EX.No:1

Date of	Experiment	
	Completion	

Score	/10
Additional Credits	

OBJECTIVE :

The objectives of this program is to integrating symmetric key algorithms into our network security strategy, we can achieve robust message confidentiality and integrity while mitigating the risk of unauthorized access and data tampering during transmission.

ALGORITHM:

1. Generates a secret key using the AES algorithm.
2. Encrypts a message using the generated secret key.
3. Decrypts the encrypted message using the same secret key.
4. Ensure you have the Java Cryptography Extension (JCE) installed and properly configured in your Java environment.
5. When you run this program, it will output the encrypted message and the decrypted message.

PROGRAM:

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import java.security.SecureRandom;
import java.util.Base64;
public class AESExample {
   public static void main(String[] args) throws Exception {
      String plaintext = "Hello, World!";
      SecretKey secretKey = generateAESKey();
      String ciphertext = encryptAES(plaintext, secretKey);
      String decryptedText = decryptAES(ciphertext, secretKey);
      System.out.println("Plaintext: " + plaintext);
      System.out.println("Ciphertext: " + ciphertext);
      System.out.println("Decrypted text: " + decryptedText);
   }
   public static SecretKey generateAESKey() throws Exception {
      KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
```

```
        keyGenerator.init(256, new SecureRandom());
        return keyGenerator.generateKey();
    }
    public static String encryptAES(String plaintext, SecretKey secretKey) throws Exception {
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] ciphertextBytes = cipher.doFinal(plaintext.getBytes());
        return Base64.getEncoder().encodeToString(ciphertextBytes);
    }
    public static String decryptAES(String ciphertext, SecretKey secretKey) throws Exception {
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] ciphertextBytes = Base64.getDecoder().decode(ciphertext);
        byte[] plaintextBytes = cipher.doFinal(ciphertextBytes);
        return new String(plaintextBytes);
    }
}
```

OUTPUT:

```
Plaintext: Hello, World!
Ciphertext: nFbh/NuO/WtMo2weOetOug==
Decrypted text: Hello, World!
```

RESULT:

Thus the above program was executed successfully.

VIVA QUESTIONS:

1. What is a symmetric key algorithm?
2. Can you explain the concept of confidentiality in the context of symmetric key algorithms?
3. How do symmetric key algorithms achieve message confidentiality?
4. What role does key management play in symmetric key algorithms?
5. How do you ensure the security of symmetric keys?
6. How are symmetric keys exchanged securely between communicating parties?
7. What is a Message Authentication Code (MAC), and why is it important in symmetric key cryptography? How does it ensure message integrity?

EX.No:2

Date of	Experiment	
	Completion	

Score	/10
Additional Credits	

OBJECTIVE :

The main objective of this program is to Implementing asymmetric key algorithms and key exchange algorithms to enhance network security by addressing confidentiality, integrity, and authenticity of data transmission.

ALGORITHMS :

1. Demonstrates RSA encryption and decryption using a generated key pair, and Diffie-Hellman key exchange between two parties (A and B).
2. Ensure you have the Java Cryptography Extension (JCE) installed and properly configured in your Java environment.
3. When you run this program, it will output the encrypted and decrypted message for RSA.
4. As well as the shared secrets generated by the Diffie-Hellman key exchange for both parties.

PROGRAM :

```
import javax.crypto.Cipher;
import javax.crypto.KeyAgreement;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.NoSuchAlgorithmException;
import java.security.InvalidKeyException;
import java.security.InvalidAlgorithmParameterException;
import java.security.spec.InvalidParameterSpecException;
import java.security.spec.NamedParameterSpec;
import java.security.spec.XECParameters;
import java.util.Base64;
public class CombinedExample {
   public static void main(String[] args) throws Exception {
```

```
        KeyPair rsaKeyPair = generateRSAKeyPair();
        KeyPair dhKeyPair = generateDHKeyPair();
        byte[] sharedSecret = performDHKeyExchange(dhKeyPair, rsaKeyPair);
        String encryptedSharedSecret = encryptRSA(sharedSecret, rsaKeyPair.getPublic());
        byte[] decryptedSharedSecret = decryptRSA(encryptedSharedSecret, rsaKeyPair.getPrivate());
        System.out.println("Original Shared Secret: " +
Base64.getEncoder().encodeToString(sharedSecret));
        System.out.println("Decrypted Shared Secret: " +
Base64.getEncoder().encodeToString(decryptedSharedSecret));
    }
    public static KeyPair generateRSAKeyPair() throws NoSuchAlgorithmException {
        KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");
        keyPairGenerator.initialize(2048, new SecureRandom());
        return keyPairGenerator.generateKeyPair();
    }
    public static KeyPair generateDHKeyPair() throws NoSuchAlgorithmException,
InvalidAlgorithmParameterException {
        KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("XDH");
        keyPairGenerator.initialize(new NamedParameterSpec("X25519"));
        return keyPairGenerator.generateKeyPair();
    }
    public static byte[] performDHKeyExchange(KeyPair dhKeyPair, KeyPair rsaKeyPair) throws
NoSuchAlgorithmException, InvalidKeyException {
        KeyAgreement keyAgreement = KeyAgreement.getInstance("XDH");
        keyAgreement.init(dhKeyPair.getPrivate());
        keyAgreement.doPhase(rsaKeyPair.getPublic(), true);
        return keyAgreement.generateSecret();
    }
    public static String encryptRSA(byte[] plaintext, PublicKey publicKey) throws Exception {
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.ENCRYPT_MODE, publicKey);
        byte[] ciphertextBytes = cipher.doFinal(plaintext);
        return Base64.getEncoder().encodeToString(ciphertextBytes);
    }
    public static byte[] decryptRSA(String ciphertext, PrivateKey privateKey) throws Exception {
        byte[] ciphertextBytes = Base64.getDecoder().decode(ciphertext);
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.DECRYPT_MODE, privateKey);
        return cipher.doFinal(ciphertextBytes);
    }
}
```

OUTPUT:-

Original Shared Secret: Z6ElSYtXXNLeZNm4ctrVr4lKyhHLYtA9lK3E4msj7BM=
Encrypted Shared Secret: EABZso0iQgbj/Q5hVW4iRvJGxLk6L0pH8shv...
Decrypted Shared Secret: Z6ElSYtXXNLeZNm4ctrVr4lKyhHLYtA9lK3E4msj7BM=

RESULT :

Thus the above program was executed successfully.

VIVA QUESTIONS:

1. What are asymmetric key algorithms?
2. Can you explain the concept of public-key cryptography? How are public and private keys used in asymmetric key algorithms?
3. How does RSA encryption work, and what are its key components?
4. What is the role of digital signatures in asymmetric key cryptography?
5. What are the advantages of using asymmetric key algorithms over symmetric key algorithms in terms of key distribution and management?
6. How do Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithms facilitate secure key negotiation between communicating parties?
7. Explain the process of key exchange using the Diffie-Hellman algorithm.

EX.No:3

Date of	Experiment	
	Completion	

Score	/10
Additional Credits	

OBJECTIVE :

The main of implementing digital signature schemes is to provide secure and reliable methods for verifying the authenticity, integrity, and non-repudiation of digital documents or messages.

ALGORITHM :

1. Generates an RSA key pair.
2. Signs a message using the private key.
3. Verifies the signature using the corresponding public key.
4. Ensure you have the Java Cryptography Extension (JCE) installed and properly configured in your Java environment.
5. When you run this program, it will output the generated signature and whether the signature is verified or not.

PROGRAM :

```
import java.security.*;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

public class DigitalSignatureExample {

    public static void main(String[] args) {
        try {
            // Generate Key Pair
            KeyPair keyPair = generateKeyPair();

            // Original message
            String originalMessage = "Hello, world!";

            // Sign the message
            byte[] signature = signMessage(originalMessage, keyPair.getPrivate());
            System.out.println("Signature: " + bytesToHex(signature));

            // Verify the signature
```

```
            boolean verified = verifySignature(originalMessage, signature, keyPair.getPublic());
            System.out.println("Signature Verified: " + verified);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static KeyPair generateKeyPair() throws Exception {
        KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");
        keyPairGenerator.initialize(2048);
        return keyPairGenerator.generateKeyPair();
    }
    public static byte[] signMessage(String message, PrivateKey privateKey) throws Exception {
        Signature signature = Signature.getInstance("SHA256withRSA");
        signature.initSign(privateKey);
        signature.update(message.getBytes());
        return signature.sign();
    }

