users\IPC>
```

Nslookup Tool:

Nslookup (stands for "Name Server Lookup") is a useful command for getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNSrelated problems.

Syntax:

nslookup [option]

Options of nslookup command:

- nslookup google.com:

nslookup followed by the domain name will display the "A Record" (IP Address) of the domain. Use this command to find the address record for a domain. It queries to domain name servers and gets the details.

```
C:\Users\IPC>nslookup google.com
Server:  dns.google
Address:  8.8.8.8

DNS request timed out.
    timeout was 2 seconds.
Name:    google.com
Address:  2404:6800:4009:810::200e
```

- **nslookup 8.8.8.8**: Reverse DNS lookup

It can do the reverse DNS look-up by providing the IP Address as an argument to nslookup

```
C:\Users\IPC>nslookup 8.8.8.8
Server:  dns.google
Address:  8.8.8.8

Name:    dns.google
Address:  8.8.8.8
```

- **nslookup -type=soa google.com**: Lookup for an soa record SOA record (start of authority), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, etc…

```
C:\Users\RAJESH>nslookup -type=soa google.com
Server:  UnKnown
Address:  192.168.0.52

Non-authoritative answer:
google.com
        primary name server = ns1.google.com
        responsible mail addr = dns-admin.google.com
        serial  = 512579957
        refresh = 900 (15 mins)
        retry   = 900 (15 mins)
        expire  = 1800 (30 mins)
        default TTL = 60 (1 min)

google.com      nameserver = ns3.google.com
google.com      nameserver = ns2.google.com
google.com      nameserver = ns4.google.com
google.com      nameserver = ns1.google.com
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  AAAA IPv6 address = 2001:4860:4802:34::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  AAAA IPv6 address = 2001:4860:4802:38::a
ns3.google.com  internet address = 216.239.36.10
ns3.google.com  AAAA IPv6 address = 2001:4860:4802:36::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  AAAA IPv6 address = 2001:4860:4802:32::a
```

- **nslookup -type=ns google.com**: Lookup for an ns record

NS (Name Server) record maps a domain name to a list of DNS servers authoritative for that domain. It will output the name serves which are associated with the given domain.

```
C:\Users\RAJESH>nslookup -type=ns google.com
Server:  UnKnown
Address:  192.168.0.52

Non-authoritative answer:
google.com      nameserver = ns3.google.com
google.com      nameserver = ns1.google.com
google.com      nameserver = ns2.google.com
google.com      nameserver = ns4.google.com
```

- **nslookup -debug google.com:**

To view the information for debugging.

```
C:\Users\RAJESH>nslookup -debug google.com
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 1, rcode = NXDOMAIN
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 0,  authority records = 1,  additional = 0

    QUESTIONS:
        52.0.168.192.in-addr.arpa, type = PTR, class = IN
    AUTHORITY RECORDS:
    ->  168.192.in-addr.arpa
        ttl = 3600 (1 hour)
        primary name server = prisoner.iana.org
        responsible mail addr = hostmaster.root-servers.org
        serial  = 1
        refresh = 604800 (7 days)
        retry   = 60 (1 min)
        expire  = 604800 (7 days)
        default TTL = 604800 (7 days)
```

6. Study about Wireshark packet sniffer tool in promiscuous and nonpromiscuous mode

1. Aim: Study of packet sniffer tool wireshark,
2. Objectives: To observe the performance in promiscuous & non promiscuous mode & to find the packets based on different filters.
3. Outcomes: The learner will be able to:-

- Identify different packets moving in/out of network using packet sniffer for network analysis.
- Understand professional, ethical, legal, security and social issues and responsibilities. Also will be able to analyze the local and global impact of computing on individuals, organizations, and society.
- Match the industry requirements in the domains of Database management, Programming and Networking with the required management skills.

4. Hardware / Software Required: Wireshark, Ethereal and tcpdump.

5. Theory:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, colorcoding and other features that let you dig deep into network traffic and inspect individual packets.

Applications:

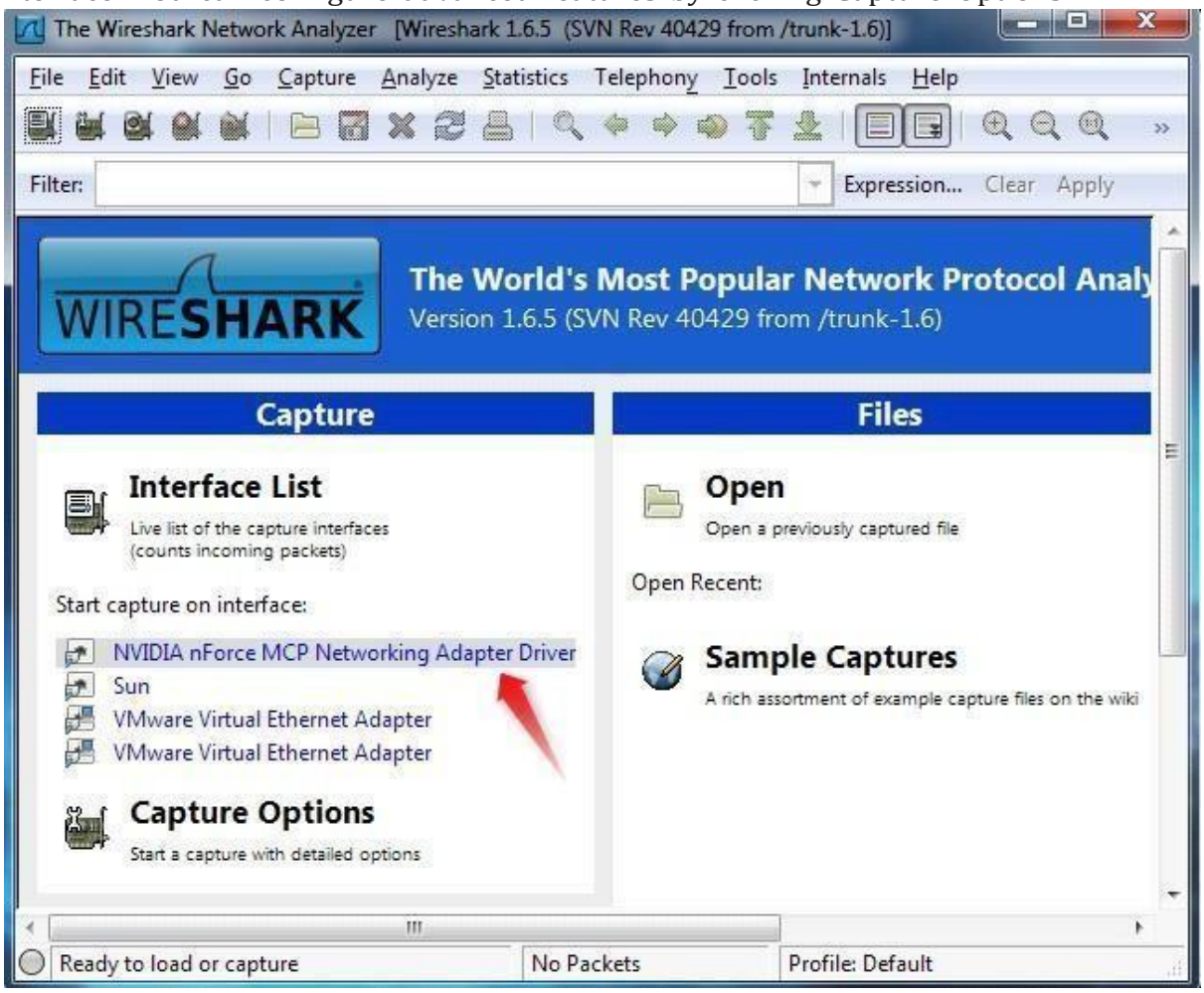
- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals beside these examples can be helpful in many other situations too.

Features:

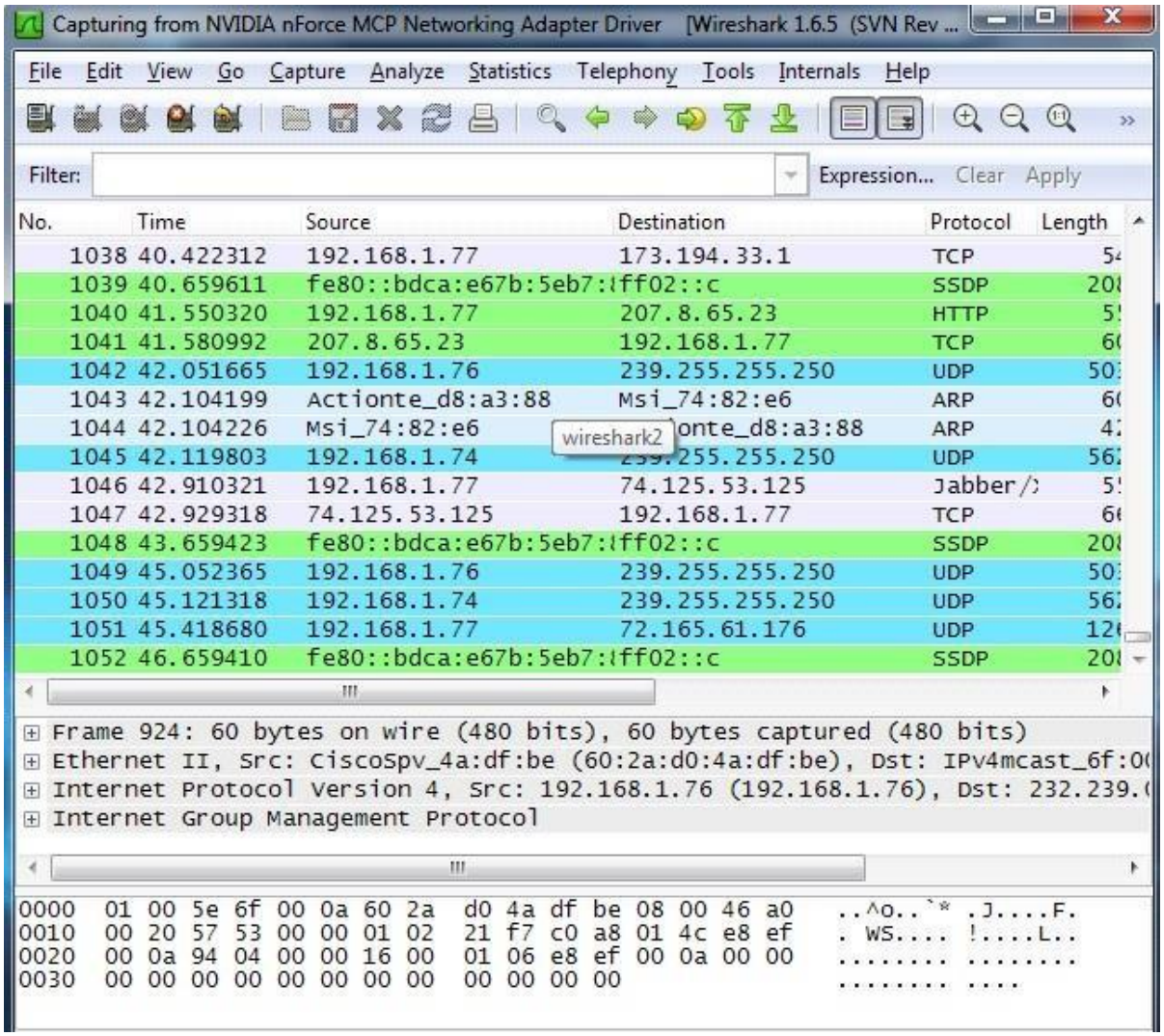
- The following are some of the many features wireshark provides:
- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data. • Display packets with very detailed protocol information. • Export some or all packets in a number of capture file formats.
- Filter packets on many criteria. • Search for packets on many criteria. • Colorize packet display based on filters.
- Create various statistics.

Capturing Packets

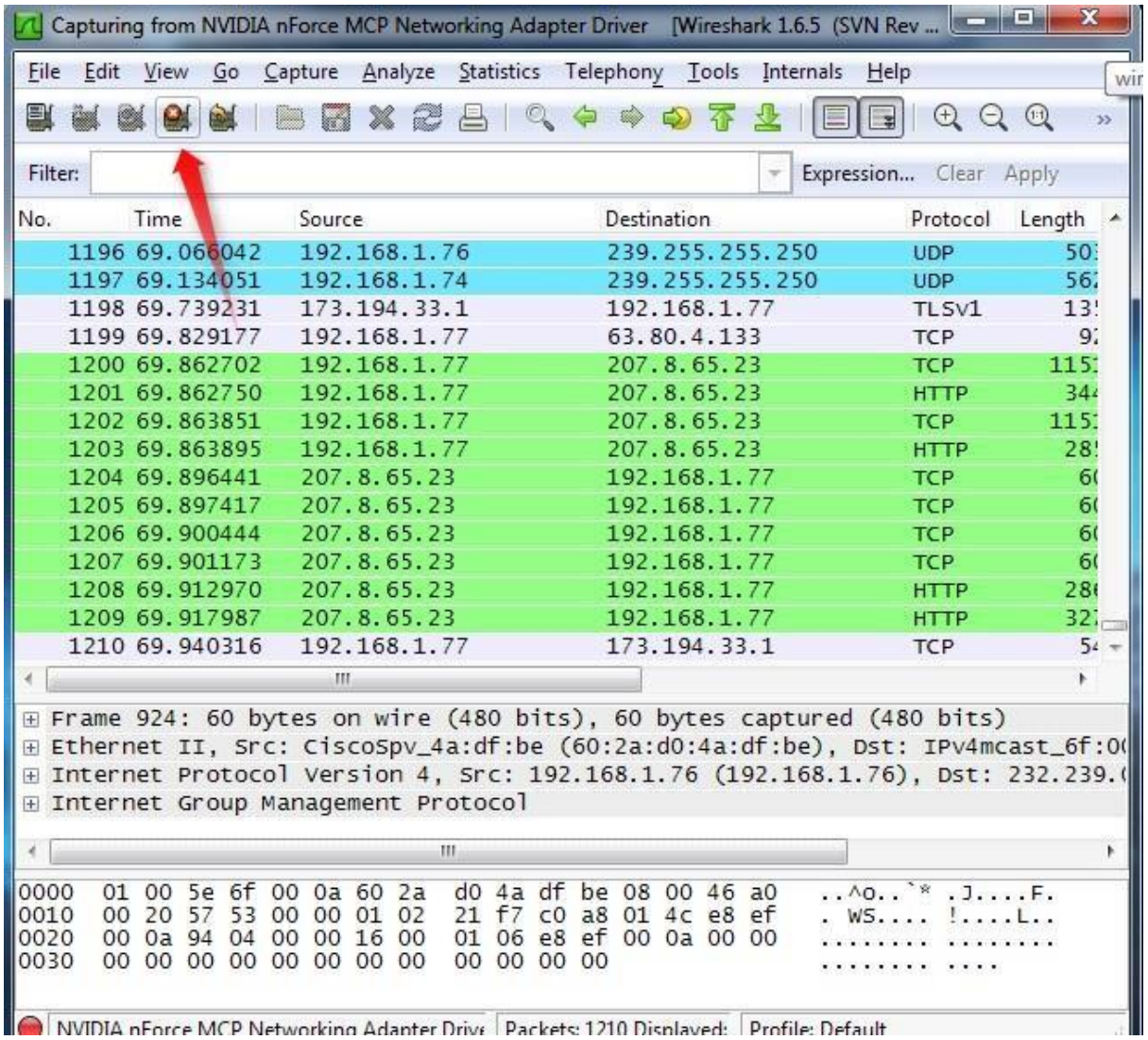
After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.



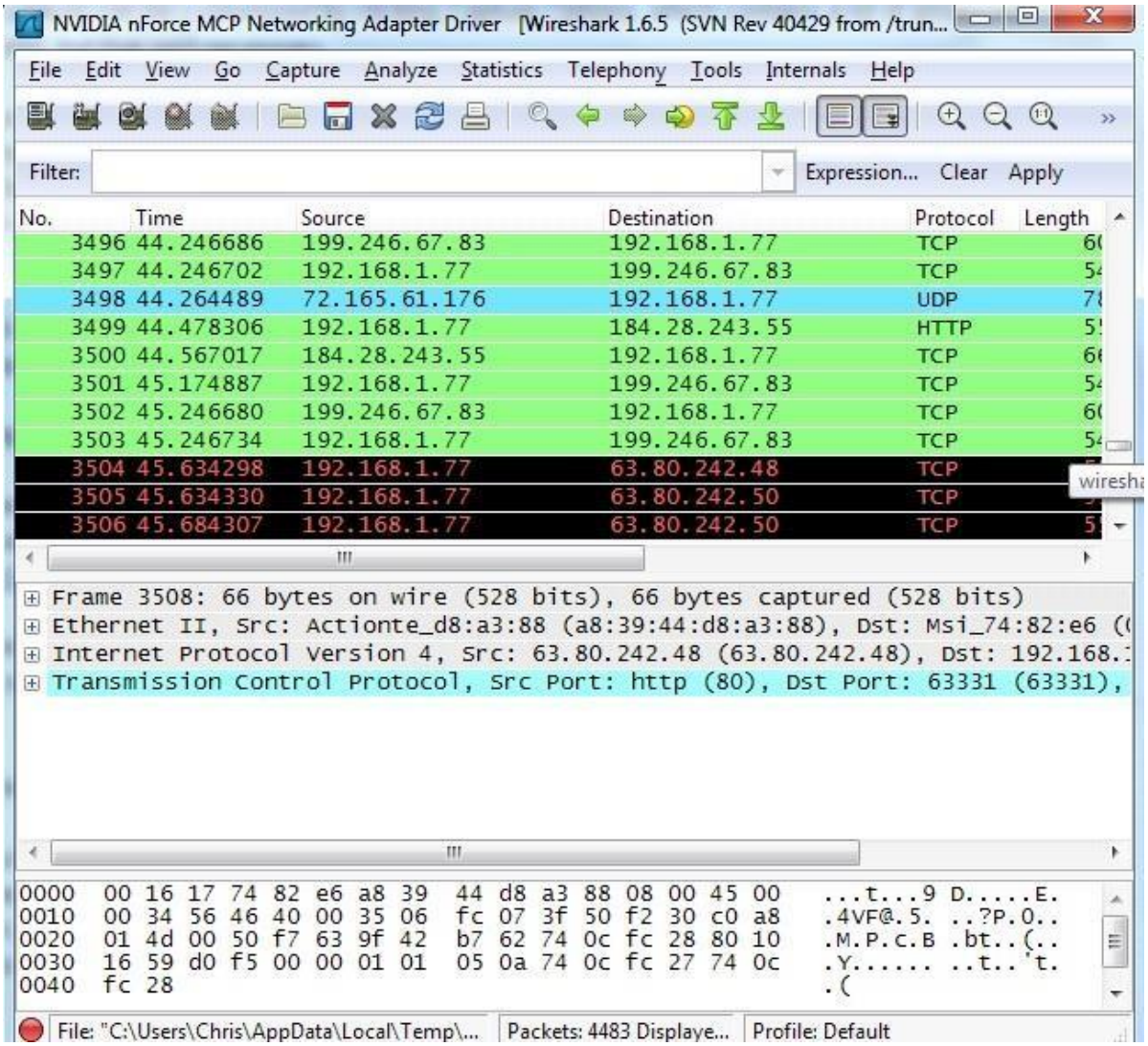
As soon as you click the interface‘s name, you‘ll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you‘re capturing on a wireless interface and have promiscuous mode enabled in your capture options, you‘ll also see other the other packets on the network.



Click the stop capture button near the top left corner of the window when you want to stop capturing traffic.

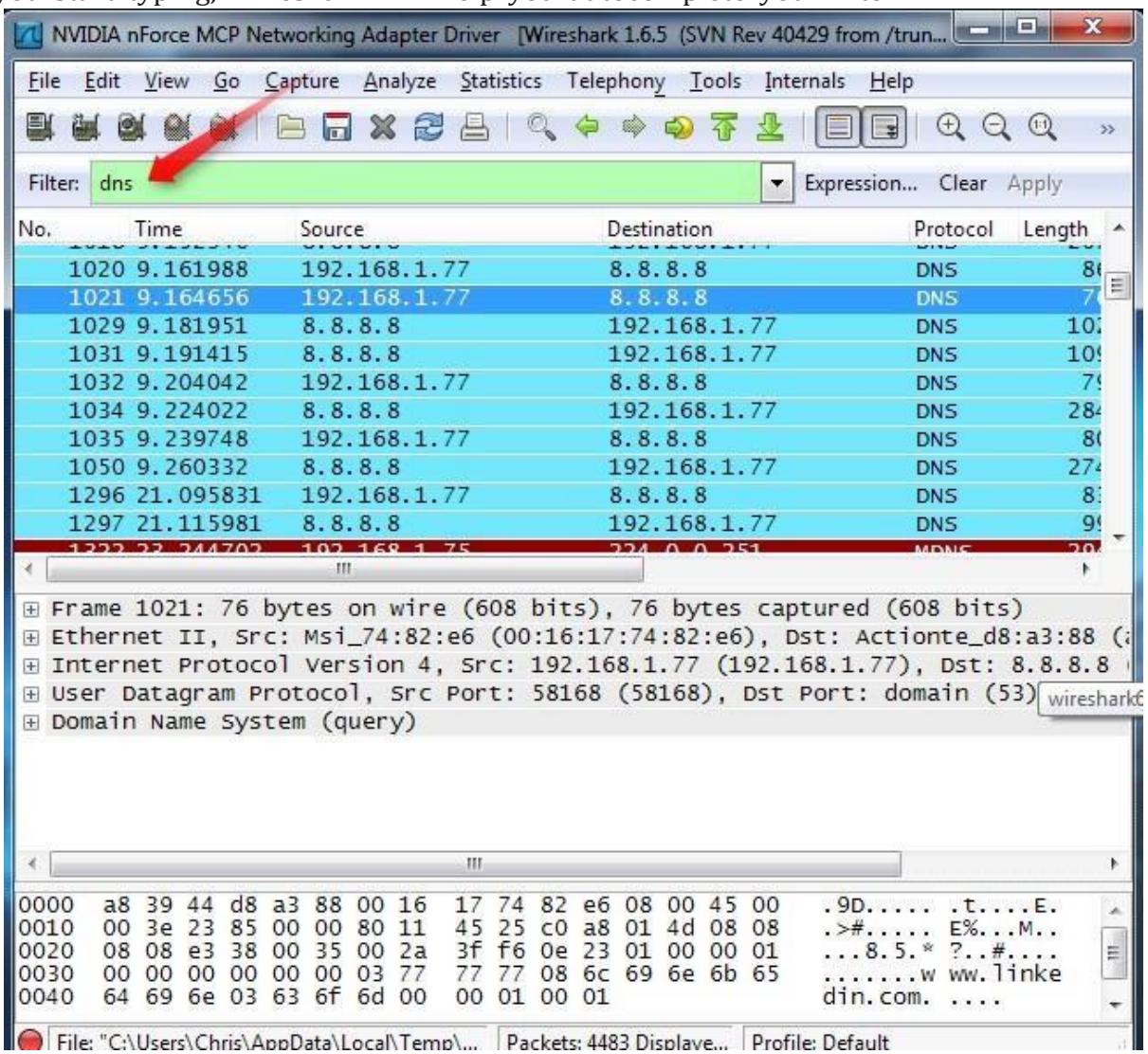


Wireshark uses colors to help you identify the types of traffic at a glance. By default,green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

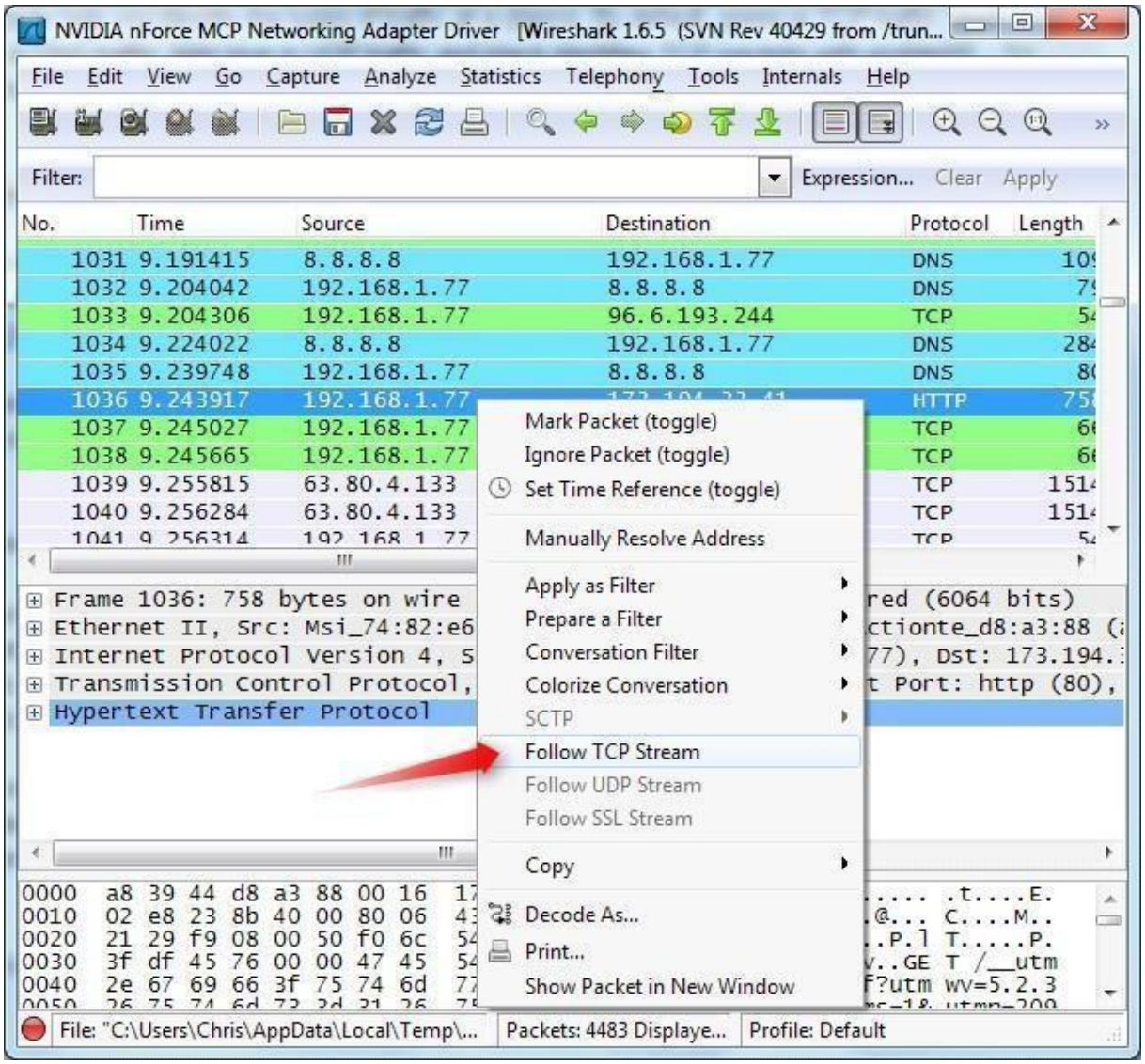

NVIDIA nForce MCP Networking Adapter Driver [Wireshark 1.6.5 (SVN Rev 40429 from /trun...
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Length