    public static boolean verifySignature(String message, byte[] signature, PublicKey publicKey)
throws Exception {
        Signature verifier = Signature.getInstance("SHA256withRSA");
        verifier.initVerify(publicKey);
        verifier.update(message.getBytes());
        return verifier.verify(signature);
    }
    public static String bytesToHex(byte[] bytes) {
        StringBuilder hexString = new StringBuilder();
        for (byte b : bytes) {
            String hex = Integer.toHexString(0xff & b);
            if (hex.length() == 1) {
                hexString.append('0');
            }
            hexString.append(hex);
        }
        return hexString.toString();
    }
}
```

OUTPUT:

Signature: <signature_value_in_hexadecimal>

Signature:
ab2e8f30c14336944e6082a0ec89a8b12789cdd4d6dd35e8f3341b161f550984cd595fea0181c4548
b1668e326e7605b163162d96760b372adda33753f21ecd7e170f4b08312cc9804f7664779d62417f
df80f8bd4716939b64da76d24169be225a5fc2aefe610fef473151f05b59a9c8b9aea824a2f045aef9c
5e78a4c0a0c12b3015abda5ede341dffd6e553bf73568bb1bd2585a1a26992b287e1d86a3c1c817fd
3e3f2dd198f4a1f0191542fe32c42fe1ad61e1dff5d1d66b5f995254daea27c30a34040e002f125ac80
eb5a136f9f3d5f9cda69e6c8a15d9f8a76589f8959734303bfa8c1d113596ead3ba9e94706678220af
24d9ab895eefb9e87a9452

Signature Verified: true

RESULT :

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

VIVA QUESTIONS:

1. What is a digital signature, and how does it differ from a handwritten signature?
2. Can you explain the components of a digital signature scheme, including public and private keys?
3. How does a digital signature ensure the authenticity of a document or message?
4. What cryptographic techniques are commonly used in digital signature schemes?
5. Describe the process of generating a digital signature.
6. How does a recipient verify the authenticity of a digital signature?
7. What role do hash functions play in digital signature schemes, and why are they necessary?
8. Discuss the concept of non-repudiation in the context of digital signatures.
9. What measures can be taken to protect the security of digital signatures, especially regarding key management?
10. Explain the difference between a digital signature and a digital certificate.

EX.No:4

OBJECTIVE :

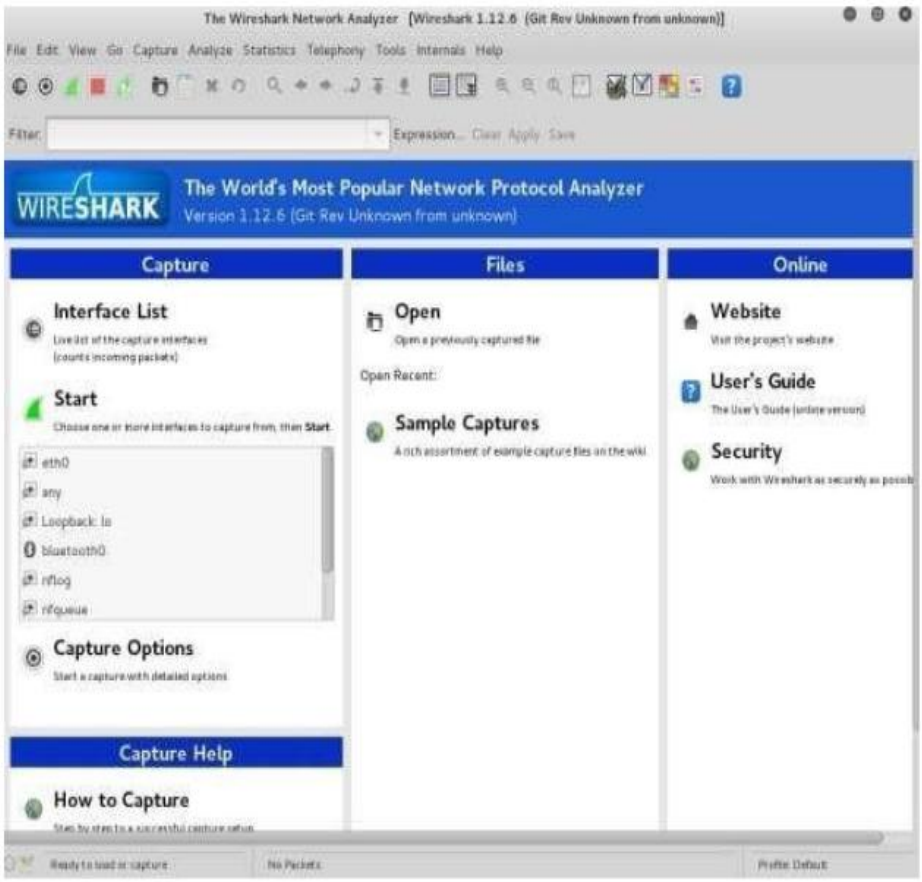
The main objective of installing Wireshark, tcpdump, and observing data transferred in client-server communication using UDP/TCP is to gain insights into network traffic and understand the structure of UDP/TCP datagrams.

ALGORITHM:

1. Start up the Wireshark program (select an interface and press start to capture packets).
2. Start up your favorite browser (ceweasel in Kali Linux).
3. In your browser, go to Wayne State homepage by typing www.wayne.edu.
4. After your browser has displayed the http://www.wayne.edu page
5. Stop Wireshark packet capture by selecting stop in the Wireshark capture window.
6. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture

Introduction:

The first part of the lab introduces packet sniffer, Wireshark. Wireshark is a free open-source network protocol analyzer. It is used for network troubleshooting and communication protocol analysis. Wireshark captures network packets in real time and display them in human-readable format. It provides many advanced features including live capture and offline analysis, three-pane packet browser, coloring rules for analysis. This document uses Wireshark for the experiments, and it covers Wireshark installation, packet capturing, and protocol analysis.



Initial Graphic User Interface of Wireshark

Background

TCP/IP Network Stack

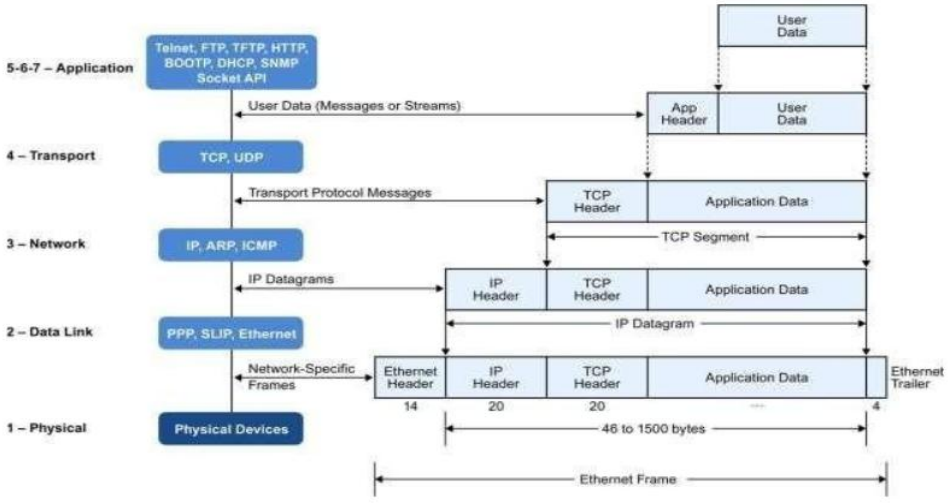


Figure 2: Encapsulation of Data in the TCP/IP Network Stack

In the CSC 4190 Introduction to Computer Networking (one of the perquisite courses), TCP/IP network stack is introduced and studied. This background section briefly explains the concept of TCP/IP network stack to help you better understand the experiments. TCP/IP is the most commonly used network model for Internet services. Because its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP) were the first networking protocols defined in this standard, it is named as TCP/IP. However, it contains multiple layers including application layer, transport layer, network layer, and data link layer.