3496 44.246686 199.246.67.83 192.168.1.77 TCP
3497 44.246702 192.168.1.77 199.246.67.83 TCP
3498 44.264489 72.165.61.176 192.168.1.77 UDP
3499 44.478306 192.168.1.77 184.28.243.55 HTTP
3500 44.567017 184.28.243.55 192.168.1.77 TCP
3501 45.174887 192.168.1.77 199.246.67.83 TCP
3502 45.246680 199.246.67.83 192.168.1.77 TCP
3503 45.246734 192.168.1.77 199.246.67.83 TCP
3504 45.634298 192.168.1.77 63.80.242.48 TCP
3505 45.634330 192.168.1.77 63.80.242.50 TCP
3506 45.684307 192.168.1.77 63.80.242.50 TCP
Frame 3508: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: Actionte_d8:a3:88 (a8:39:44:d8:a3:88), Dst: Msi_74:82:e6
Internet Protocol Version 4, Src: 63.80.242.48 (63.80.242.48), Dst: 192.168.
Transmission Control Protocol, Src Port: http (80), Dst Port: 63331 (63331),
0000 00 16 17 74 82 e6 a8 39 44 d8 a3 88 08 00 45 00 ...t...9 D.....E.
0010 00 34 56 46 40 00 35 06 fc 07 3f 50 f2 30 c0 a8 .4VF@.5. ..?P.0..
0020 01 4d 00 50 f7 63 9f 42 b7 62 74 0c fc 28 80 10 .M.P.c.B .bt..(..
0030 16 59 d0 f5 00 00 01 01 05 0a 74 0c fc 27 74 0c .Y..... ..t..'t.
0040 fc 28 .(
File: "C:\Users\Chris\AppData\Local\Temp\... Packets: 4483 Displaye... Profile: Default

Filtering Packets

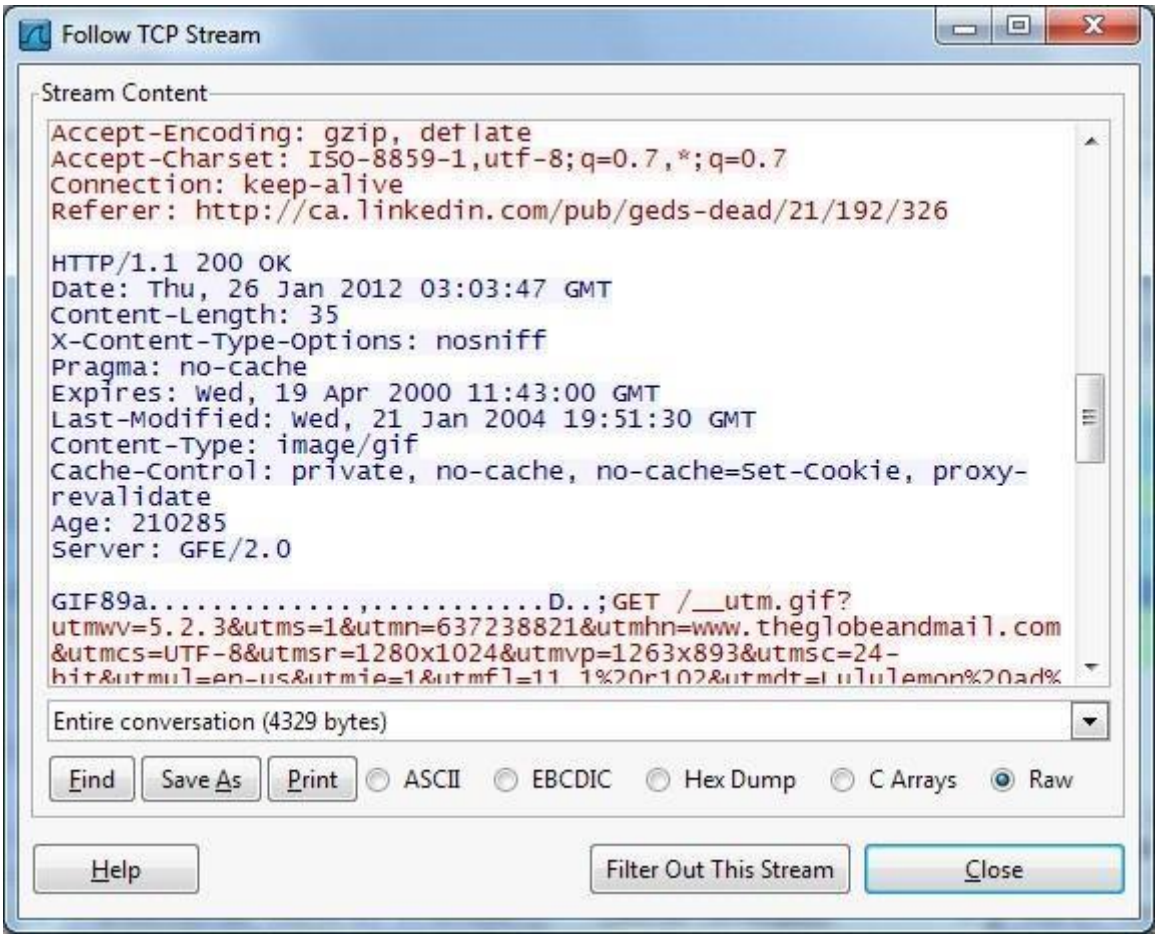
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in. The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type —dns‖ and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



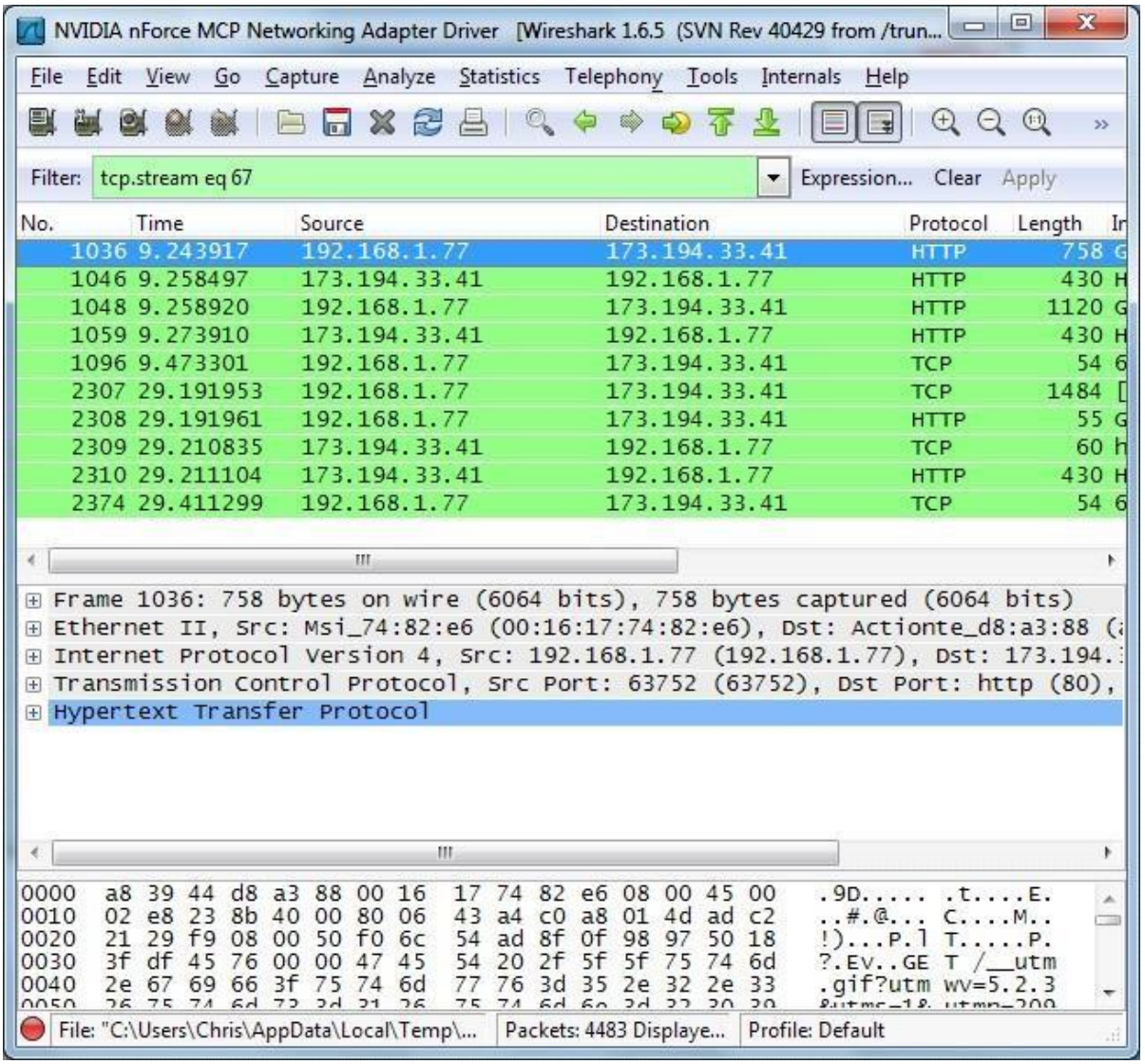
Another interesting thing you can do is right-click a packet and select Follow TCPStream



You'll see the full conversation between the client and the server.

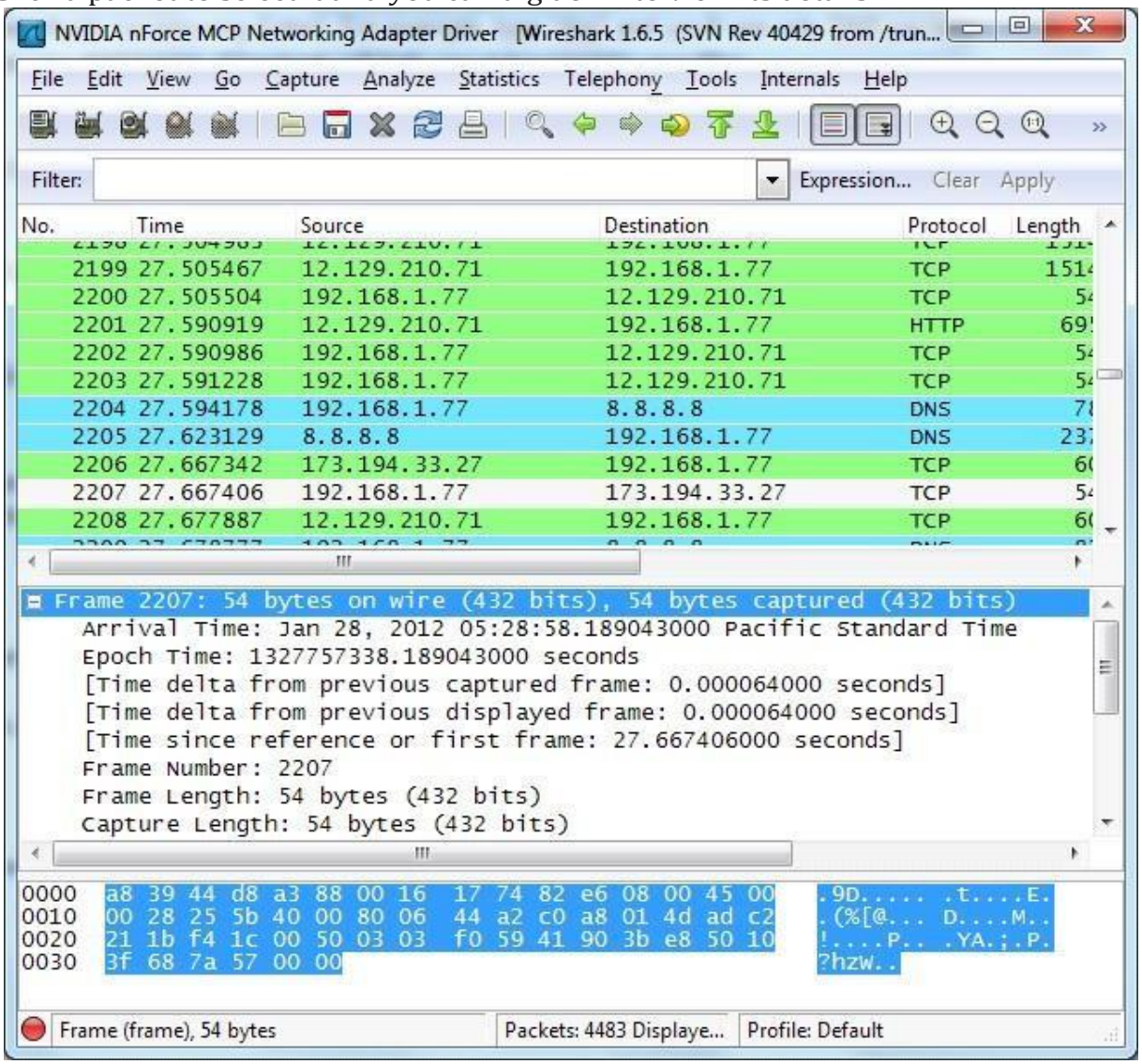


Close the window and you'll find a filter has been applied automatically — Wireshark is showing you the packets that make up the conversation.

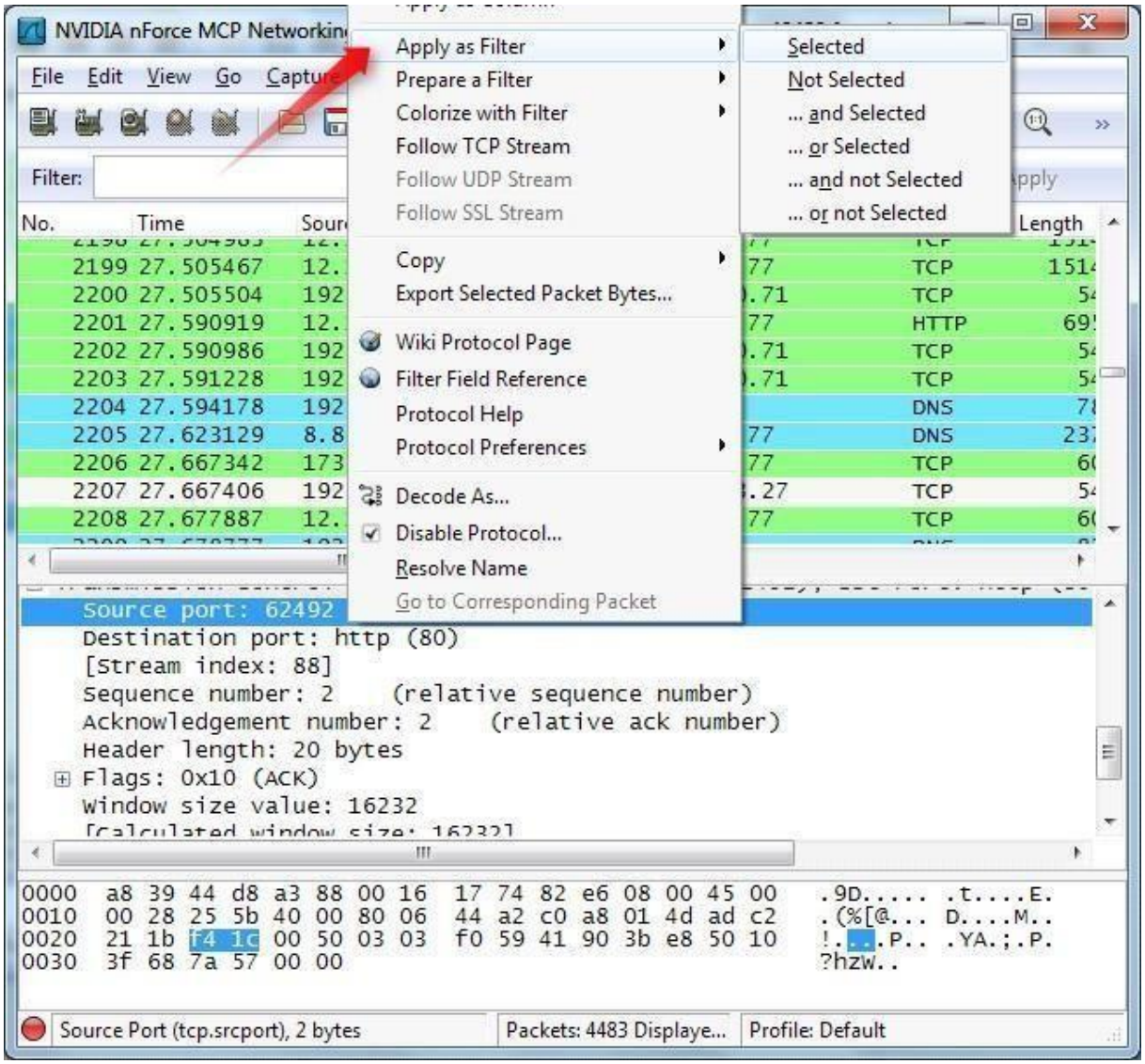

NVIDIA nForce MCP Networking Adapter Driver [Wireshark 1.6.5 (SVN Rev 40429 from /trun...
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: tcp.stream eq 67
Expression... Clear Apply
No. Time Source Destination Protocol Length
1036 9.243917 192.168.1.77 173.194.33.41 HTTP 758
1046 9.258497 173.194.33.41 192.168.1.77 HTTP 430
1048 9.258920 192.168.1.77 173.194.33.41 HTTP 1120
1059 9.273910 173.194.33.41 192.168.1.77 HTTP 430
1096 9.473301 192.168.1.77 173.194.33.41 TCP 54
2307 29.191953 192.168.1.77 173.194.33.41 TCP 1484
2308 29.191961 192.168.1.77 173.194.33.41 HTTP 55
2309 29.210835 173.194.33.41 192.168.1.77 TCP 60
2310 29.211104 173.194.33.41 192.168.1.77 HTTP 430
2374 29.411299 192.168.1.77 173.194.33.41 TCP 54
Frame 1036: 758 bytes on wire (6064 bits), 758 bytes captured (6064 bits)
Ethernet II, Src: Msi_74:82:e6 (00:16:17:74:82:e6), Dst: Actionte_d8:a3:88
Internet Protocol Version 4, Src: 192.168.1.77 (192.168.1.77), Dst: 173.194.
Transmission Control Protocol, Src Port: 63752 (63752), Dst Port: http (80),
Hypertext Transfer Protocol