- Application Layer: The application layer includes the protocols used by most applications for providing user services. Examples of application layer protocols are Hypertext Transfer Protocol (HTTP), Secure Shell (SSH), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).

- Transport Layer: The transport layer establishes process-to-process connectivity, and it provides end-to-end services that are independent of underlying user data. To implement the process-to-process communication, the protocol introduces a concept of port. The examples of transport layer protocols are Transport Control Protocol (TCP) and User Datagram Protocol (UDP). The TCP provides flow- control, connection establishment, and reliable transmission of data, while the UDP is a connectionless transmission model.

- Internet Layer: The Internet layer is responsible for sending packets to across networks. It has two functions: 1) Host identification by using IP addressing system (IPv4 and IPv6); and 2) packets routing from source to destination. The examples of Internet layer protocols are Internet Protocol (IP), Internet Control Message Protocol (ICMP), and Address Resolution Protocol (ARP).

- Link Layer: The link layer defines the networking methods within the scope of the local network link. It is used to move the packets between two hosts on the same link. An common example of link layer protocols is Ethernet.

Packet Sniffer

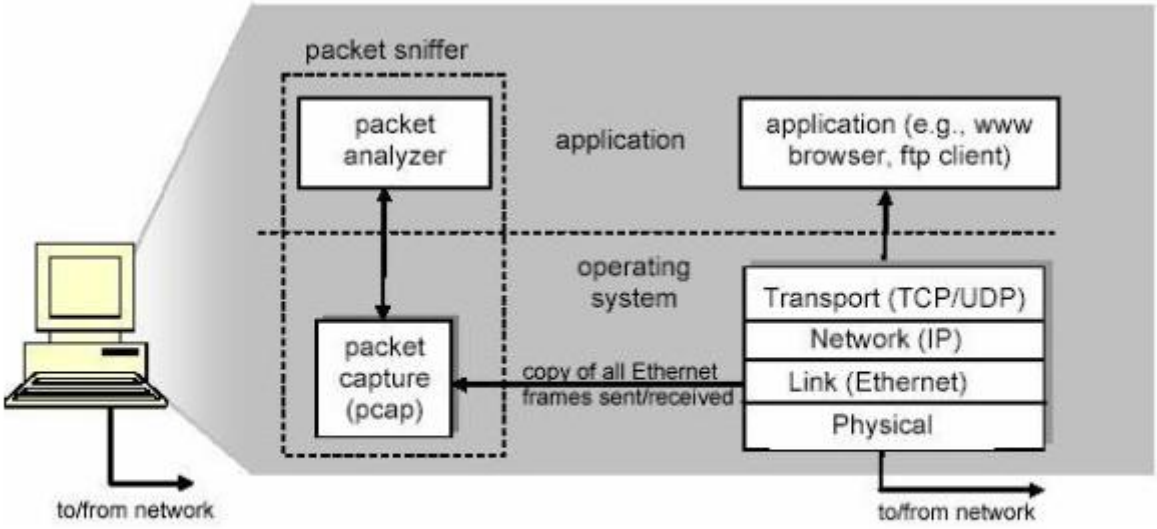
Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures ("sniffs") packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.

Figure 3 shows the structure of a packet sniffer. At the right of Figure 3 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in Figure 3 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. Messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer framesthat are transmitted over physical media such as an Ethernet cable. In Figure 1, the assumed physical media is an Ethernet, and so all upper-layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames

thus gives you access to all messages sent/received from/by all protocols and applications executing in your computer.

The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer

must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the



Packet Sniffer Structure

var ious fields in messages exchanged by the HTTP protocol in Figure 3. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string "GET," "POST," or "HEAD".

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers.

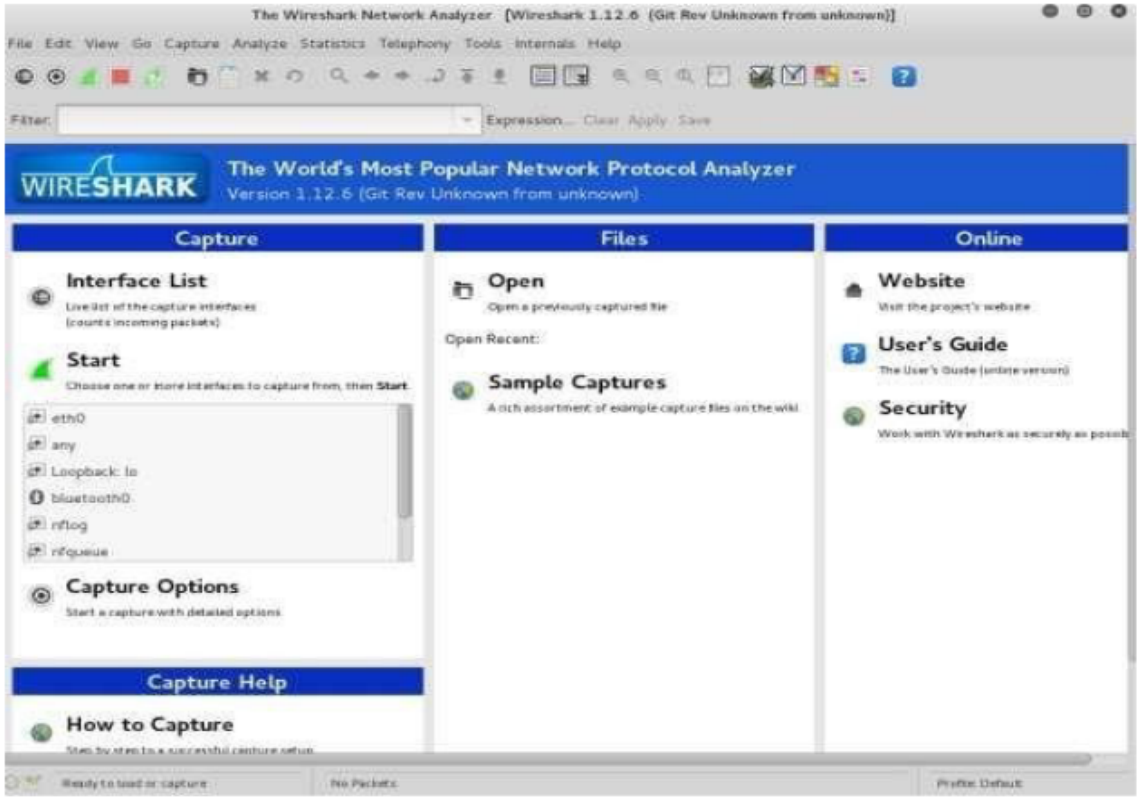
Getting Wireshark

The Kai Linux has Wireshark installed. You can just launch the Kali Linux VM and open Wireshark there.Wireshark can also be downloaded from here: https://www.wireshark.org/download.html.

(Download Page of Wireshark)

Starting Wireshark:

When you run the Wireshark program, the Wireshark graphic user interface will be shown as Figure 5.Currently, the program is not capturing the packets.

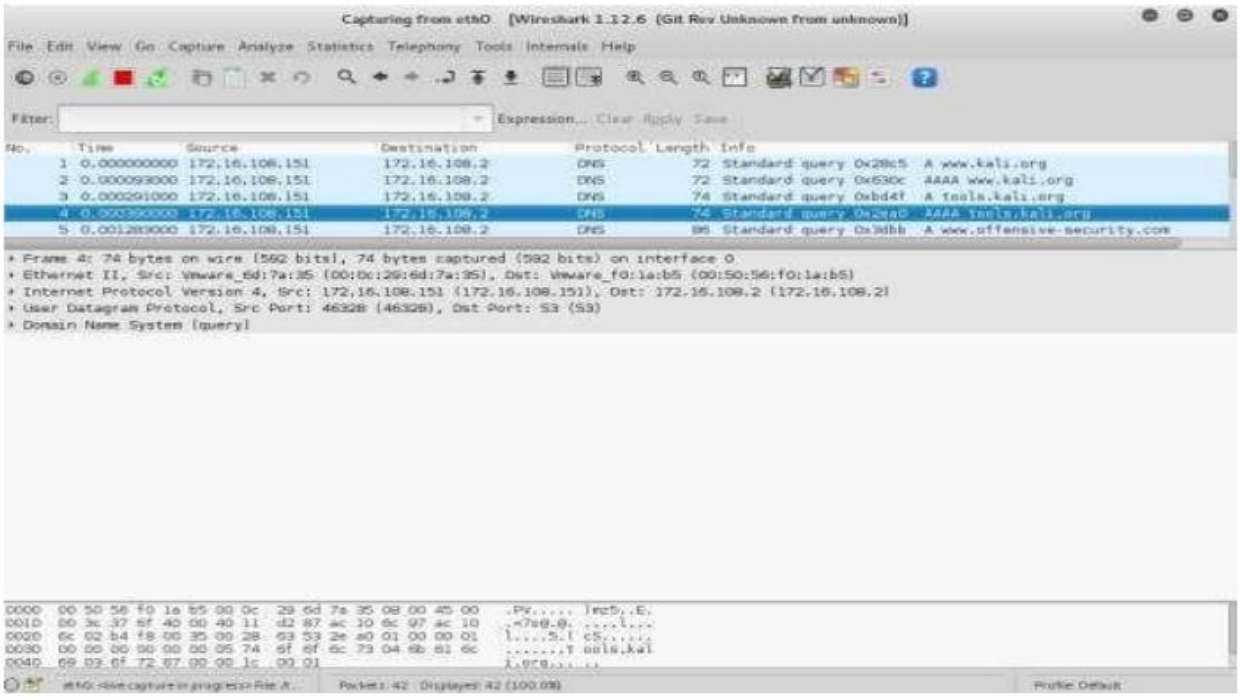


Initial Graphic User Interface of Wireshark

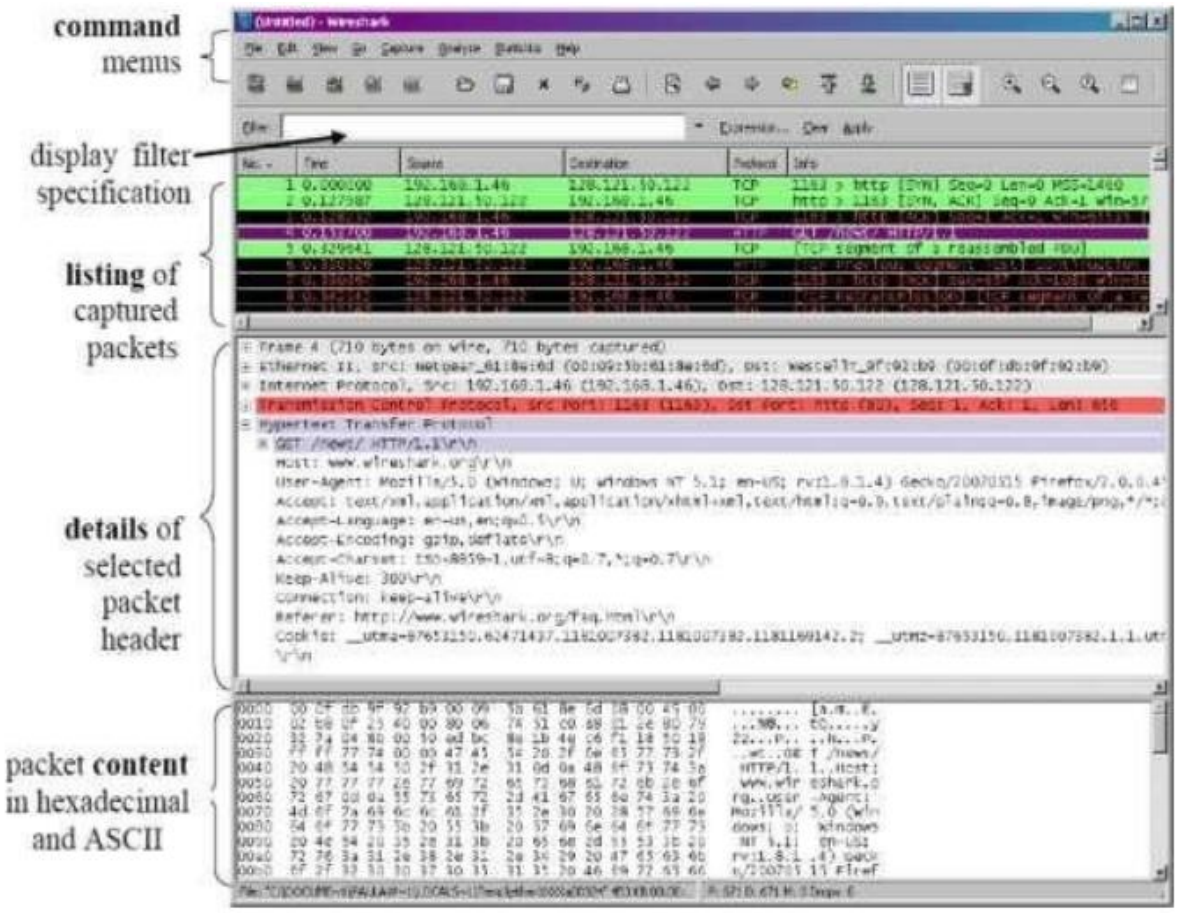
Then, you need to choose an interface. If you are running the Wireshark on your laptop, you need to select WiFi interface. If you are at a desktop, you need to select the Ethernet interface being used. Note that there could be multiple interfaces. In general, you can select any interface but that does not mean that traffic will flow through that interface. The network interfaces (i.e., the physical connections) that your computer has to the network are shown. The attached Figure 6 was taken from my computer.

After you select the interface, you can click start to capture the packets as shown in Figure 7.

Capture Interfaces in Wireshark



Capturing Packets in Wireshark



(Wireshark Graphical User Interface on Microsoft Windows)

The Wireshark interface has five major components:

The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now is the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

The **packet-listing** window displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest- level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

The **packet-header** details window provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one- line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by