0000 a8 39 44 d8 a3 88 00 16 17 74 82 e6 08 00 45 00 .9D..... .t....E.
0010 02 e8 23 8b 40 00 80 06 43 a4 c0 a8 01 4d ad c2 ..#.@... C....M..
0020 21 29 f9 08 00 50 f0 6c 54 ad 8f 0f 98 97 50 18 !)...P.l T.....P.
0030 3f df 45 76 00 00 47 45 54 20 2f 5f 5f 75 74 6d ?.Ev..GE T /__utm
0040 2e 67 69 66 3f 75 74 6d 77 76 3d 35 2e 32 2e 33 .gif?utm wv=5.2.3
File: "C:\Users\Chris\AppData\Local\Temp\...
Packets: 4483 Displaye...
Profile: Default

Inspecting Packets

Click a packet to select it and you can dig down to view its details.



You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



Wireshark is an extremely powerful tool, and this tutorial is just scratching the surface of what you can do with it. Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

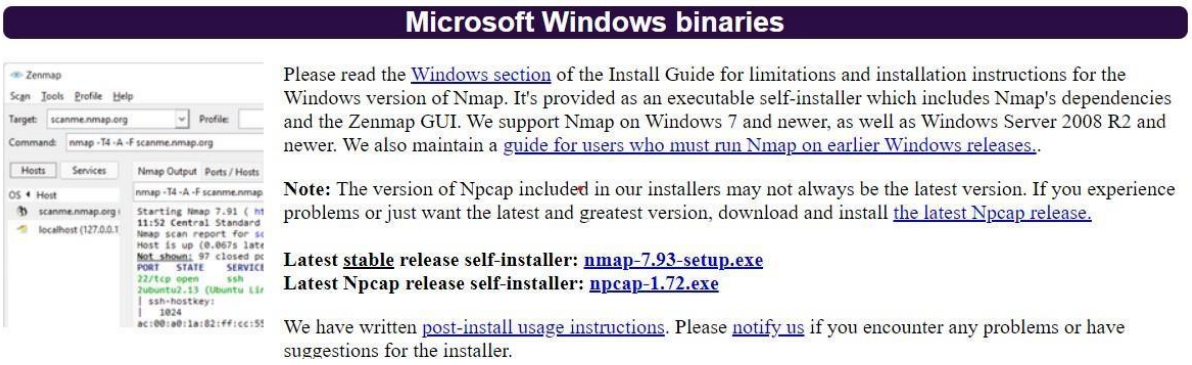
Conclusion:

In this experiment we analyze wireshark packet sniffing tool that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is network monitoring tool. It is opted for network monitoring, traffic analysis, troubleshooting, Packet grapping, message, protocol analysis, penetration testing and many other purposes.

6. Download and install nmap. Use it with different options to scan open ports,perform OS fingerprinting, do a ping scan, tcp port scan, udp port scan.

Installing nmap :

Step 1: Visit the official website using the URL https://nmap.org/download.html on any web browser the click on nmap-7.93-setup.exe.



Microsoft Windows binaries

Please read the Windows section of the Install Guide for limitations and installation instructions for the Windows version of Nmap. It's provided as an executable self-installer which includes Nmap's dependencies and the Zenmap GUI. We support Nmap on Windows 7 and newer, as well as Windows Server 2008 R2 and newer. We also maintain a guide for users who must run Nmap on earlier Windows releases..

Note: The version of Npcap included in our installers may not always be the latest version. If you experience problems or just want the latest and greatest version, download and install the latest Npcap release.

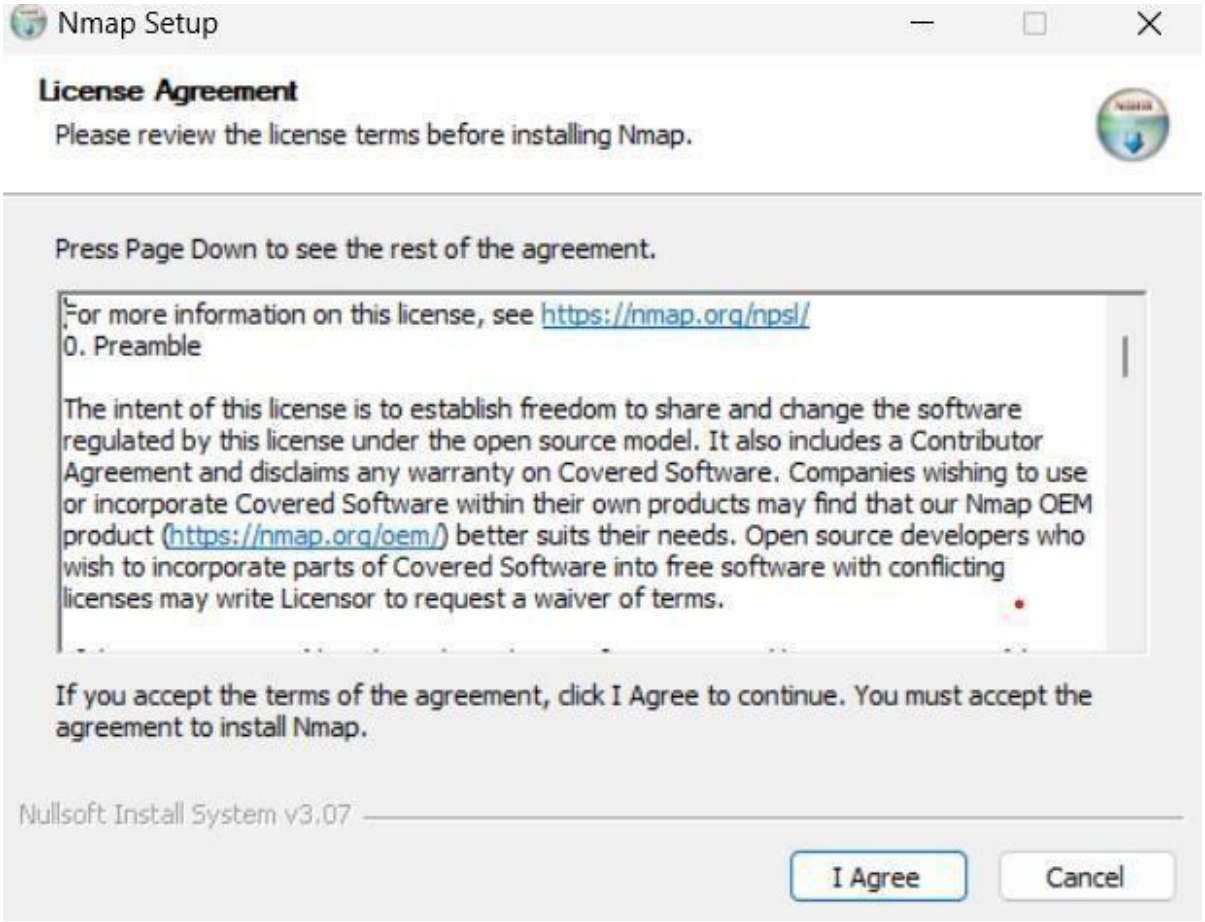
Latest stable release self-installer: nmap-7.93-setup.exe
Latest Npcap release self-installer: npcap-1.72.exe

We have written post-install usage instructions. Please notify us if you encounter any problems or have suggestions for the installer.

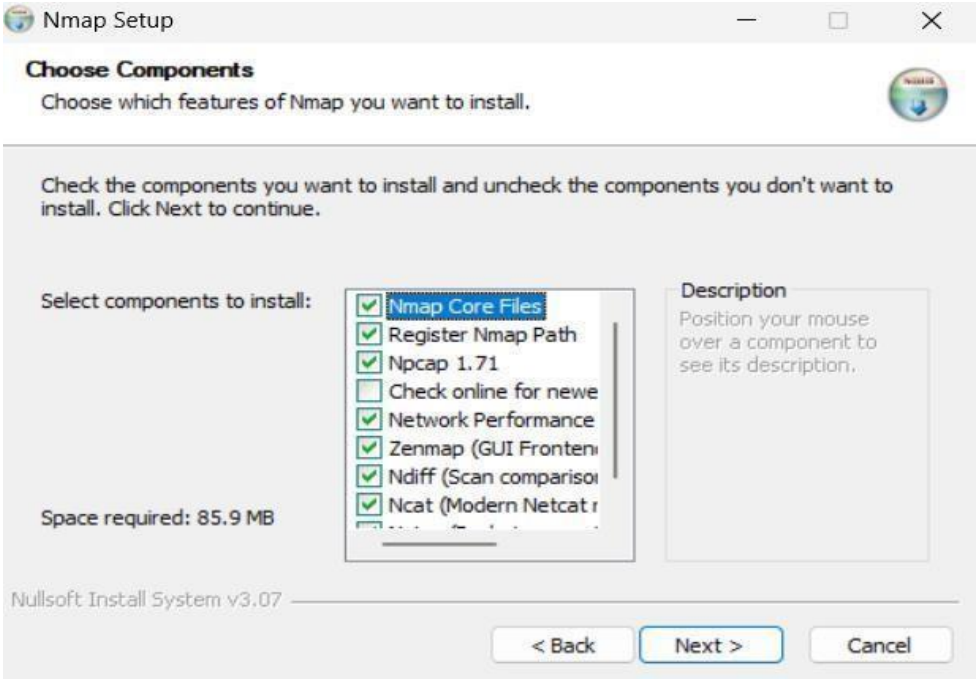
Step 2: Now check for the executable file in downloads in your system and run it.

Step 3: It will prompt confirmation to make changes to your system.
Click on Yes.

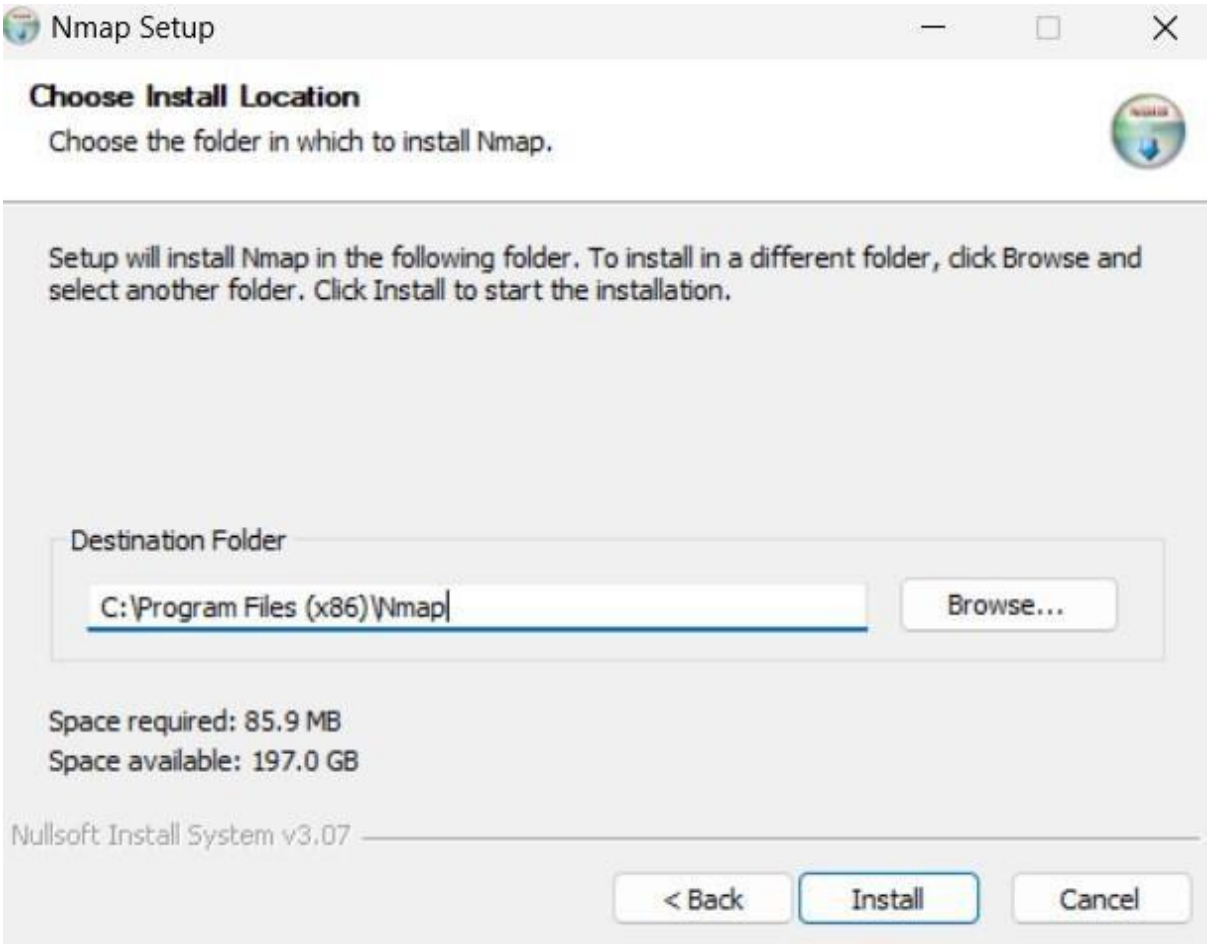
Step 4: The next screen will be of License Agreement , Click on I Agree.



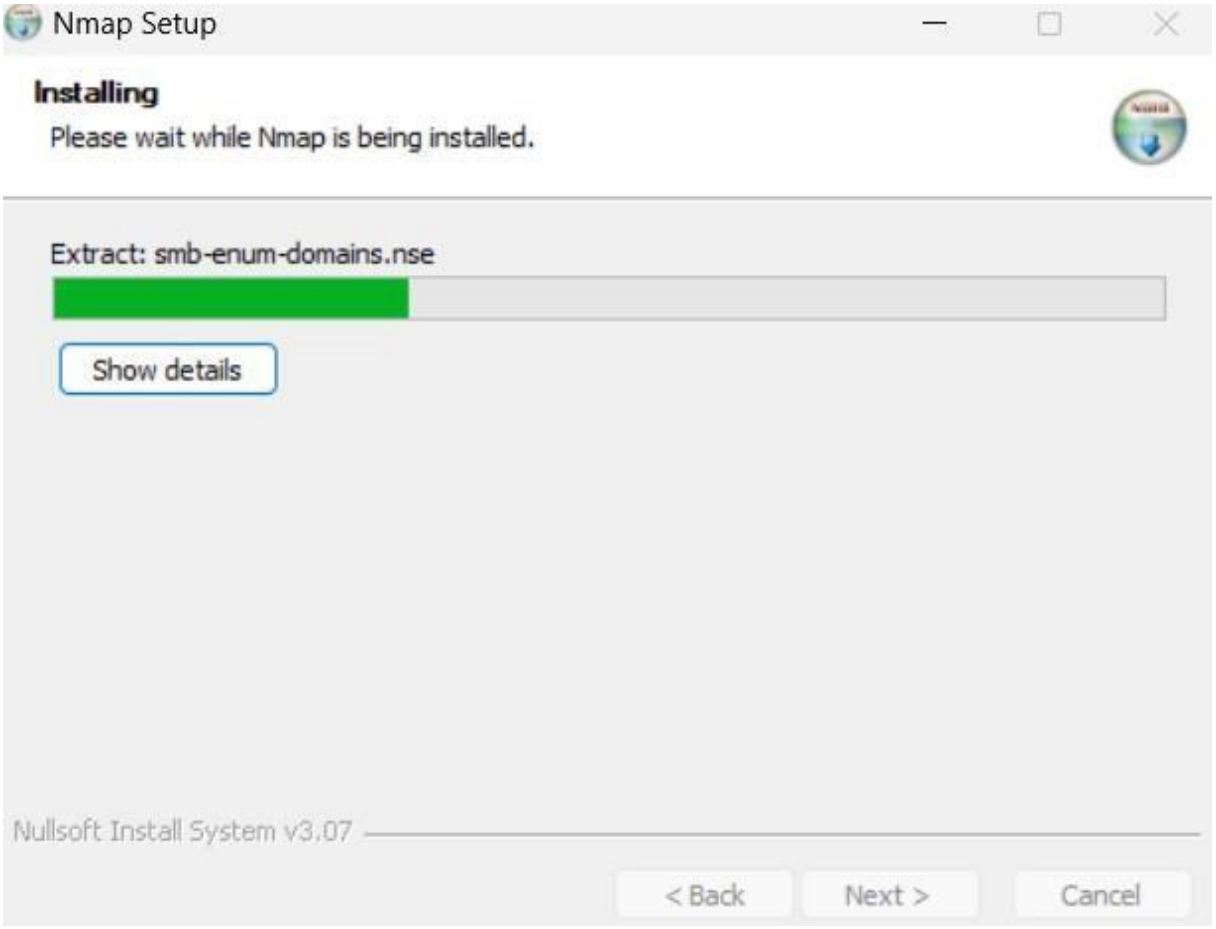
Step 5: Next Screen is of choosing components, all components are already marked so don't change anything just click on the Next button.