clicking on the right- pointing or down- pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

The **packet-contents** window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

Towards the top of the Wireshark graphical user interface, is the packet display filter field, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

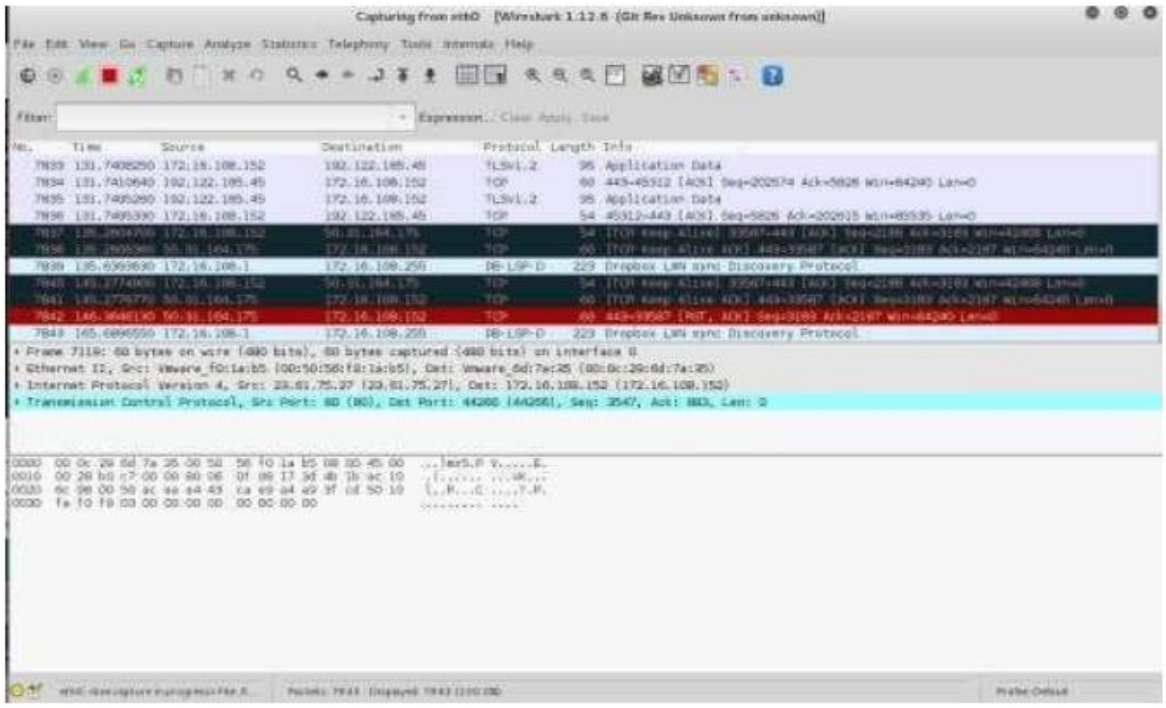
Capturing Packets

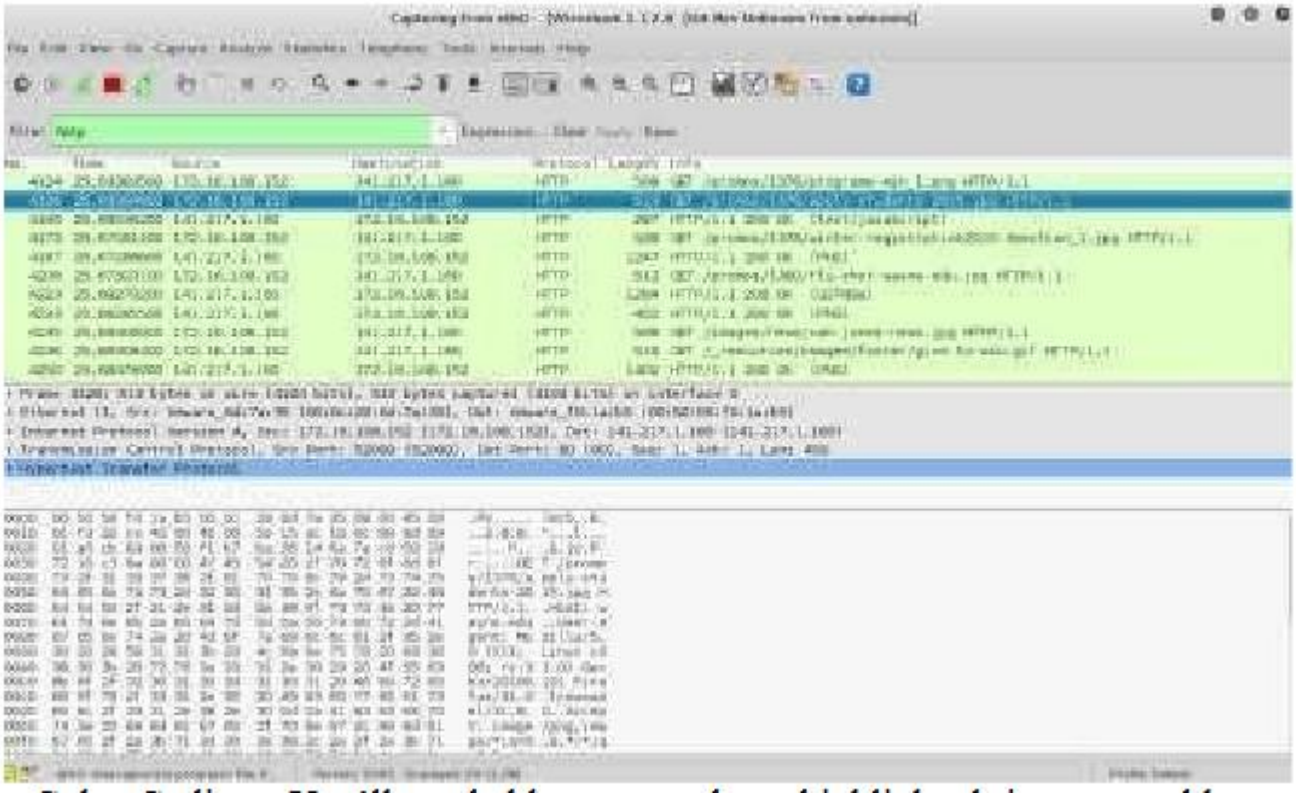
After downloading and installing Wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface.

Test Run

Do the following steps:

1. Start up the Wireshark program (select an interface and press start to capture packets).
2. Start up your favorite browser (ceweasel in Kali Linux).
3. In your browser, go to Wayne State homepage by typing www.wayne.edu.
4. After your browser has displayed the http://www.wayne.edu page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture see image below:

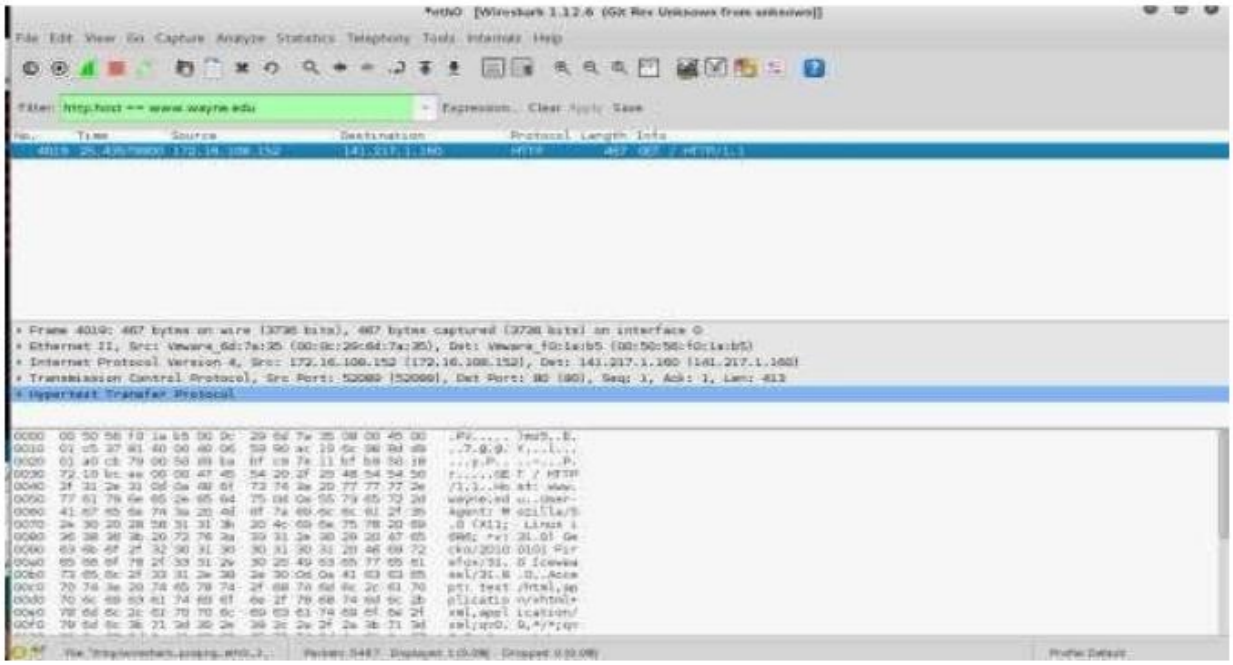




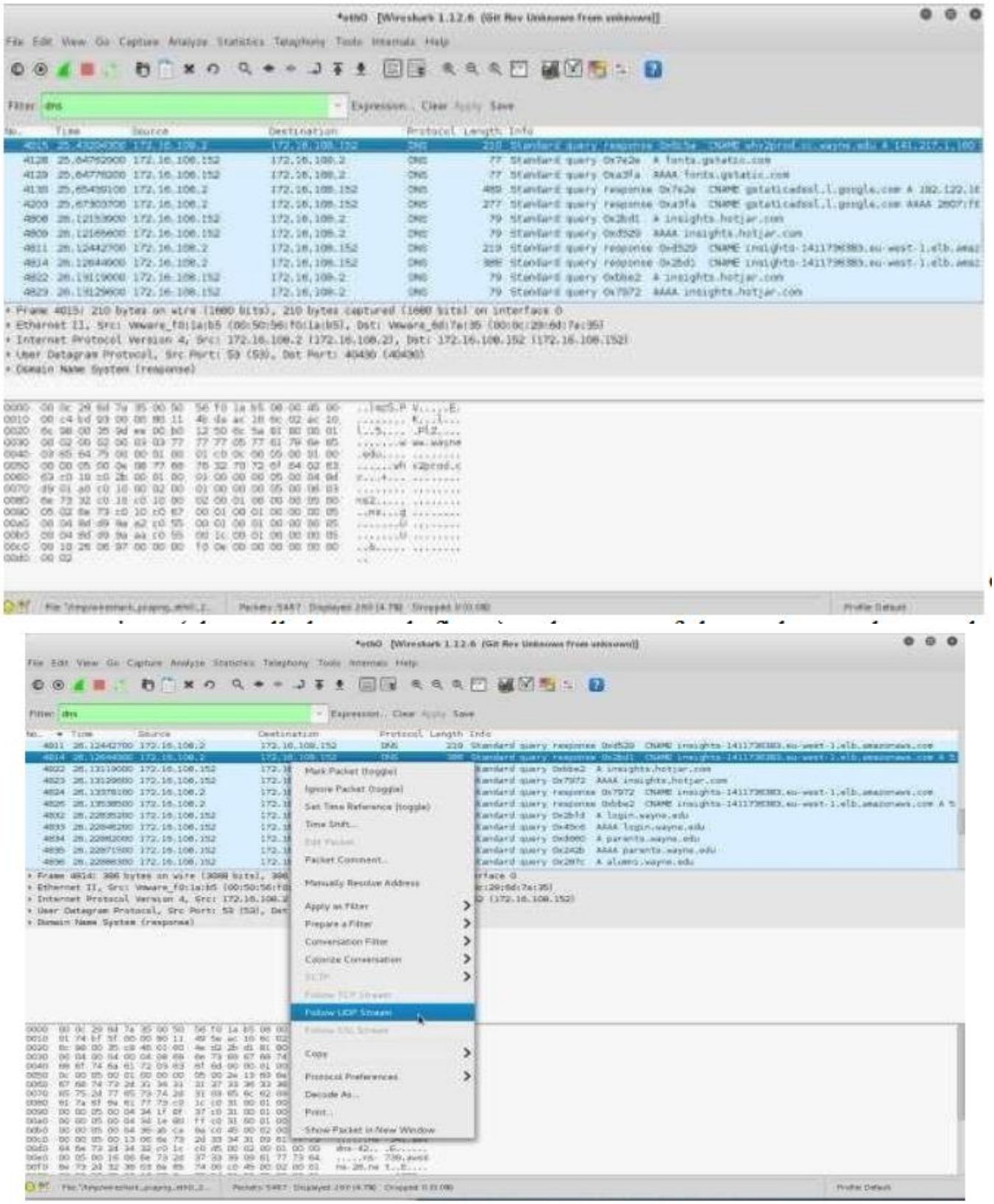
5.Color Coding: You'll probably see packets highlighted in green, blue, and black. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

6. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! However, as you will notice the HTTP messages are not clearly shown because there are many other packets included in the packet capture. Even though the only action you took was to open your browser, there are many other programs in your computer that communicate via the network in the 24 background. To filter the connections to the ones we want to focus on, we have to use the filtering functionality of Wireshark by typing “http” in the filtering field as shown below:
Notice that we now view only the packets that are of protocol HTTP. However, we also still do not have the exact communication we want to focus on because using HTTP as a filter is not descriptive enough to allow us to find our connection to http://www.wayne.edu. We need to be more precise if we want to capture the correct set of packets.

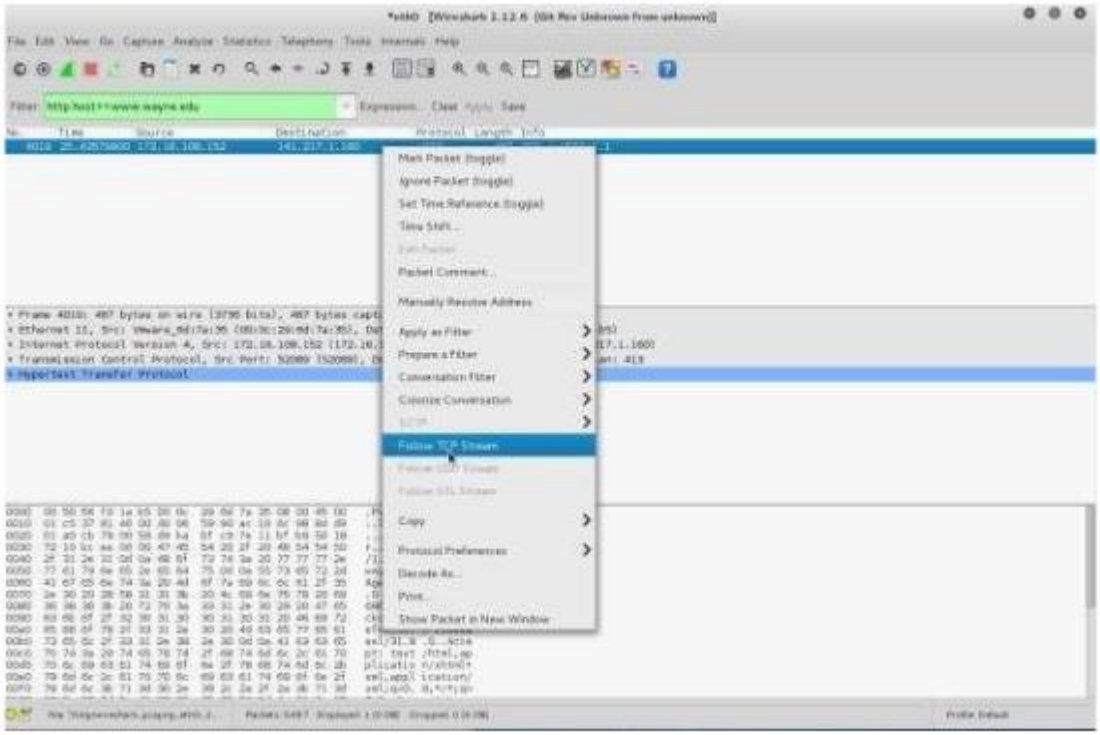
7. To further filter packets in Wireshark, we need to use a more precise filter. By setting the http.host www.wayne.edu, we are restricting the view to packets that have as an http host the www.wayne.edu website. Notice that we need two equal signs to perform the match not just one. See the screenshot below:



8. Now, we can try another protocol. Let’s use Domain Name System (DNS) protocol as an example here:



Click on Follow UDP Stream, and then you will see following screen.



11. If we close this window and change the filter back to “http.hos ww.wayne.edu” and then follow a Packet from the list of packets that match that filter, we should get the something similar to the following screens. Note that we click on Follow TCP Stream this time.

RESULT:

Installation of Wire shark, tcpdump and observe data transferred in client-server communication using UDP/TCP and identify the UDP/TCP datagram.

VIVA QUESTIONS :

1. What are Wireshark and tcpdump, and how do they differ in terms of functionality?
2. Can you explain the process of installing Wireshark on Windows and tcpdump on Unix-like operating systems?
3. How do you initiate packet capture in Wireshark?
4. Can you describe a scenario where you would use tcpdump to capture network traffic on a Unix-like system?
5. What are UDP and TCP?
6. How can you filter captured packets to display only UDP or TCP traffic in Wireshark
7. How can Wireshark or tcpdump be used to identify security threats?

EX.No:5

OBJECTIVE :

The main objective of this program is to demonstrates how to check message integrity and confidentiality using SSL/TLS.

ALGORITHM :

1. The client initiates a secure connection to the server using SSL/TLS protocol.
2. SSL/TLS ensures that the data exchanged between client and server is encrypted, maintaining confidentiality.
3. Before transmission, a Message Authentication Code (MAC) is generated for each message.
4. The client sends the encrypted message along with the MAC to the server over the secure SSL/TLS connection.
5. The server receives the encrypted message and the MAC from the client.
6. If the message integrity is confirmed, the server processes the decrypted message as needed.
7. Both client and server close the SSL/TLS connection once the communication is complete.

PROGRAM :

```
import javax.net.ssl.*;
import java.io.*;
import java.security.*;

public class SSLClient {

   public static void main(String[] args) {
      String serverHost = "localhost";
      int serverPort = 1234;

      try {
         // Create a trust manager that accepts all certificates
         TrustManager[] trustAllCerts = new TrustManager[]{
               new X509TrustManager() {
                  public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                     return null;
                  }
                  public void checkClientTrusted(
                        java.security.cert.X509Certificate[] certs, String authType) {
                  }
                  public void checkServerTrusted(
```

```
                    java.security.cert.X509Certificate[] certs, String authType) {
                }
            }
        };

        // Create SSL context
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, trustAllCerts, new SecureRandom());

        // Create socket factory with the SSL context
        SSLSocketFactory sslSocketFactory = sslContext.getSocketFactory();

        // Connect to the server
        SSLSocket sslSocket = (SSLSocket) sslSocketFactory.createSocket(serverHost, serverPort);
        System.out.println("Connected to " + sslSocket.getRemoteSocketAddress());

        // Create streams for communication
        PrintWriter out = new PrintWriter(sslSocket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(sslSocket.getInputStream()));

        // Send a message to the server
        String message = "Hello, server!";
        out.println(message);
        System.out.println("Sent: " + message);

        // Receive a response from the server
        String response = in.readLine();
        System.out.println("Received: " + response);

        // Close the streams and the socket
        out.close();
        in.close();
        sslSocket.close();
    } catch (IOException | NoSuchAlgorithmException | KeyManagementException e) {
        e.printStackTrace();
    }
  }
}
```

OUTPUT :