Step 6: In this step,we choose the installation location of Nmap.By default, it uses the C drive but you can change it into another drive that will have sufficient memory space for installation

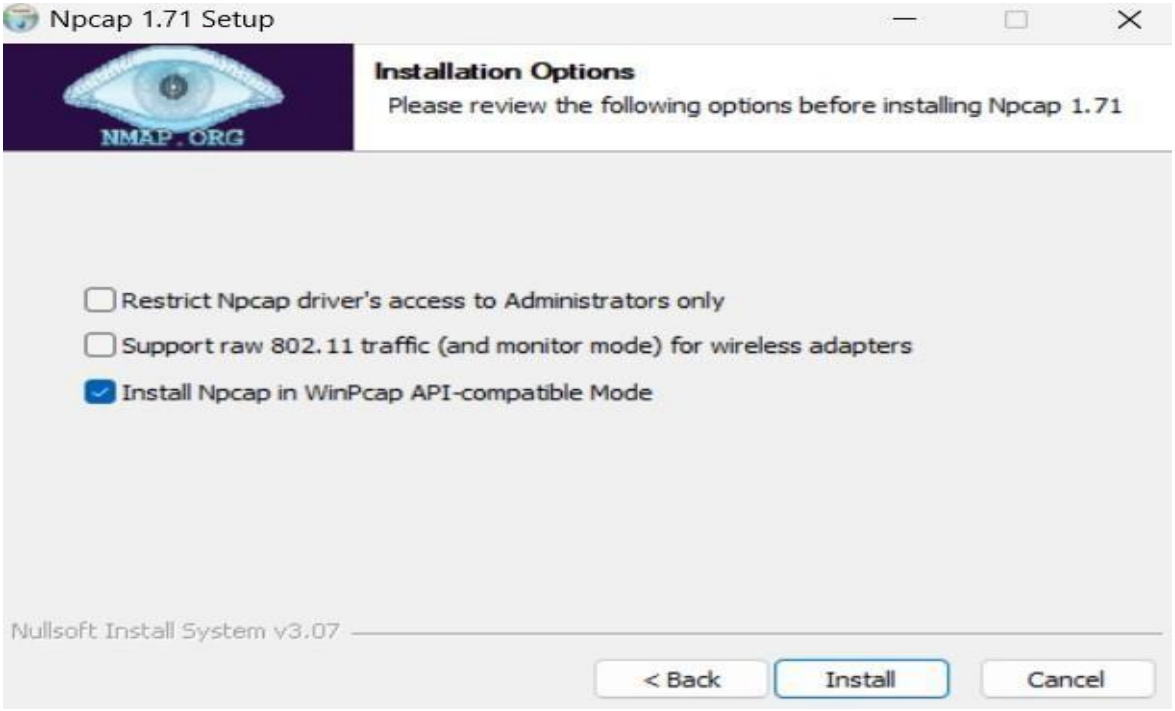


Step 7: After this installation process it will take a few minutes to complete the installation.

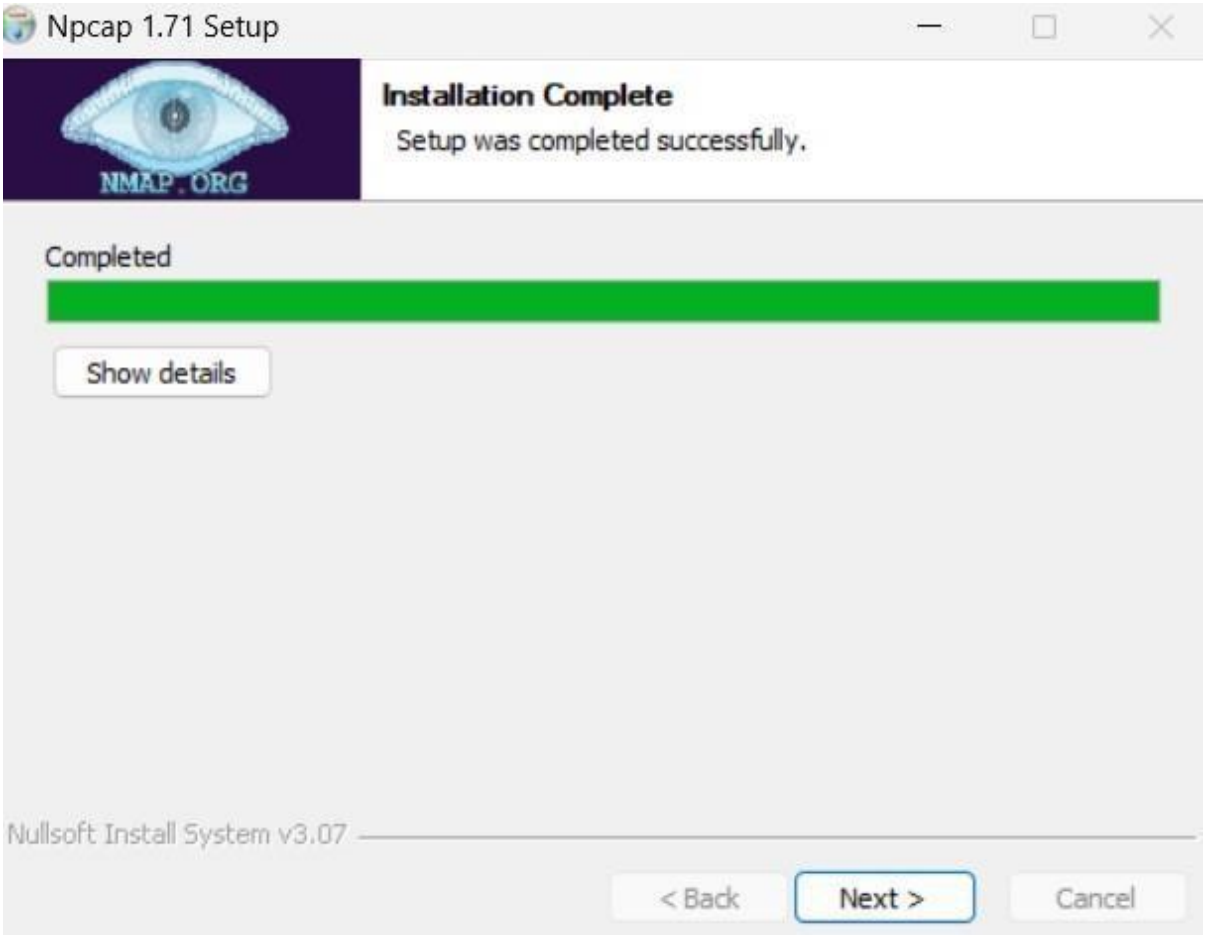


Step 8: Npcap installation will also occur with it, the screen of License Agreement will appear, click on I Agree.

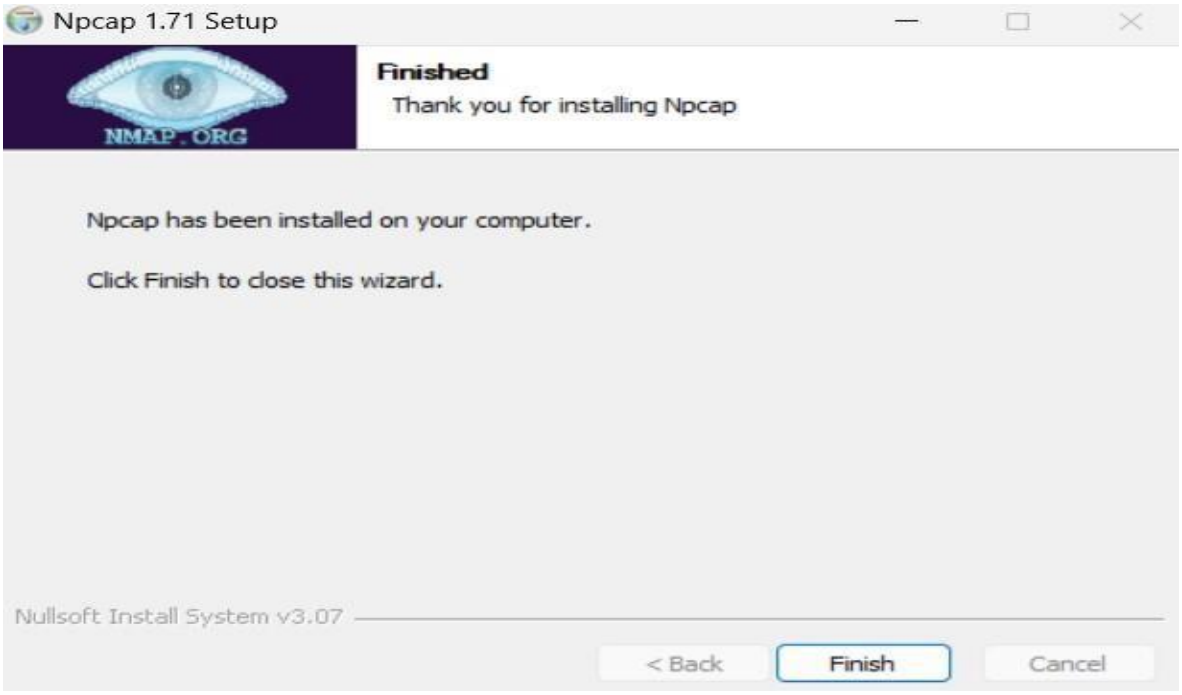
Step 9: Next Screen is of installation options don't change anything and click on the Install button.



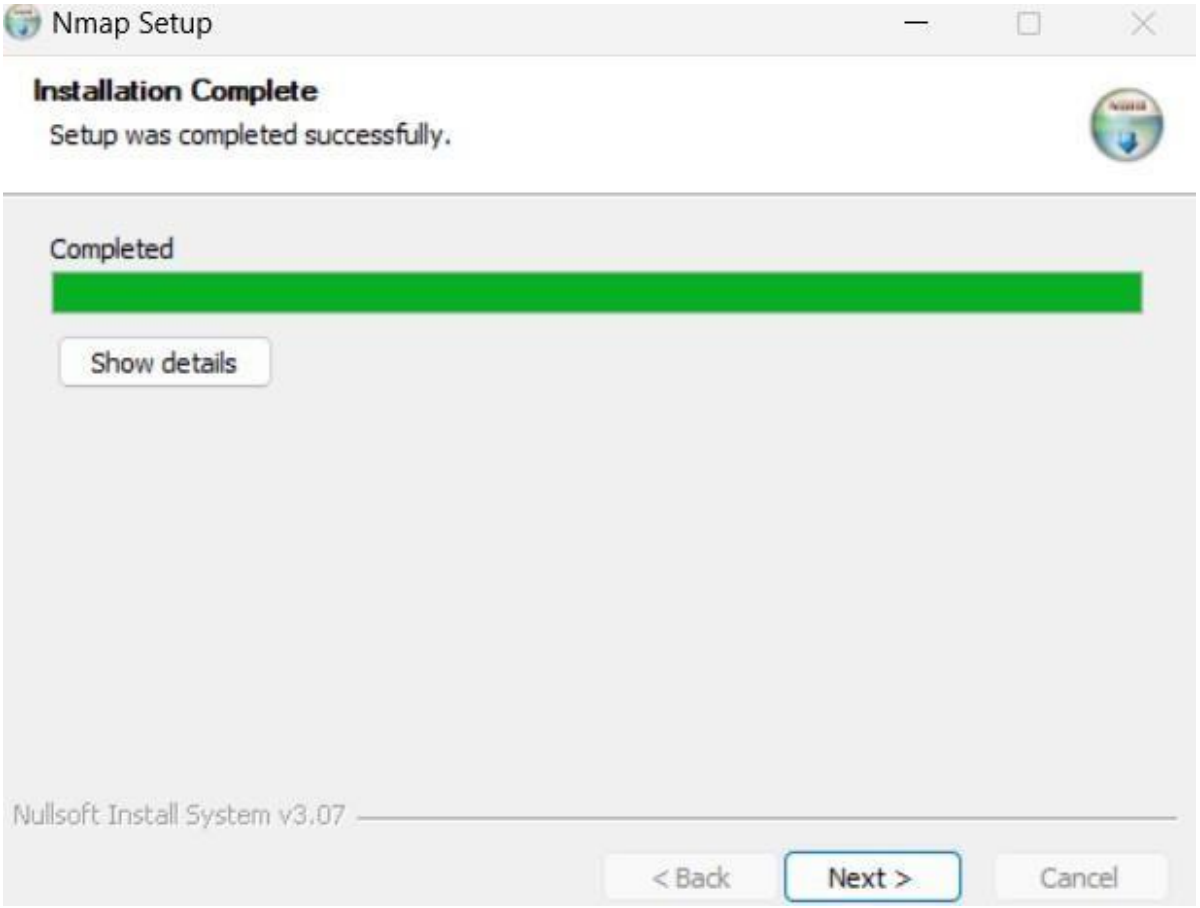
Step 10: After completion of Installation click on Next button.



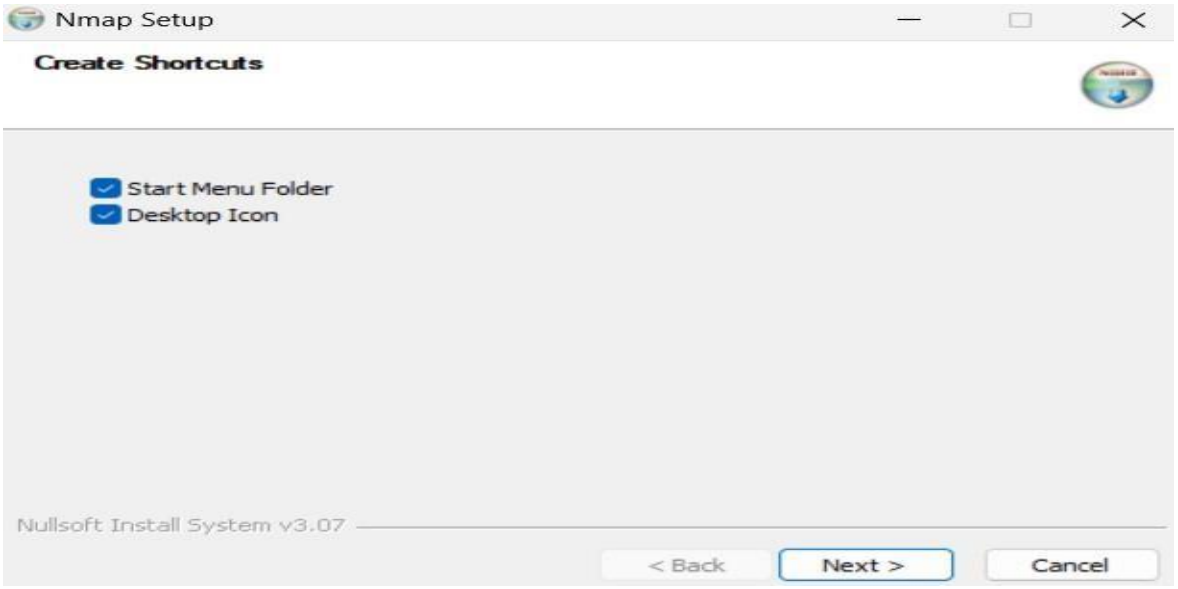
Step 11: Click on the Finish button to Finish the installation of Npcap.



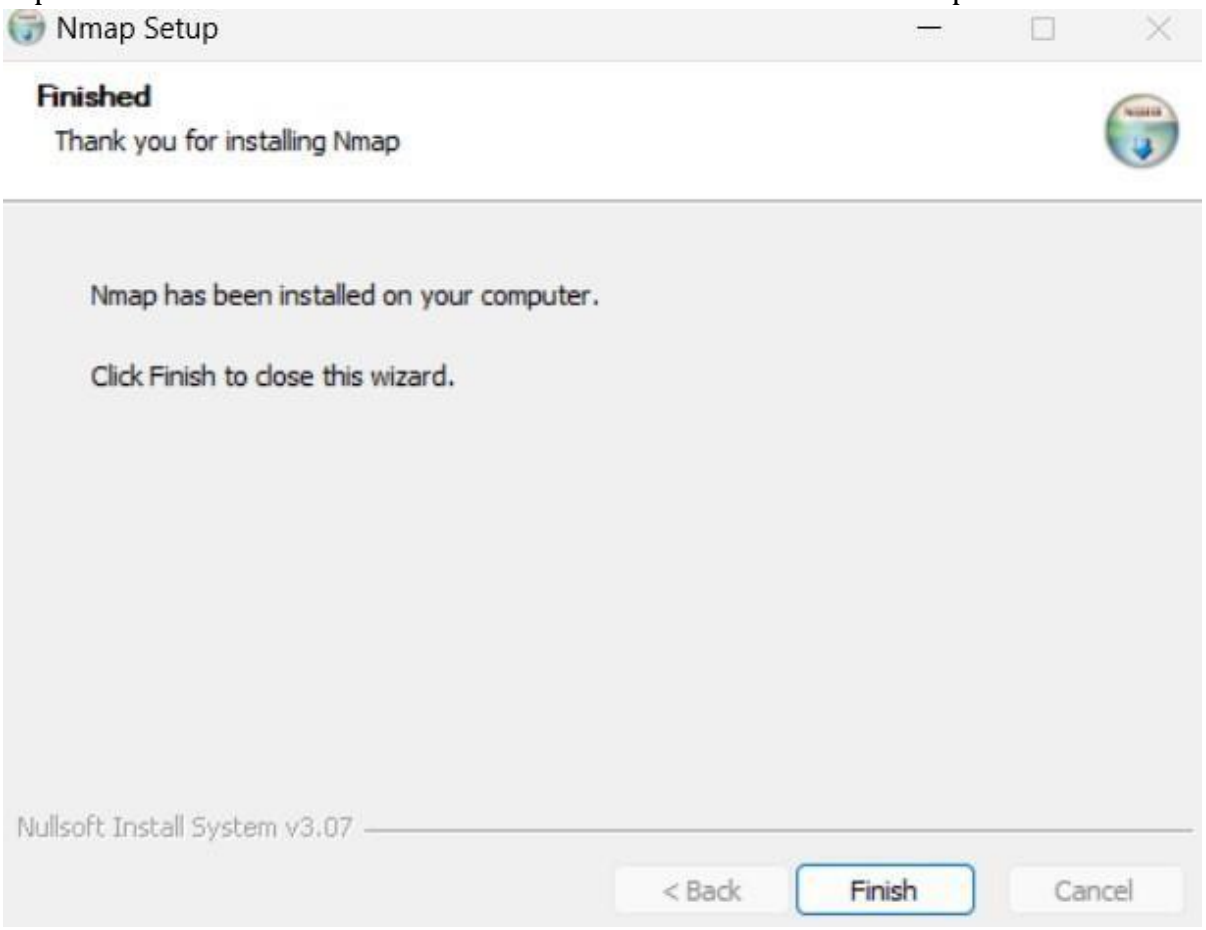
Step 12: After completion of the installation of Nmap click on **Next** button.



Step 13: Screen for creating shortcut will appear, click on Next button.

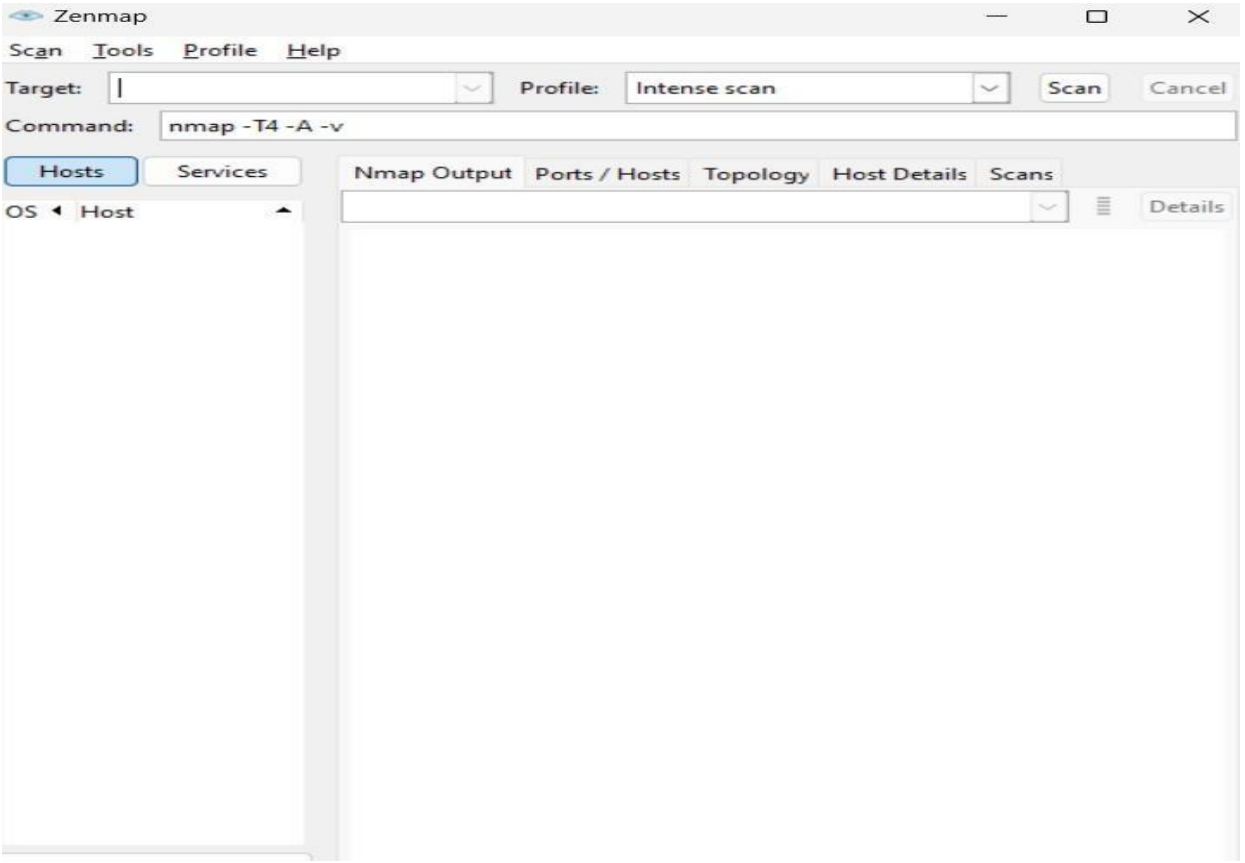


Step 14: Click on the Finish button to finish the installation of Nmap.



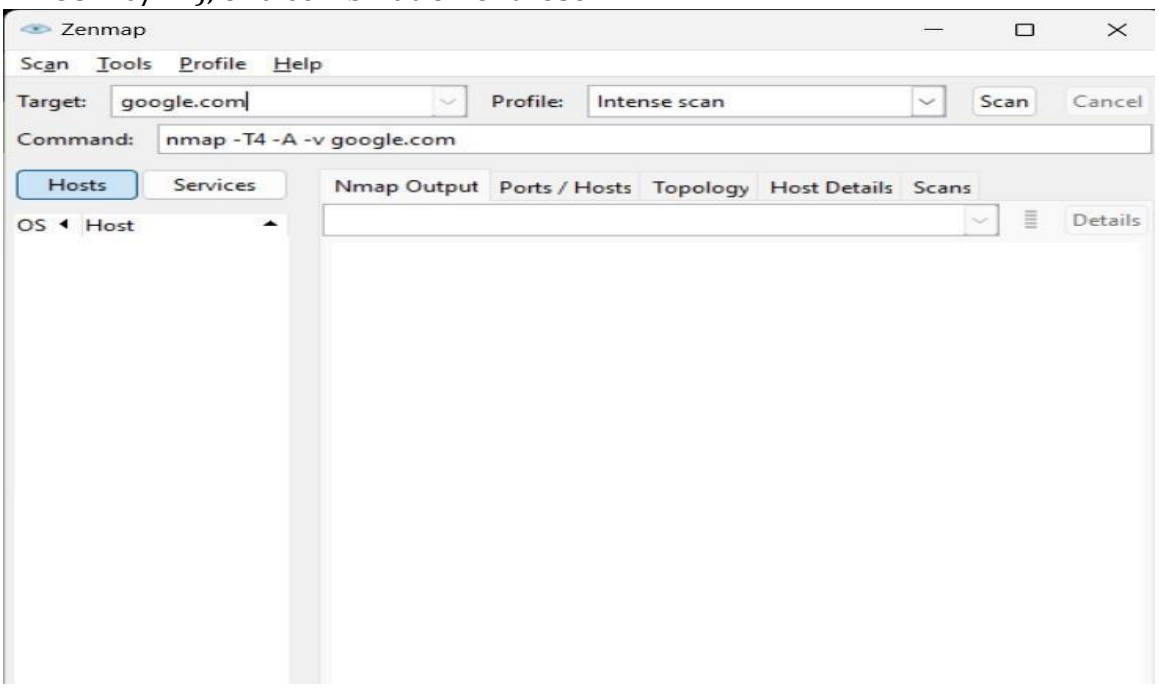
Step 15: Nmap is successfully installed on the system and an icon is created on the desktop.

Step 16: Run the Software and see the interface as below.

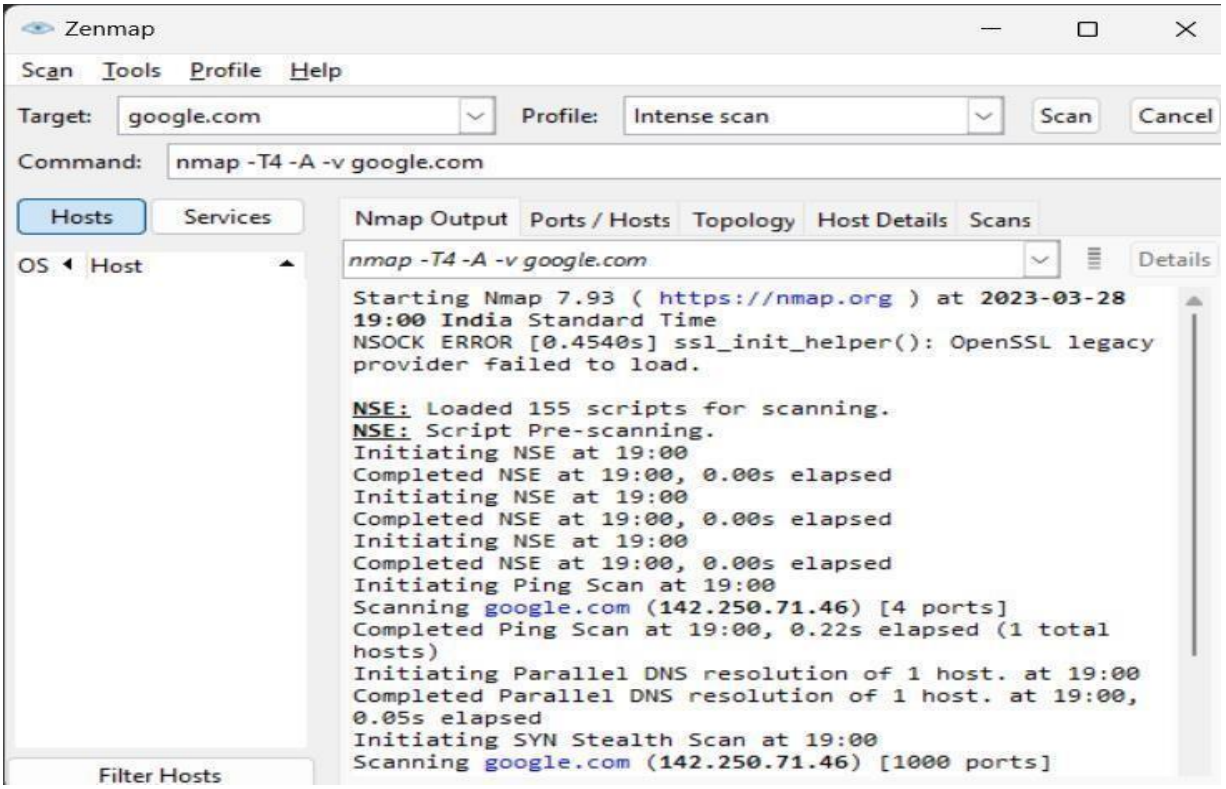


USING ZENMAP:

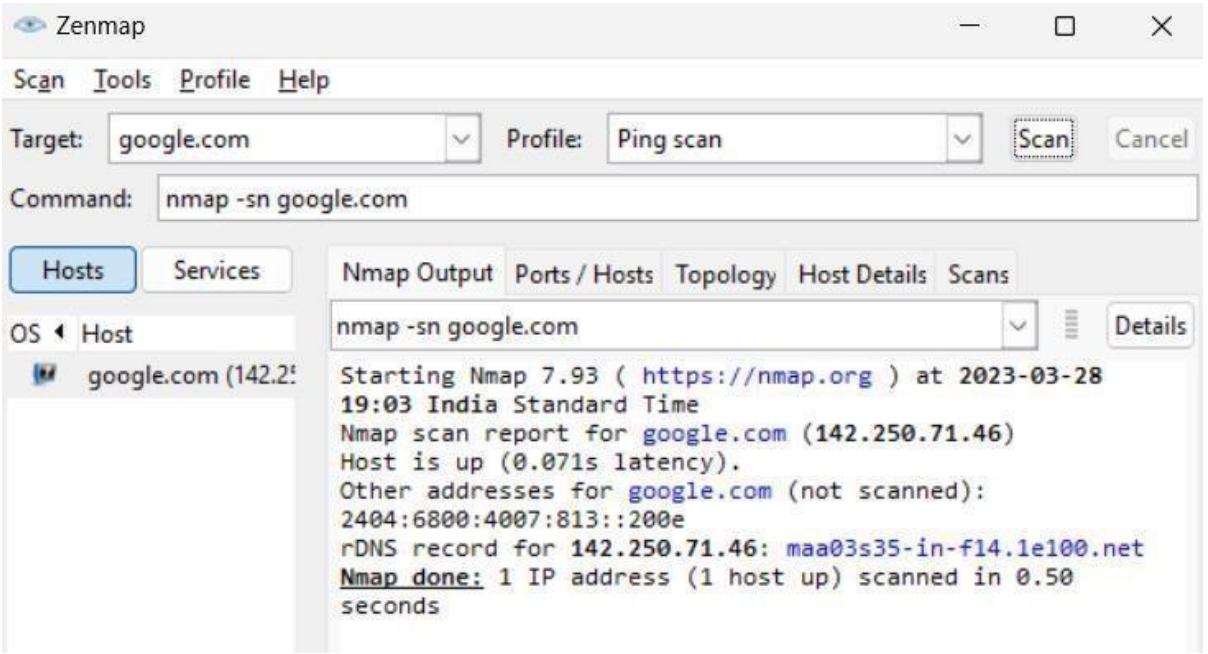
The Zenmap program makes scanning a fairly simple process. The first step to running a scan is choosing your target. You can enter a domain (example.com), an IP address (127.0.0.1), a network (192.168.1.0/24), or a combination of those.



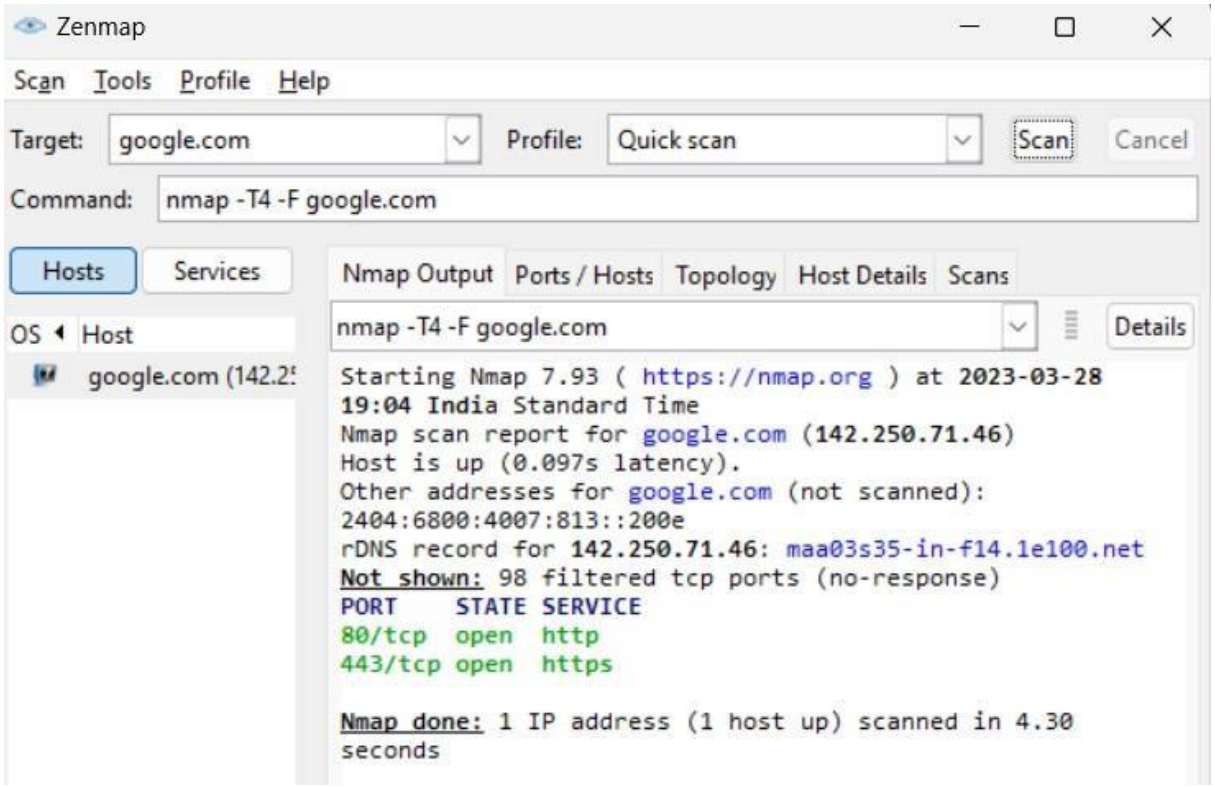
1. Intense Scan:



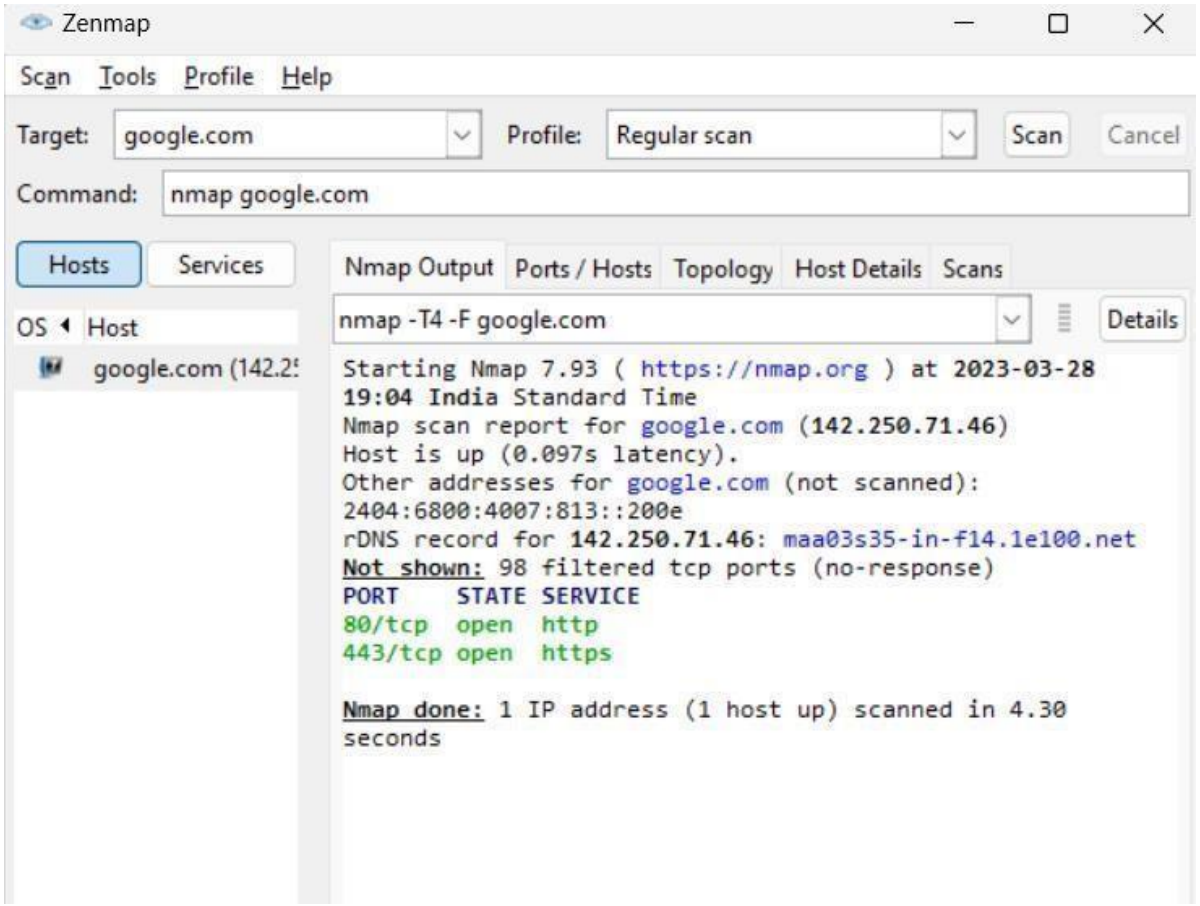
2. Ping Scan:



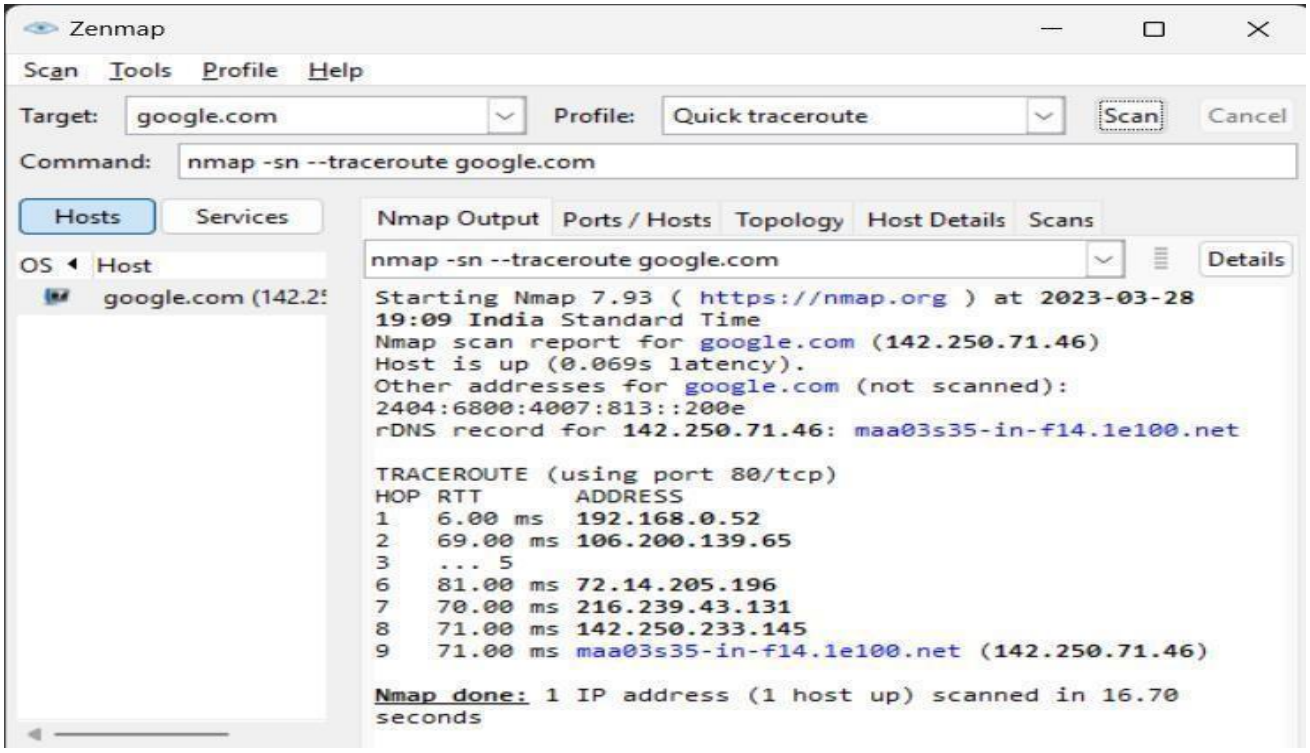
3. Quick Scan:



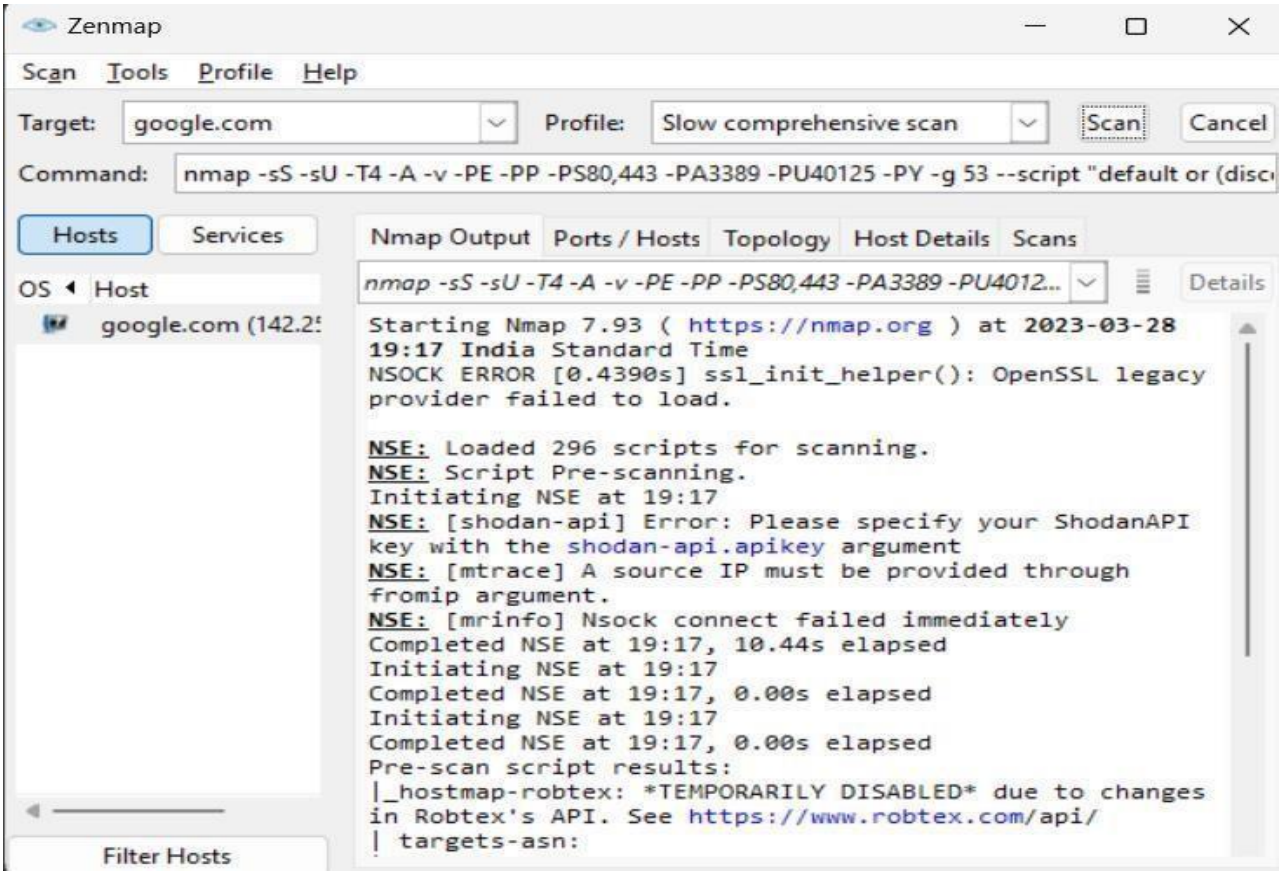
4. Regular Scan:



5. Quickscan traceroute:



6. Slow comprehensive scan:



8. IPTABLES IN LINUX

DESCRIPTION

Iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a 'target', which may be a jump to a user-defined chain in the same table.

Targets

A firewall rule specifies criteria for a packet, and a target. If the packet does not match, the next rule in the chain is the examined; if it does match, then the next rule is specified by the value of the target, which can be the name of a user-defined chain or one of the special values ACCEPT, DROP, QUEUE, or RETURN.

COMMANDS

These options specify the specific action to perform. Only one of them can be specified on the command line unless otherwise specified below. For all the long versions of the command and option names, you need to use only enough letters to ensure that iptables can differentiate it from all other options.

-A, --append chain rule-specification

Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

Syntax: iptables [-t table] --append [chain]

[parameters]

-D, --delete chain rule-specification

Delete one or more rules from the selected chain. There are two versions of this command: the rule can be specified as a number in the chain (starting at 1 for the first rule) or a rule to match.

Syntax:

iptables [-t table] --delete [chain] [rule_number]

-I, --insert chain [rulenum] rule-specification

Insert one or more rules in the selected chain as the given rule number. So, if the rule number is 1, the rule or rules are inserted at the head of the chain. This is also the default if no rule number is specified.

-R, --replace chain rulenum rule-specification

Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1.

-L, --list [chain]

PARAMETERS

The following parameters make up a rule specification (as used in the add, delete, insert, replace and append commands).

-p, --protocol [!] protocol

The protocol of the rule or of the packet to check. The specified protocol can be one of tcp, udp, icmp, or all, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to all. Protocol all will match with all protocols and is taken as default when this option is omitted.

-s, --source [!] address[/mask]

Source specification.

Syntax:

iptables [-t table] -A [chain] -s {source_address} [target]

-d, --destination [!] address[/mask]

Destination specification.

Syntax:

iptables [-t table] -A [chain] -d {destination_address} [target]

Example: This command appends a rule in the OUTPUT chain to drop all packets destined for 192.168.1.123.

iptables -t filter -A OUTPUT -d 192.168.1.123 -j DROP

-j, --jump target

This specifies the target of the rule

Syntax:

iptables [-t table] -A [chain] [parameters] -j {target}

-g, --goto chain

This specifies that the processing should continue in a user specified chain. Unlike the --jump option return will not continue processing in this chain but instead in the chain that called us via --jump.

-i, --in-interface [!] name

Name of an interface via which a packet was received

Syntax:

iptables [-t table] -A [chain] -i {interface} [target]

-o, --out-interface [!] name

Name of an interface via which a packet is going to be sent

Conclusion:

There are many other firewall utilities and some that may be easier, but iptables is a good learning tool, if only because it exposes some of the underlying net filter structure and because it is present in so many systems.

9. Demonstrate intrusion detection system (ids) using any tool (snort or any other s/w).

Aim: Installing Snort 2.9.17 on Windows 10.

Installing Snort 2.9.17 on Windows 10 :

For Windows 10 64 bit supported SNORT's executable file can be downloaded from **here**.

1. Open the downloaded snort executable file.

2. Click On 'I Agree' on the license agreement.

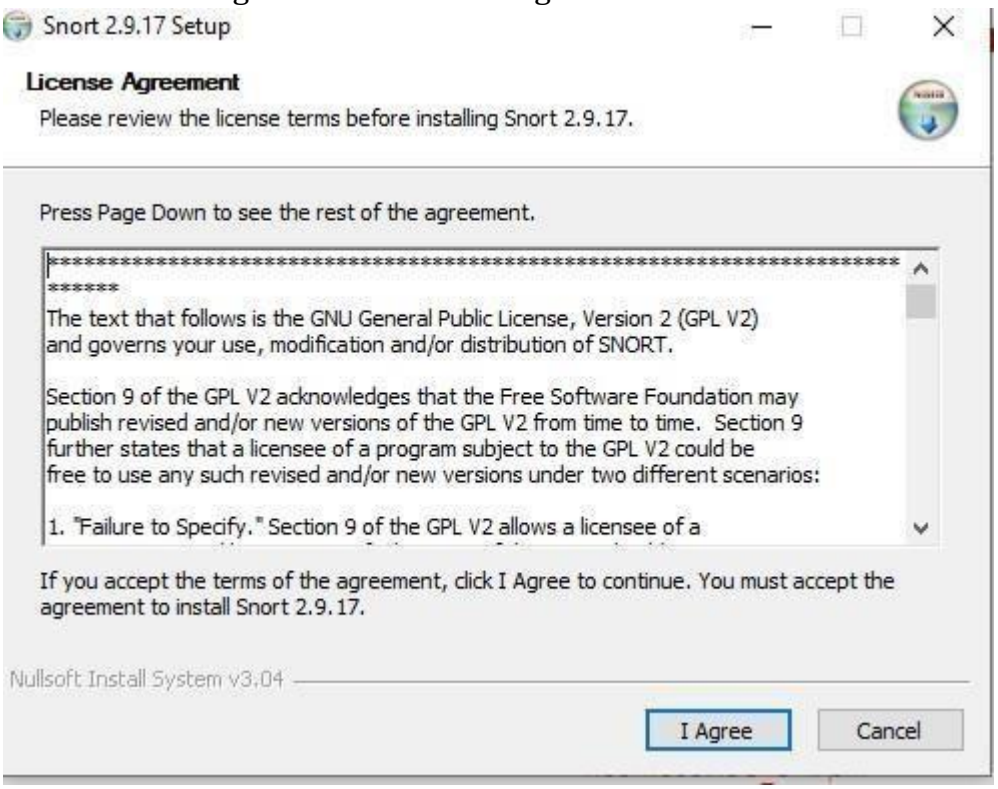


Figure 01: License agreement for Snort 2.9.17

3. Choose components of Snort to be installed.

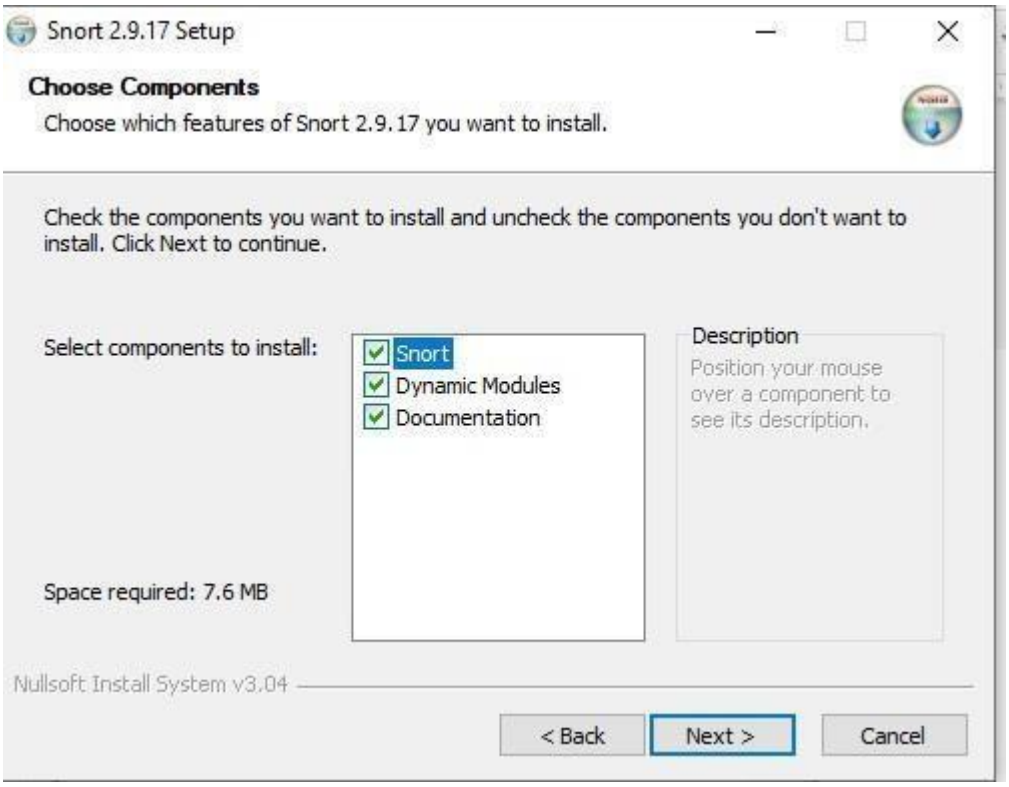


Figure 02: Choosing Components for Snort 2.9.17