```
Connected to localhost/127.0.0.1:1234
Sent: Hello, server!
Received: Message received: Hello, server!
```

RESULT:

Thus the confidentiality and Integrity using SSL was verified

VIVA QUESTION:

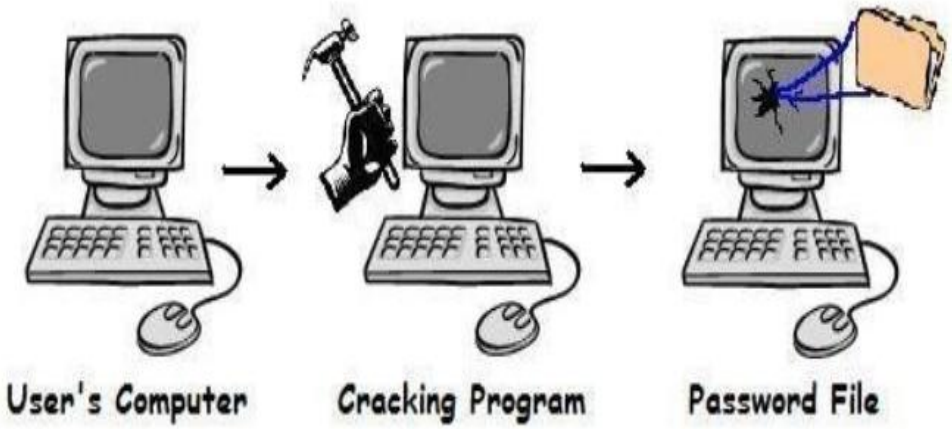
1. What is confidentiality in SSL?
2. What is SSL handshake protocol?
3. How SSL protocol is used for secure transaction?
4. What are the different versions of SSL/TLS?
5. What is the role of Certificate Authorities (CAs) in SSL?
6. How is SSL configured on a web server?
7. Can you describe the process of session key negotiation in SSL?

EX.No:6

OBJECTIVE :

To experiment eavesdropping, Dictionary attacks, MIMT attacks

VISUAL OBJECTS :



INTRODUCTION :
Password cracking is a term used to describe the penetration of a network, system, or resource with or without the use of tools to unlock a resource that has been secured with a password. Password cracking tools may seem like powerful decryptors, but in reality are little more than fast, sophisticated guessing machines.

Types of password breaking
Dictionary attack
A simple dictionary attack is usually the fastest way to break into a machine. A dictionary file (a text file full of dictionary words) is loaded into a cracking application, which is run against user accounts located by the application.

Brute force attack
A brute force attack is a very powerful form of attack, though it may often take a long time to work depending on the complexity of the password. The program will begin trying any and every combination of numbers and letters and running them against the hashed passwords.

Hybrid attack
Another well-known form of attack is the hybrid attack. A hybrid attack will add numbers or symbols to the search words to successfully crack a password. Many people change their passwords by simply adding a number to the end of their current password. Therefore, this type of attack is the most versatile, while it takes longer then a standard dictionary attack it does not take as long as a brute force attack.

Cracking Process
Since a brute force attack is the most time consuming and is not likely to break any passwords that are not composed of random characters, the best plan is to use techniques that are computationally efficient compared to untargeted and unspecific techniques. By applying what is known about how users select passwords, an intruder can tremendously increase the odds in their favor of finding passwords. With the right techniques, some poor passwords can be cracked in under a second.

The real power of dictionary attacks come from understanding the ways in which most people vary names and dictionary words when attempting to create a password. By applying all the common transformations to every word in the electronic list and encrypting each result the number tested passwords multiplies rapidly. Cracking tools can often detect "clever" ways of manipulating words to hide their origin. For example, such cracking programs often subject each word to a list of rules. A rule could be anything, any manner in which a word might appear.

Typical rules might include

- Alternate upper- and lowercase lettering.
- Spell the word forward and then backward, and then fuse the two results (for example: cannac).
- Add the number 1 to the beginning and/or end of each word.

Naturally, the more rules one applies to the words, the longer the cracking process takes. However, more rules also guarantee a higher likelihood of success.

Task 1 – Microsoft Office Password Recovery
Many applications require you to establish an ID and password that may be saved and automatically substituted for future authentication. The password will usually appear on the screen as a series of asterisks. This is fine as long as your system remembers the password for you but what if it "forgets" or you need it for use on another system. Fortunately, many utilities have been written to recover such passwords. In this task, you will use OfficeKey to recover the password for a MS word document.

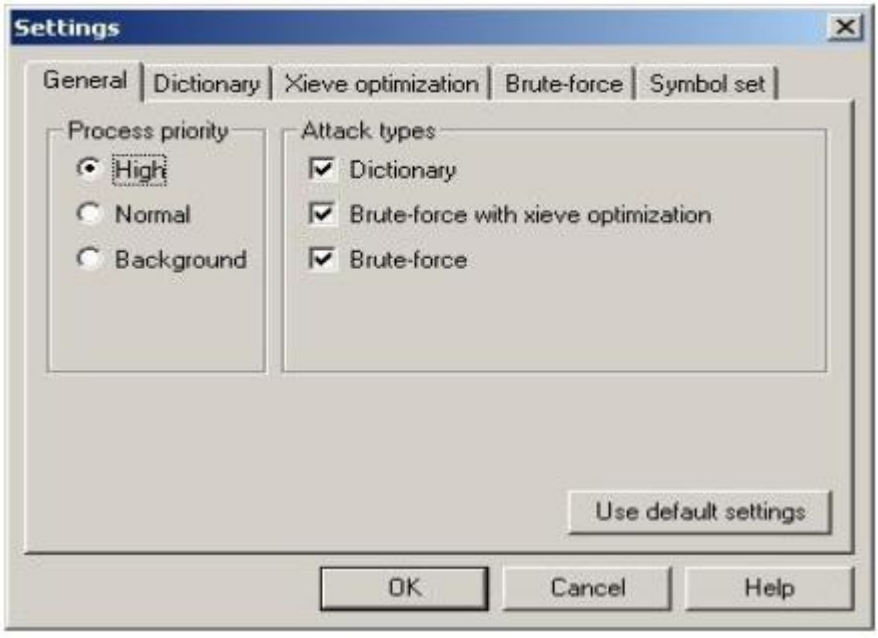
Step 1: Find the folder "Lab1" on your desktop, and open it. You will find OfficeKey and a MS document in the folder.

Step 2: Open the Office Key – Password Recovery tool

Step 3: Press the "Recover" button in the upper left corner, or select File Recover

Step 4: Choose the password protected MS Office File you have saved to the Desktop.

Step 5: After running the first password auditing session, check to see if Office key has cracked the password. If the password has not been cracked press the Settings button on the upper tool bar



Step 6: Once in the Settings menu you will be able to modify the search parameters and customize a more targeted search.

Step 7: Repeat steps 3 and 4 until the password has been cracked and opens the MS Office File.

Step 8: Write down the contents of the MS word document and the password into your lab report and submit it to your TA.

Task 2 – Password Auditing (Windows platform):

The purpose of this task is to familiarize you with act of password cracking/recovery. Password cracking software uses a variety of approaches, including intelligent guessing, dictionary attacks and automation that tries every possible combination of characters. Given enough time the automated method can crack any password, but more effective passwords will last months before breaking.

When a password is entered and saved on a computer it is encrypted, the encrypted password becomes a string of characters called a “hash” and is saved to a password file. A password cannot be reverse-decrypted. So a cracking program encrypts words and characters given to it (wordlist or randomly generated strings of characters) and compares the results with hashed passwords. If the hashes match then the password has successfully been guessed or “cracked”.

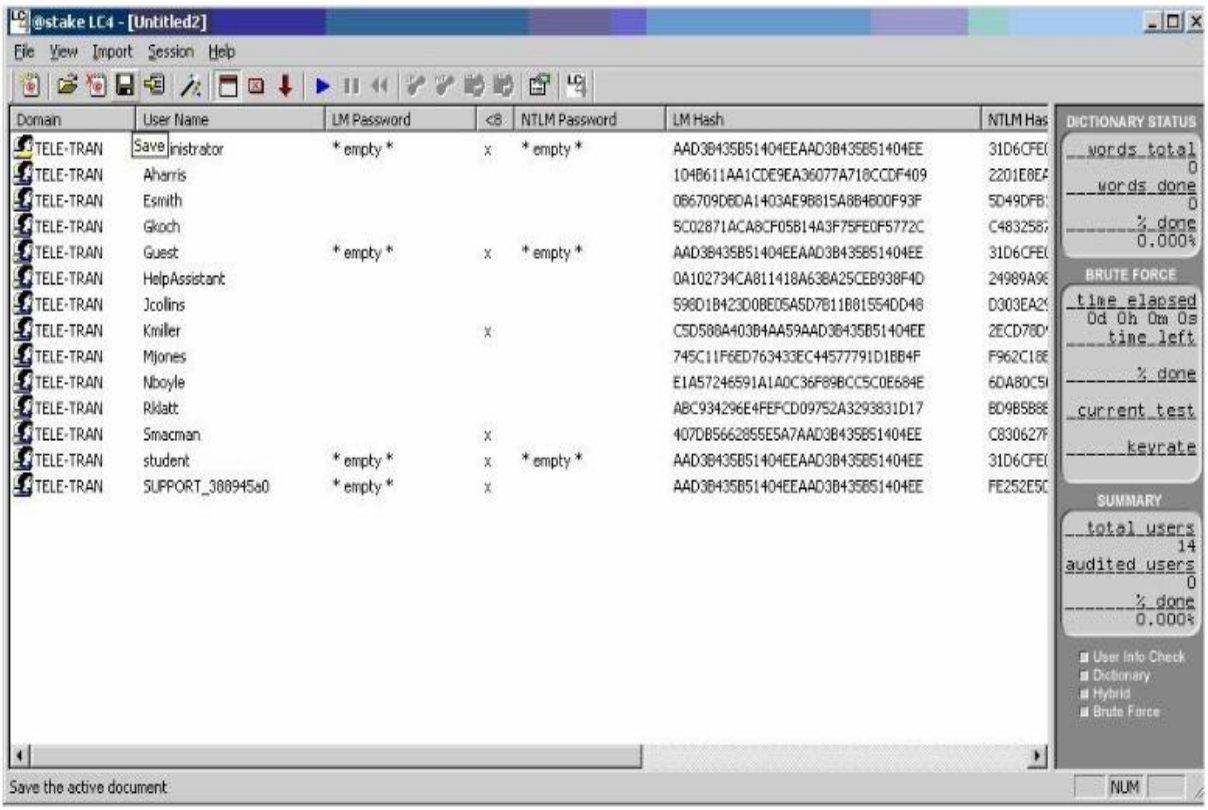
This process is usually performed offline against a captured password file so that being locked

out of the account is not an issue, and guessing can go on continuously. Thus, revealing the

passwords is simply a mater of CPU time and dictionary size