4. Click "Next" and then choose install location for snort preferably a separate folder in Windows C Drive.

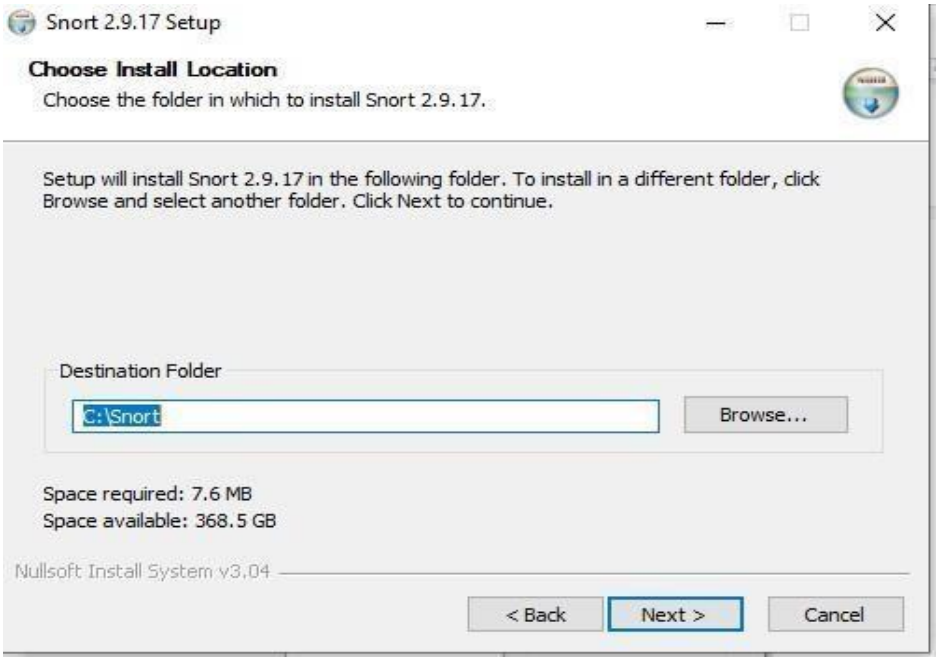


Figure 03: Choose Install location for Snort 2.9.17

5. Click “Next” Installation process starts and then it completes as shown in figure 04:

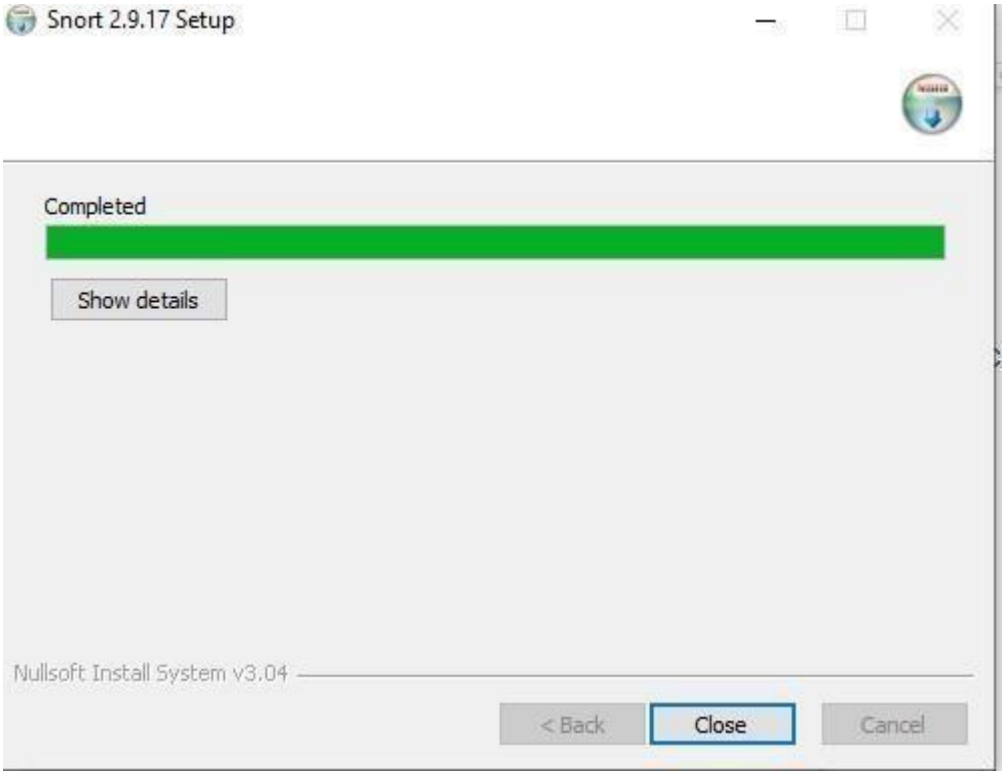


Figure 04: Setup Complete for Snort 2.9.17

6. When you click “ Close” you are prompted with this dialogue box:

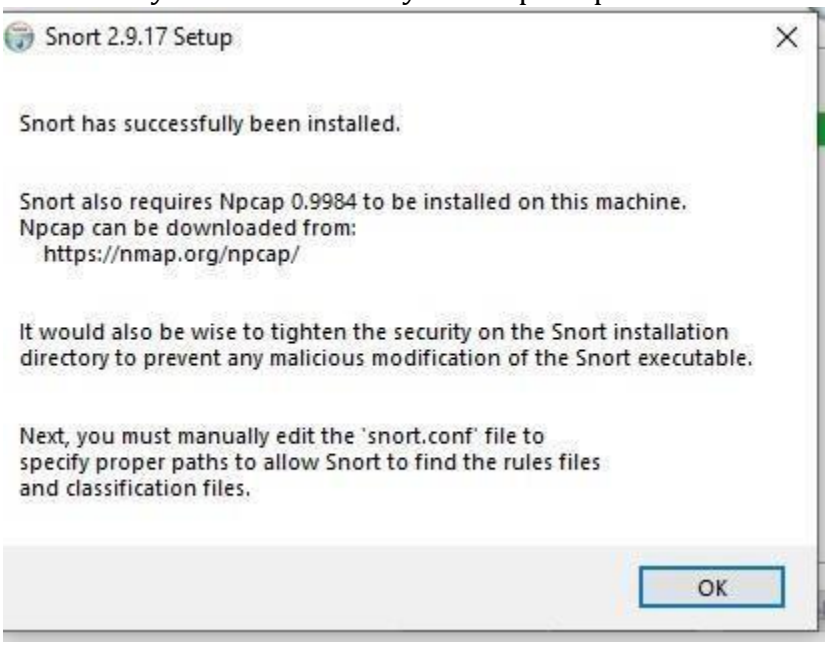


Figure 05: Window showing details of software needed to run Snort successfully

7. Installing Npcap is required by snort for proper functioning.

8. Npcap for Windows 10 can be downloaded from **here**.

9. Opening Npcap setup file, Click on 'I Agree' To license agreement.

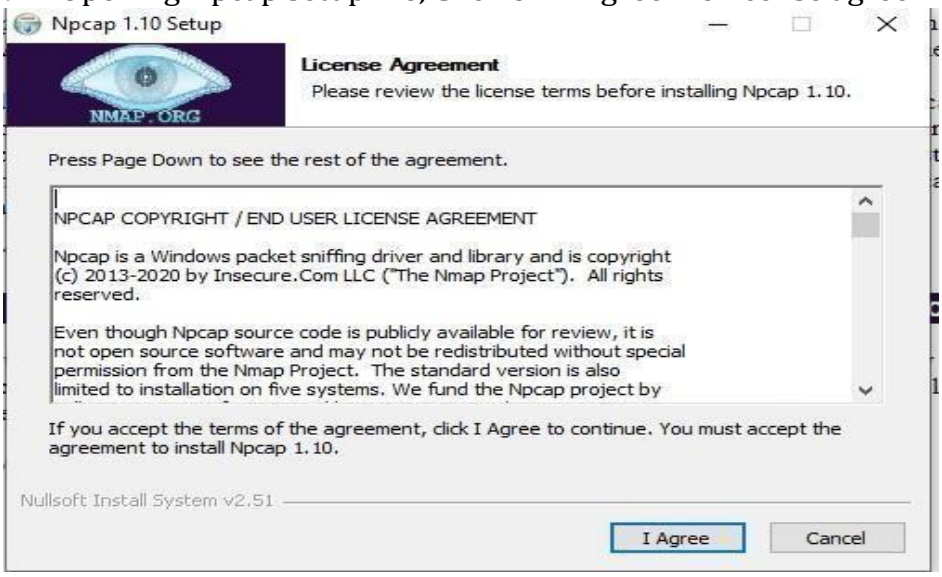


Figure 06: License agreement for Npcap 1.10

10. Now we proceed to choose which components of Npcap are to be installed and then clicking on "Install".

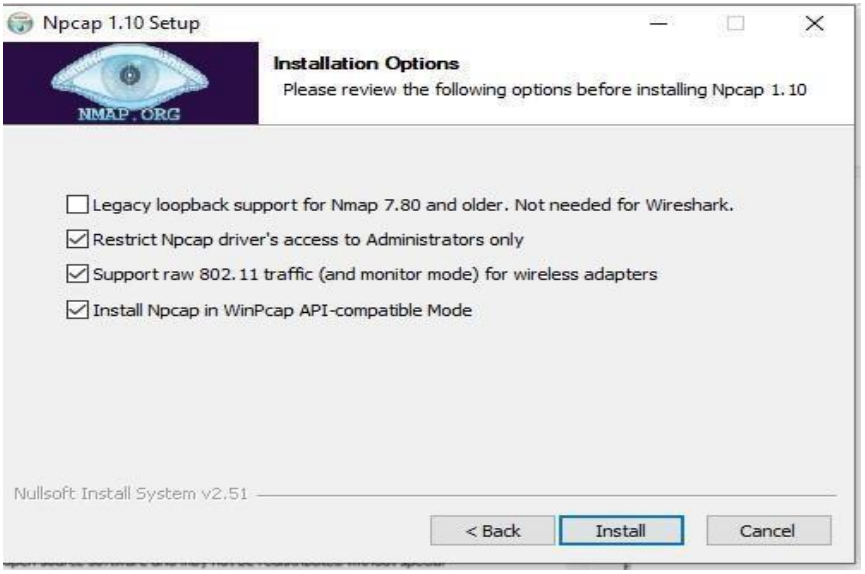


Figure 07: Choose Components to install for Npcap 1.10

11. Installation process starts and completes. Clicking on “Next” we have:

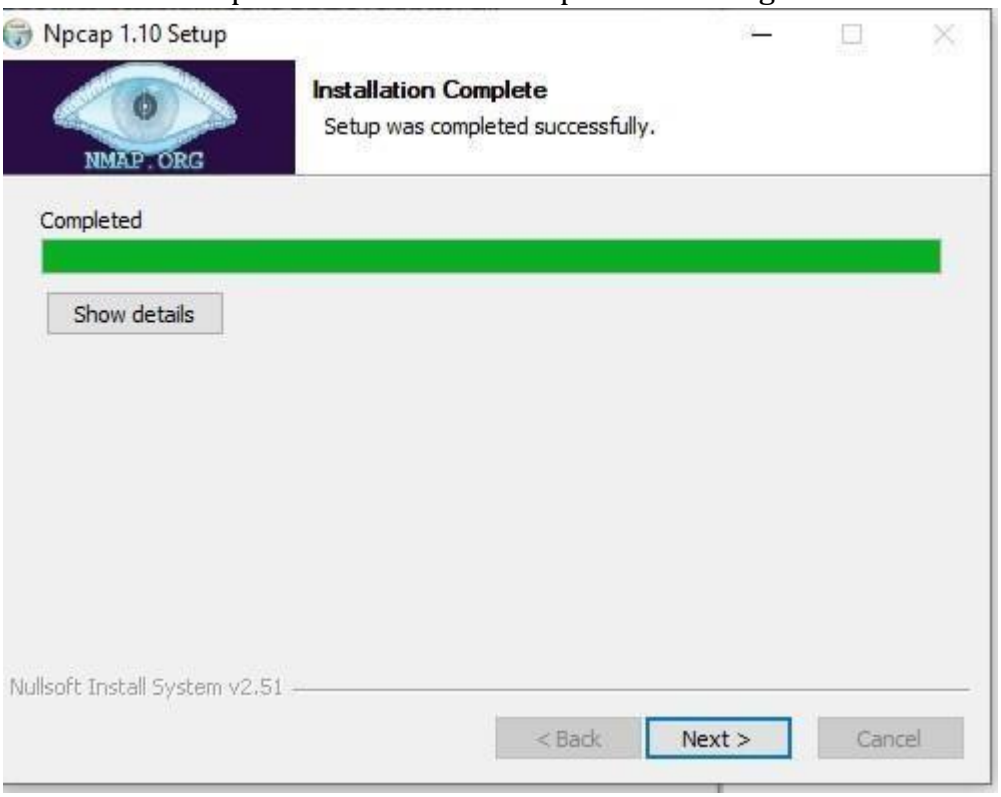


Figure 08: Setup completed for Npcap 1.10

12. Now the window for installation of Npcap shows it has been installed. Clicking “Finish”.

Figure 09: Successful installation for Npcap 1.10 completed

13. After installing Snort and Npcap enter these commands in windows 10 Command prompt to check snorts working

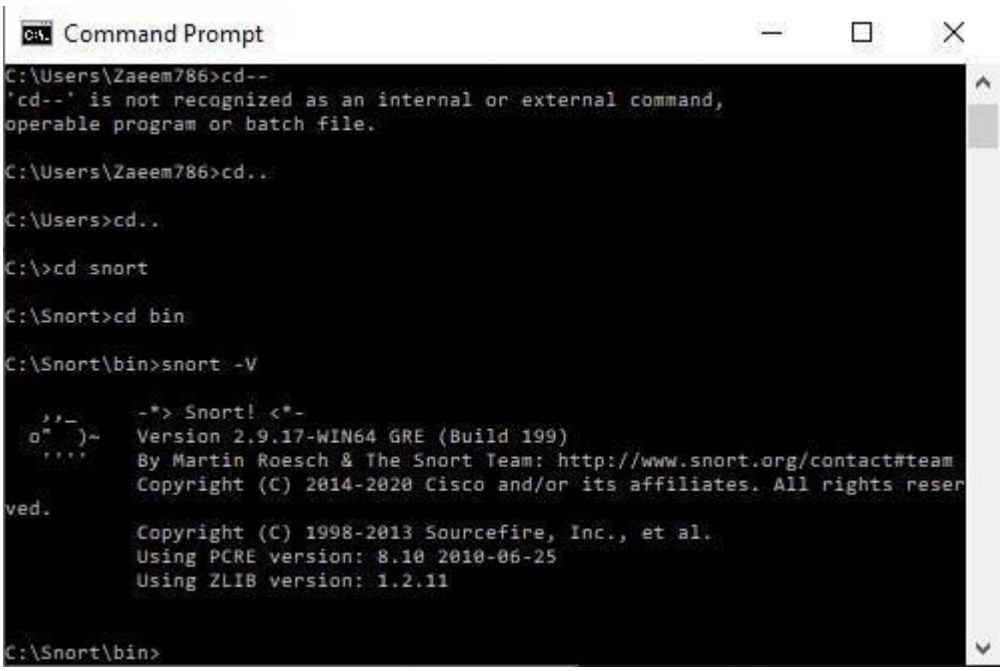


Figure 10: Successfully running Snort on Windows 10 through command prompt

14. As you can see in the above figure that snort runs successfully.

This is how you can download and install Snort along with its dependency i.e. Npcap.

Configuring Snort 2.9.17 on Windows 10:

After installing Snort on Windows 10, Another important step to get started with Snort is configuring it on Windows 10.

Note: The italicized portion with a left hand side border states commands which were pre-written in the configuration file of Snort so we need to make changes according to the commands mentioned in the images, to be precise we need to enter configuration commands as shown in the images to configure snort.

1. Go to this **link** and download latest snort rule file.

2. Extract 3 folders from the downloaded snortrules-snapshot- 29170.tar folder into the Snorts corresponding folders in C drive.

Folders to be extracted are: rules , preproc_rules , etc

- rules folder contains the rules files and the most important local.rules file. Which we will use to enter all our rules.
- etc folder contains all configuration files and the most important file is snort.conf file which we will use for configuration

3. Now open the snort.conf file through the notepad++ editor or any other text editor to edit configurations of snort to make it work like we want it to.

4. Setup the network addresses you are protecting