1. You obtain a dictionary file, which is no more than a flat file (plain text) list of words (commonly referred to as wordlists).
2. These words are fed through any number of programs that encrypt each word. Such encryption conforms to the DES standard.
3. Each resulting encrypted word is compared with the target password. If a match occurs, there is better than a 90 percent chance that the password was cracked.

Step 1: Go to Lab1 folder, and open LC4 to audit the passwords on your Windows system.
Select File New Session
Select Import Import from PWDUMP File (in the same folder)
Select the "Passwords" file that has been provided to you.



Objectives

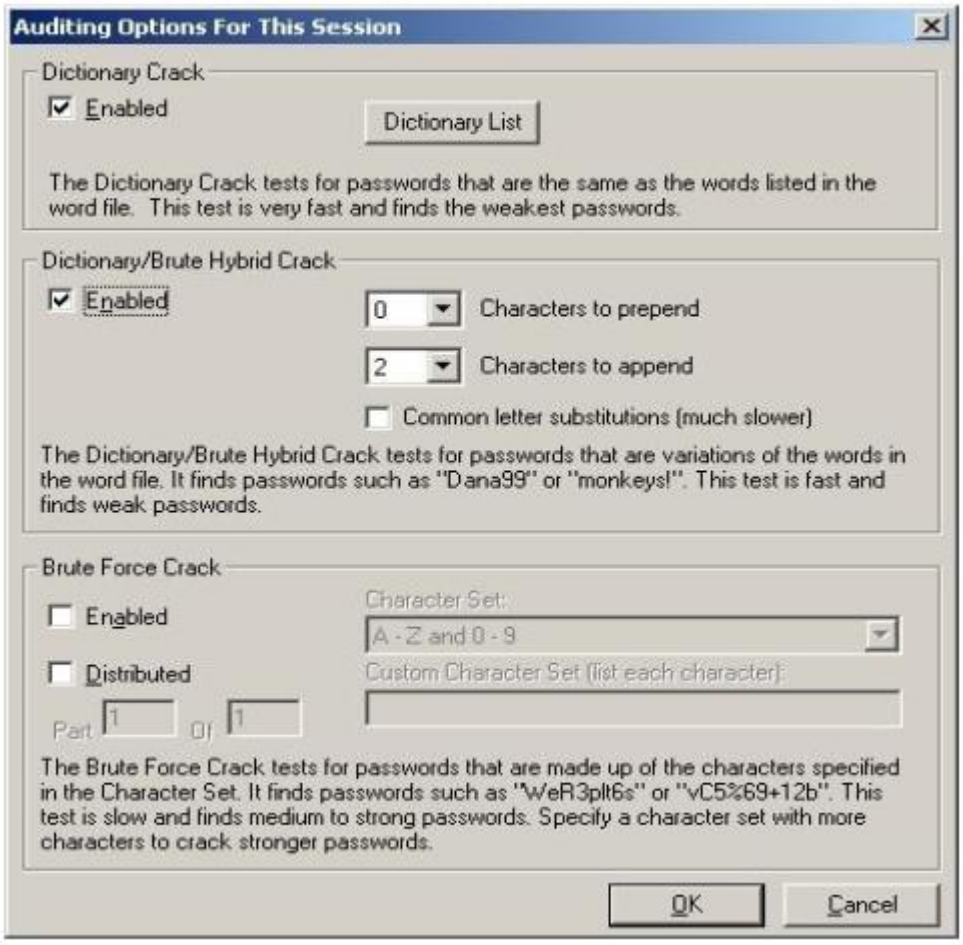
This password file has been retrieved from a system that we must gain access to. To do this you must crack as many passwords as possible as quickly as possible. We have captured the user names and encrypted passwords for ten users. The user names follow a standard pattern of first initial and last name, but the passwords have no set standards. We do know that users of this system are encouraged to add numbers and other characters to the words they chose for passwords.

To aid you in cracking these passwords we have managed to collect some basic information about the users. This personal information may help you target your searches as to what the user's password may be.

Kmiller	Ken Miller is an avid fly fisher and his record number of catches is just under 30
Smacman	Steven MacMan has a fiancé who's name is 4 letters long and starts with a "K"
Gkoch	Gina Koch grew up with her German grandmother, who used to call her 'Little Precious' *
Mjones	Matt Jones was born in 1979. He compares himself to a Shakespearean character who was born via C section
Tgriffin	Tim Griffin loves funky '70's and '80s music. And songs about 'Love'
Rklatt	Ryan Klatt is a big Star Trek fan and has most likely chosen an obscure reference for his password *
Nboyle	Nboyle Nancy Boyle is an a fan of the books of British writer Douglas Adams
Esmith	Edward Smith was very close to his grandfather who died in 1968. We know his grandfather's name was a less common name starting with 'L'
Jcollins	Jim Collins keeps a copy of the book "The Prince" *
Hharris	Alan Harris has a wife named Sue and a daughter named Megan Alan was married on May 3rd. His daughter was born on August 6th

Step 2: Select Session Session Options

Use this menu to customize your password search. Here you can add different word list for Dictionary attacks, change Hybrid attack features. Keep in mind you are working with a short dead line and more in depth searches will take longer then you have. You must use the information given to you to target your search most specifically at more likely passwords.



Step 3: Select Session Begin "Audit" or Press the blue play button on the upper toolbar to start the password search.

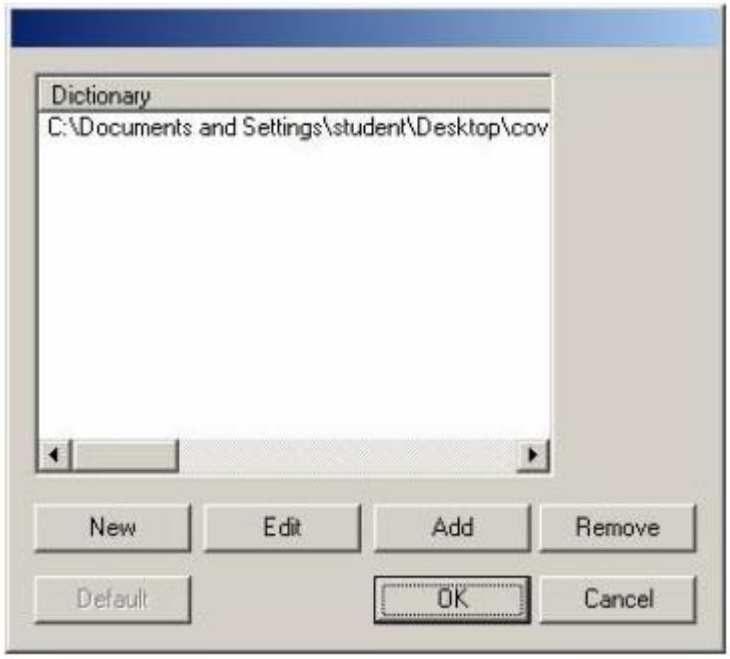
Step 4: After the first search has run check your progress. Have some of the passwords been cracked all the way though or have some only been partially cracked. Use what you've learned from this first search to target your next few searches. You will need to search the internet and use the information you have been given about each user to find words they may have used as their password.

Note: The question marks in the partially cracked passwords do not necessarily represent the number of remaining undiscovered characters.

Step 5: Add words to your wordlist
Session Session Options

Press the 'Dictionary List' button in the Dictionary crack section. Here you can edit your current word list and add words by selecting the 'EDIT' button and entering each wordon a new line. You can also add multiple dictionaries and wordlist.

Step 6: You may chose to conduct dictionary attacks with other wordlists. You can find additional wordlist to use here: ftp://ftp.cerias.purdue.edu/pub/dict



Step 7: Continue searching for possible passwords during the remainder of the lab. Repeating steps 3 and 4 each time you modify your search.

Step 8: Once you have cracked all the passwords in the file, write them down in your lab reportor once the lab time has ended, submit the passwords you were able to crack.

RESULT :

Thus the experiment for Eavesdropping, Dictionary attacks, MITM attacks was done successfully.

VIVA QUESTION :

1. What is eavesdropping in the context of cybersecurity?
2. Can you explain how eavesdropping attacks work?
3. Define what a dictionary attack is and how it differs from brute force attacks.
4. Explain the concept of a man-in-the-middle (MITM) attack and how it undermines the confidentiality and integrity of communication.
5. How can techniques such as SSL/TLS and certificate pinning help mitigate the risk of MITM attacks?
6. Which tool is used for the MITM attack?
7. What types of attacks fall under MITM?

EX.No:7

OBJECTIVE :

To perform an Experiment to Sniff Traffic using ARP Poisoning.

DESCRIPTION:

ARP is the acronym for Address Resolution Protocol. It is used to convert IP address to physical addresses [MAC address] on a switch. The host sends an ARP broadcast on the network, and the recipient computer responds with its physical address [MAC Address]. The resolved IP/MACaddress is then used to communicate. ARP poisoning is sending fake MAC addresses to the switch so that it can associate the fake MAC addresses with the IP address of a genuine computer on a network and hijack the traffic.

ARP Poisoning Countermeasures

Static ARP entries: these can be defined in the local ARP cache and the switch configured to ignoreall auto ARP reply packets. The disadvantage of this method is, it's difficult to maintain on large networks. IP/MAC address mapping has to be distributed to all the computers on the network. ARP poisoning detection software: these systems can be used to cross check the IP/MAC address resolution and certify them if they are authenticated. Uncertified IP/MAC address resolutions can then be blocked.

Operating System Security: this measure is dependent on the operating system been used. The following are the basic techniques used by various operating systems.

Linux based: these work by ignoring unsolicited ARP reply packets.