```
ipvar HOME_NET any
```

Note: Mention your own host IP addresses that you want to protect.

```
# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.100.27/24
```

Figure 11: Setting up the Home Network Address in Snort

5. Setup the external network into anything that is not the home network. That is why ! is used in the command it denotes 'not'.

```
# Set up the external network addresses. Leave as "any" in most situationsipvar
EXTERNAL_NET any
```

```
# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET
```

Figure 12: Setting up the external Network Addresses in Snort

6. Now we have to define the directory for our rules and preproc rules folder

```
# Path to your rules files (this can be a relative path)# Note
for Windows users: You are advised to make this an absolute
path,# such as: c:\snort\rulesvar RULE_PATH ../rulesvar
SO_RULE_PATH ../so_rulesvar PREPROC_RULE_PATH ../preproc_rules
```

```
101  # Path to your rules files (this can be a relative path)
102  # Note for Windows users:  You are advised to make this an absolute path,
103  # such as:  c:\Snort\rules
104  var RULE_PATH c:\Snort\rules
105  # var SO_RULE_PATH ../so_rules
106  var PREPROC_RULE_PATH c:\Snort\preproc_rules
```

Figure 13: Setting up path to our rules files and preproc rules folder in Snort

7. Now we have to setup our white list and black list path it will be in our snorts' rule folder

```
# If yo   are using reputation preprocessor set thesevar
WHITE_LIST_PATH ../rulesvar BLACK_LIST_PATH ../rules
```

```
113  var WHITE_LIST_PATH c:\Snort\rules
114  var BLACK_LIST_PATH c:\Snort\rules
```

Figure 14: Setting up our White List and Black List files paths in Snort

8. Next we have to enable to log directory, so that we store logs in our log folder. Uncomment this line and set absolute path to log directory

```
# Configure default log directory for snort to log to. For more information see snort -
h command line options (-l)## config logdir:
```

```
186  config logdir: c:\Snort\log
```

Figure 15: Setting up Log Directory Path in Snort

9. Now we will set the path to dynamic preprocessors and dynamic engine

```
# path to dynamic preprocessor libraries  dynamic
preprocessor
directory /usr/local/lib/snort_dynamicpreprocessor
```

```
246  # path to dynamic preprocessor libraries
247  dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
```

Figure 16: Setting up path to dynamic preprocessors and dynamic engine in Snort

10. We will do same thing for dynamic preprocessor engine

```
# path to base preprocessor enginedynamicengine
/usr/local/lib/snort_dynamicengine/libsf_engine.so
```

```
249  # path to base preprocessor engine
250  dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
```

Figure 17: Setting up the path to dynamic preprocessor engine in Snort

11. Now lets set our reputation preprocessors:

```
# path to dynamic rules libraries# dynamicdetection directory
/usr/local/lib/snort_dynamicrules
```

```
252  # path to dynamic rules libraries
253  # dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

Figure 18: Path to dynamic rules libraries in Snort

12. Just comment out these lines as shown in figure 19 in doing so we are excluding packet normalization of different packets.

```
263  # Inline packet normalization. For more information, see README.normalize
264  # Does nothing in IDS mode
265  # preprocessor normalize_ip4
266  # preprocessor normalize_tcp: ips ecn stream
267  # preprocessor normalize_icmp4
268  # preprocessor normalize_ip6
269  # preprocessor normalize_icmp6
```

Figure 19: Commenting out packet normalization commands in Snort

13. Scroll down to the reputation preprocessors. We will just change the name of the files since white list , black list are not rules they are just the list of IP addresses labelled as black or white

```
# Reputation preprocessor. For more information see
README.reputationpreprocessor reputation: \memcap 500, \priority whitelist,
\nested_ip inner, \whitelist $WHITE_LIST_PATH/whitelist,
\blacklist
$BLACK_LIST_PATH\black.list
```

```
511     whitelist $WHITE_LIST_PATH/white.list, \
512     blacklist $BLACK_LIST_PATH\black.list
```

Figure 20: Whitelisting and Blacklisting IPs through the command as shown in figure

14. Converted back slashes to forward slashes in lines 546–651.

```
545  # site specific rules
546  include $RULE_PATH\local.rules
547
548  include $RULE_PATH\app-detect.rules
549  include $RULE_PATH\attack-responses.rules
550  include $RULE_PATH\backdoor.rules
551  include $RULE_PATH\bad-traffic.rules
552  include $RULE_PATH\blacklist.rules
553  include $RULE_PATH\botnet-cnc.rules
554  include $RULE_PATH\browser-chrome.rules
555  include $RULE_PATH\browser-firefox.rules
556  include $RULE_PATH\browser-ie.rules
557  include $RULE_PATH\browser-other.rules
558  include $RULE_PATH\browser-plugins.rules
559  include $RULE_PATH\browser-webkit.rules
560  include $RULE_PATH\chat.rules
561  include $RULE_PATH\content-replace.rules
562  include $RULE_PATH\ddos.rules
563  include $RULE_PATH\dns.rules
564  include $RULE_PATH\dos.rules
565  include $RULE_PATH\experimental.rules
566  include $RULE_PATH\exploit-kit.rules
567  include $RULE_PATH\exploit.rules
568  include $RULE_PATH\file-executable.rules
569  include $RULE_PATH\file-flash.rules
570  include $RULE_PATH\file-identify.rules
571  include $RULE_PATH\file-image.rules
572  include $RULE_PATH\file-multimedia.rules
573  include $RULE_PATH\file-office.rules
574  include $RULE_PATH\file-other.rules
575  include $RULE_PATH\file-pdf.rules
576  include $RULE_PATH\finger.rules
577  include $RULE_PATH\ftp.rules
578  include $RULE_PATH\icmp-info.rules
579  include $RULE_PATH\icmp.rules
```

Figure 21 : Converted back slashes to forward slashes in specific lines in snort.conf file

```
include $RULE_PATH\rservices.rules
include $RULE_PATH\scada.rules
include $RULE_PATH\scan.rules
include $RULE_PATH\server-apache.rules
include $RULE_PATH\server-iis.rules
include $RULE_PATH\server-mail.rules
include $RULE_PATH\server-mssql.rules
include $RULE_PATH\server-mysql.rules
include $RULE_PATH\server-oracle.rules
include $RULE_PATH\server-other.rules
include $RULE_PATH\server-webapp.rules
include $RULE_PATH\shellcode.rules
include $RULE_PATH\smtp.rules
include $RULE_PATH\snmp.rules
include $RULE_PATH\specific-threats.rules
include $RULE_PATH\spyware-put.rules
include $RULE_PATH\sql.rules
include $RULE_PATH\telnet.rules
include $RULE_PATH\tftp.rules
include $RULE_PATH\virus.rules
include $RULE_PATH\voip.rules
include $RULE_PATH\web-activex.rules
include $RULE_PATH\web-attacks.rules
include $RULE_PATH\web-cgi.rules
include $RULE_PATH\web-client.rules
include $RULE_PATH\web-coldfusion.rules
include $RULE_PATH\web-frontpage.rules
include $RULE_PATH\web-iis.rules
include $RULE_PATH\web-misc.rules
include $RULE_PATH\web-php.rules
include $RULE_PATH\x11.rules
```

Figure 22: Converted back slashes to forward slashes in specific lines in snort.conf file

15. Again just convert forward slashes to backslashes and uncomment the lines below:

```
# decoder and preprocessor event rules# include
$PREPROC_RULE_PATH/preprocessor.rules# include
$PREPROC_RULE_PATH/decoder.rules# include
$PREPROC_RULE_PATH/sensitive-data.rules
```

```

# decoder and preprocessor event rules
  include $PREPROC_RULE_PATH\preprocessor.rules
  include $PREPROC_RULE_PATH\decoder.rules
  include $PREPROC_RULE_PATH\sensitive-data.rules
```

Figure 23 : Converted back slashes to forward slashes in specific lines and uncommenting specific lines in snort.conf file

16. Now we just need to verify the presence of this command at the bottom of snort.conf file.

```
688  # Event thresholding or suppression commands. See threshold.conf
689  include threshold.conf
```

Figure 24: verifying presence of "include threshold.conf" command in snort.conf file

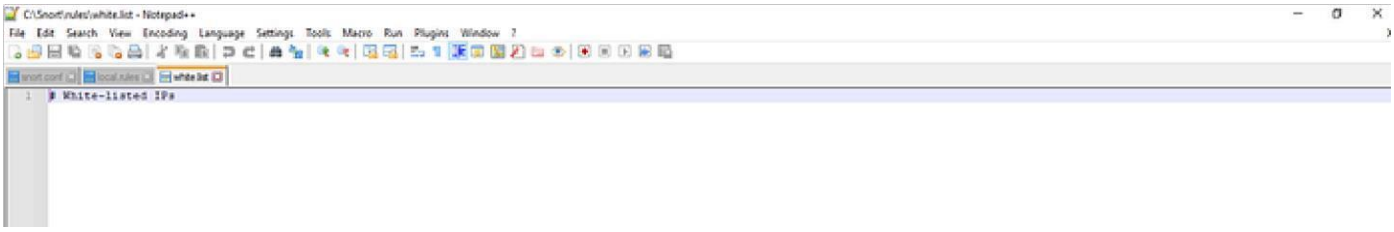
17. Click on Save file and save all changes to save the configuration file (snort.conf).

18. Now recalling the **Step 13** white list , black list are not rules they are just the list of IP addresses labelled as black or white right now these files don't exist in our rule path which is why we have to create them manually , save them in this folder **C:\Snort\rules.**

- Go to Notepad++ and create new file.
- Comment it #White-listed IPs.
- Name the file white.list and save the file.

Figure 25 : Creating White List IPs file

- Create another new file.



- Comment it #Black-listed IPs.
- Name the file black.list and save the file.

Figure 26 : Creating Black List IPs file in Snort

19. Now we test snort again by running Command prompt as admin. To check if it's running fine after all the configurations.

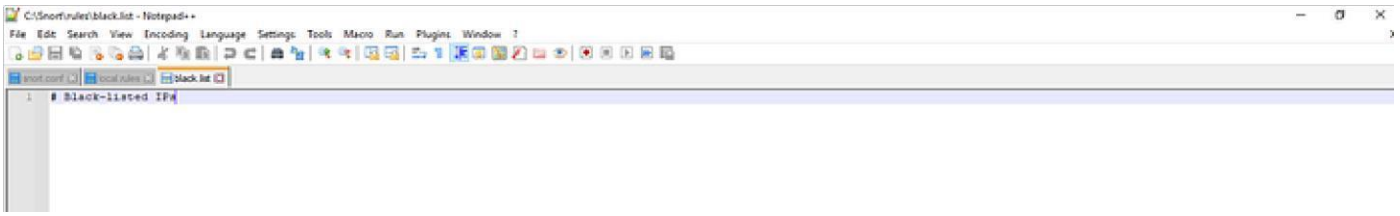
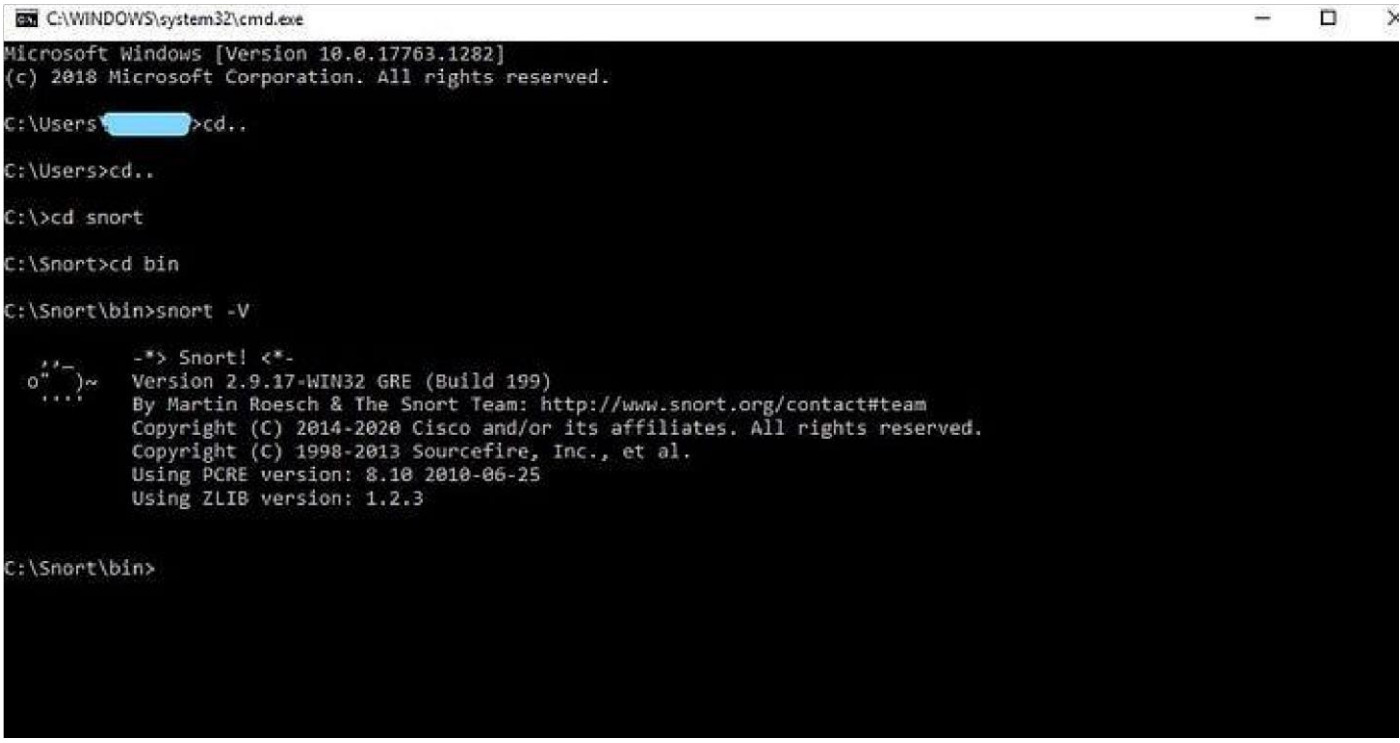


Figure 27: Test Running of Snort in Windows 10 after Configuration

20. We can also the check the wireless interface cards from which we will be using snort by using the command below we can see the list of our



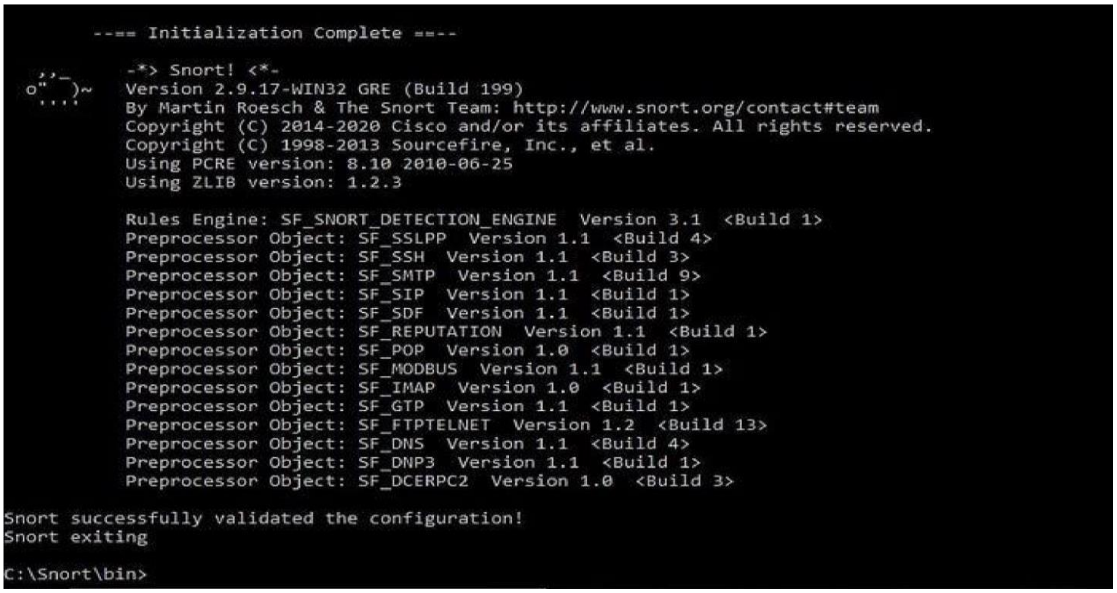
wireless interface cards through entering this command in command prompt.

```
Snort   W
```

21. configuration validation check command:

Now we will enter a command To check validation of snort's configuration

by choosing a specific wireless interface card (1) the rest of command shows the



config file path . The command is :

snort -i 1 -c C:\Snort\etc\snort.conf -T

Figure 28 : Checking Validation of Snort Configuration in Command Prompt

Conclusion:

It can be seen in the given figure that Snort successfully validates our configuration. This brings us to the end of our installation and configuration.