Microsoft Windows: the ARP cache behavior can be configured via the registry. The following list includes some of the software that can be used to protect networks against sniffing;

AntiARP– provides protection against both passive and active sniffing

Agnitum Outpost Firewall–provides protection against passive sniffing

XArp– provides protection against both passive and active sniffing

Mac OS: ArpGuard can be used to provide protection. It protects against both active andpassive sniffing.

Computers communicate using networks. These networks could be on a local area network LAN or exposed to the internet. Network Sniffers are programs that capture low-level package data that is transmitted over a network. An attacker can analyze this informationto discover valuable information such as user ids and passwords.

In this article, we will introduce you to common network sniffing techniques and tools used to sniff networks

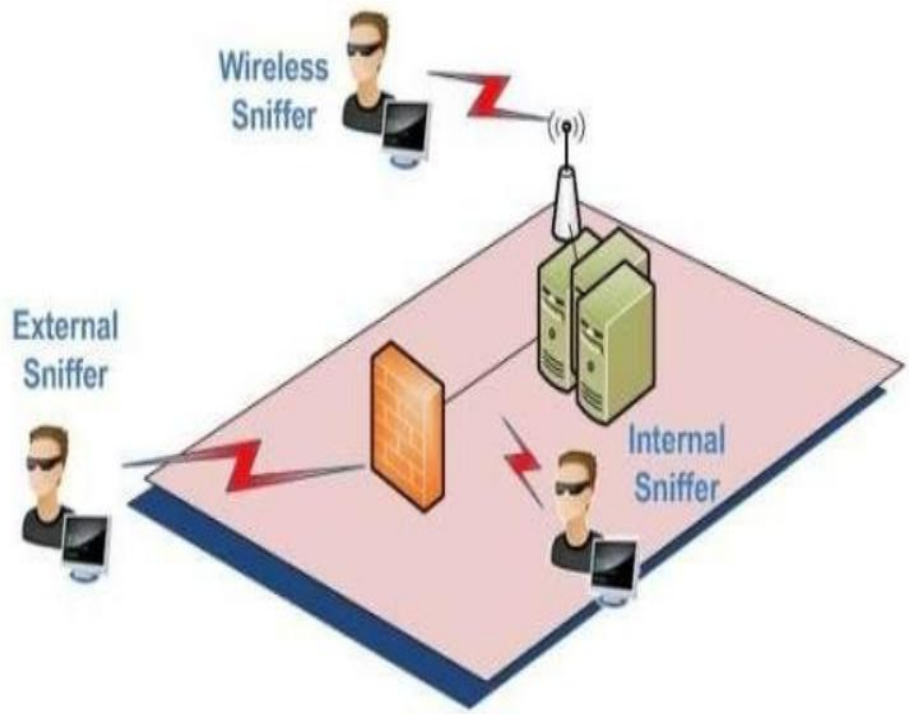
What is network sniffing?

Computers communicate by broadcasting messages on a network using IP addresses. Once a message has been sent on a network, the recipient computer with the matching IP address responds with its MAC address.

Network sniffing is the process of intercepting data packets sent over a network. This can be done by the specialized software program or hardware equipment. Sniffing can be used to;

- Capture sensitive data such as login credentials
- Eavesdrop on chat messages
- Capture files have been transmitted over a network. The following are protocols that
- are vulnerable to sniffing
- Telnet
- Rlogin
- HTTP
- SMTP
- NNTP
- POP
- FTP
- IMAP

The above protocols are vulnerable if login details are sent in plain text

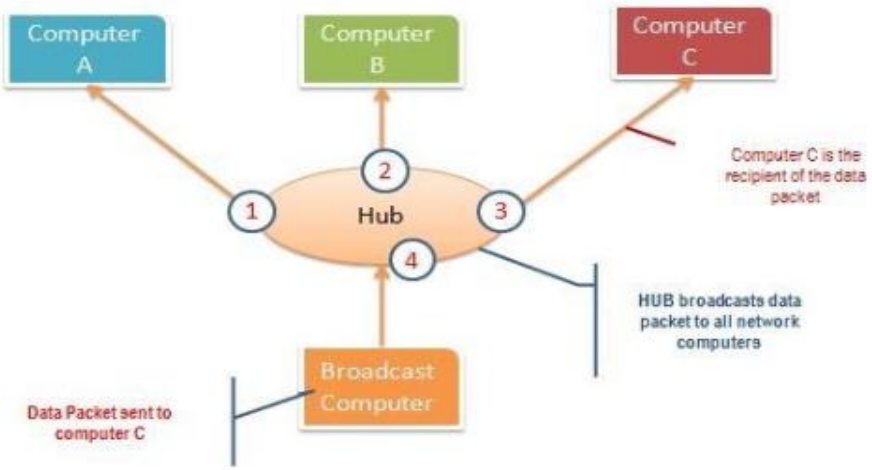


Passive and Active Sniffing

Before we look at passive and active sniffing, let's look at two major devices used to network computers; hubs and switches.

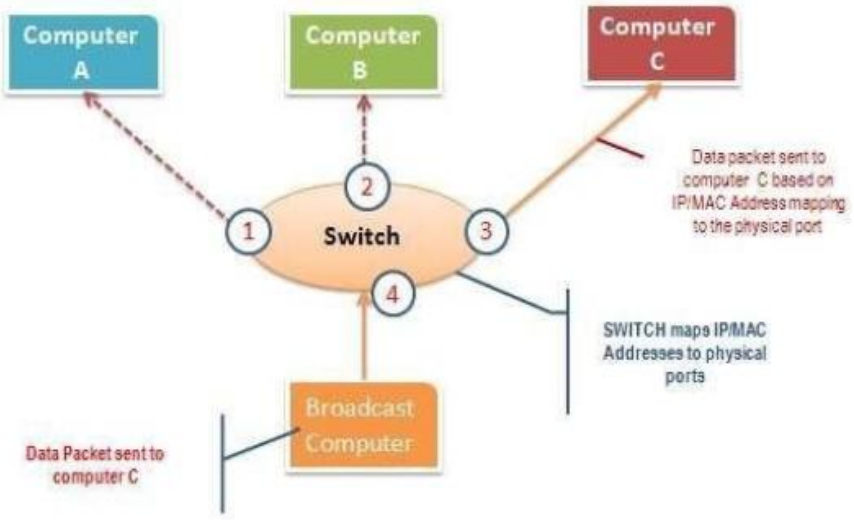
A hub works by sending broadcast messages to all output ports on it except the one that has sent the broadcast. The recipient computer responds to the broadcast message if the IP address matches. This means when using a hub, all the computers on a network can see the broadcast message. It operates at the physical layer (layer 1) of the OSI Model.

The diagram below illustrates how the hub works.



A switch works differently; it maps IP/MAC addresses to physical ports on it. Broadcast messages are sent to the physical ports that match the IP/MAC address configurations for the recipient computer. This means broadcast messages are only seen by the recipient computer. Switches operate at the data link layer (layer 2) and network layer (layer 3).

The diagram below illustrates how the switch works.



Passive sniffing is intercepting packages transmitted over a network that uses a hub. It is Called passive sniffing because it is difficult to detect. It is also easy to perform as the hub sends broadcast messages to all the computers on the network.

Active sniffing is intercepting packages transmitted over a network that uses a switch. There

are two main methods used to sniff switch linked networks, ARP Poisoning, and MAC flooding.

Sniffing the network using Wireshark

The illustration below shows you the steps that you will carry out to complete this exercise without confusion

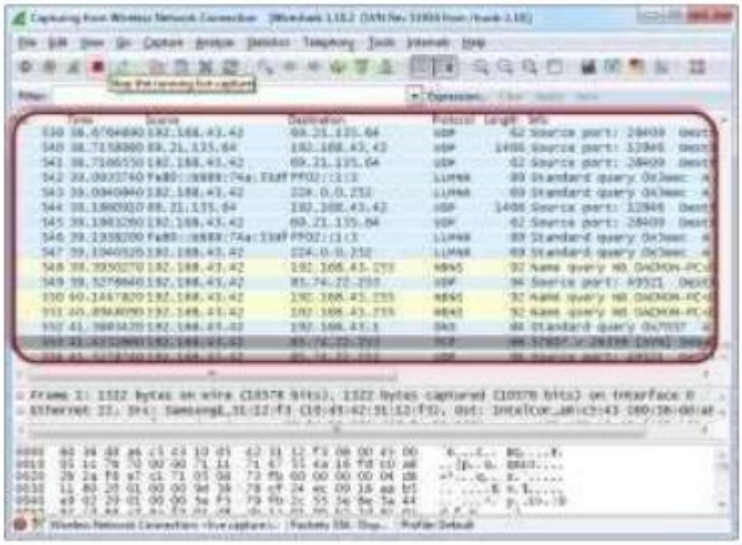
Download Wireshark from this link http://www.wireshark.org/download.html

Open Wireshark

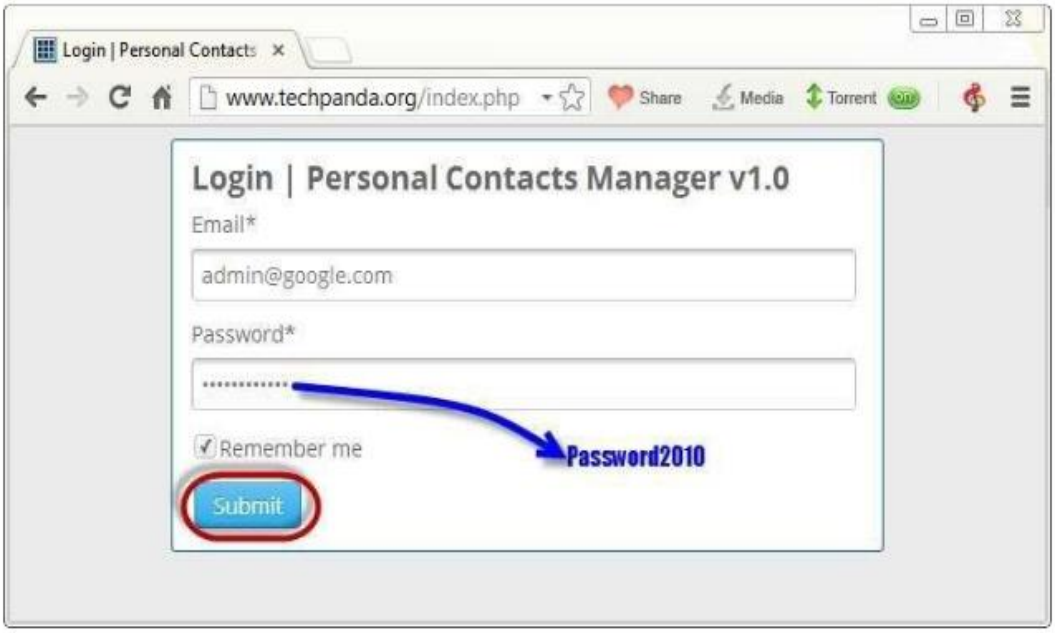
You will get the following screen

Select the network interface you want to sniff. Note for this demonstration, we are using a wireless network connection. If you are on a local area network, then you should select the local area network interface.

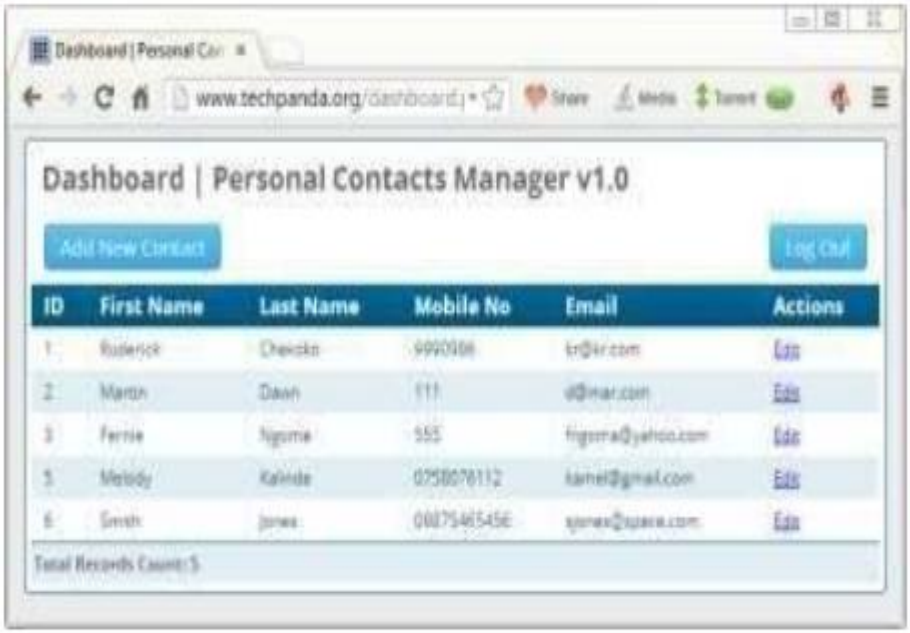
Click on start button as shown above



Open your web browser and type in http://www.techpanda.org/

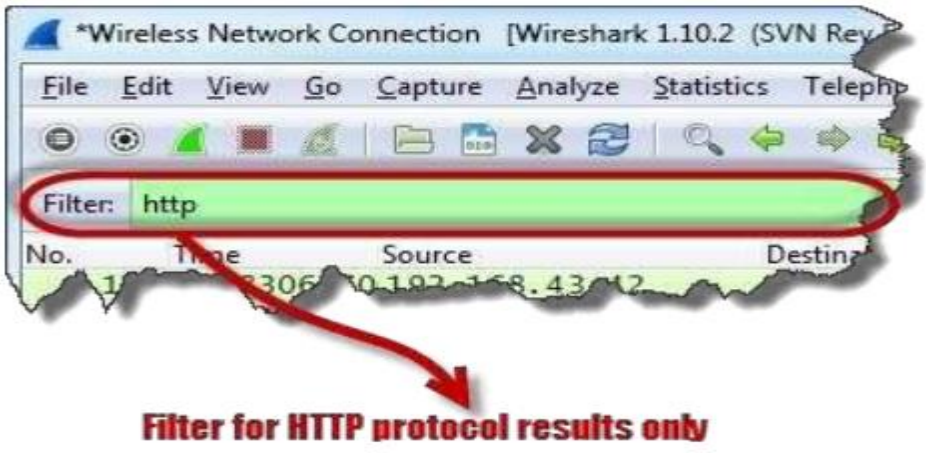


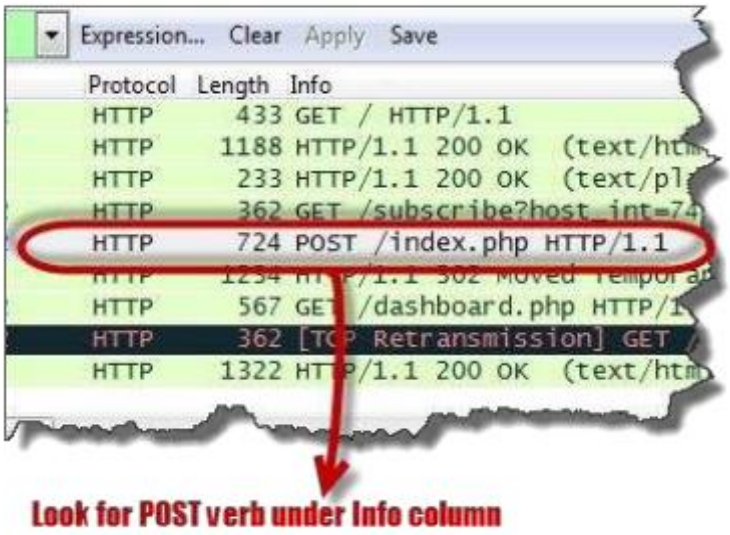
The login email is admin@google.com and the password is Password2010
Click on submit button
A successful logon should give you the following dashboard



Go back to Wireshark and stop the live capture

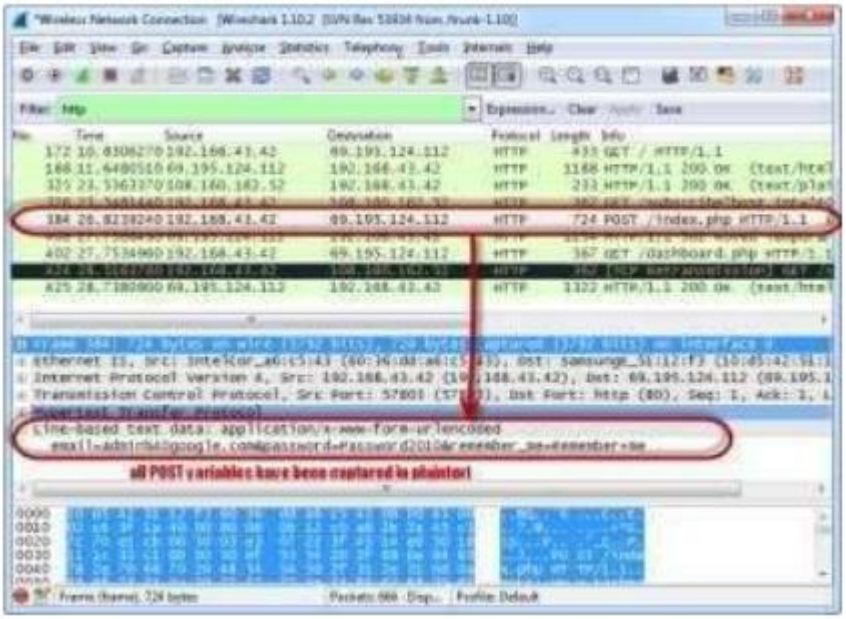
Filter for HTTP protocol results only using the filter textbox





Locate the Info column and look for entries with the HTTP verb POST and click on it

Just below the log entries, there is a panel with a summary of captured data.
Look for the summary that says Line-based text data: application/x-www-form-url encoded



You should be able to view the plaintext values of all the POST variables submitted to the server via HTTP protocol.

RESULT:

Thus the experiment to Sniff Traffic using ARP Poisoning was performed.

VIVA QUESTIONS :

1. What is ARP poisoning for sniffing traffic?
2. Which tool is used to perform ARP poisoning?
3. Why is ARP traffic useful?
4. What does ARP stand for?
5. List the two features of ARP?
6. Name the function of NAT?

EX.No:8

OBJECTIVE :

The main objective is to demonstrate Intrusion Detection System (IDS) using Snort software tool.

STEPS ON CONFIGURING AND INTRUSION DETECTION:

1. Download Snort from the Snort.org website. (http://www.snort.org/snort-downloads)

2. Download Rules(https://www.snort.org/snort-rules). You must register to get the rules. (You should

download these often)

3. Double click on the .exe to install snort. This will install snort in the "C:\Snort" folder.It is

important to have WinPcap (https://www.winpcap.org/install/) installed

4. Extract the Rules file. You will need WinRAR for the .gz file.

5. Copy all files from the "rules" folder of the extracted folder. Now paste the rules into

"C:\Snort\rules" folder.

6. Copy "snort.conf" file from the "etc" folder of the extracted folder. You must paste it into "C:\

Snort\etc" folder. Overwrite any existing file. Remember if you modify your snort.conf file

and download a new file, you must modify it for Snort to work.

7. Open a command prompt (cmd.exe) and navigate to folder "C:\Snort\bin" folder. (at the Prompt, type cd\snort\bin)

8. To start (execute) snort in sniffer mode use following command:

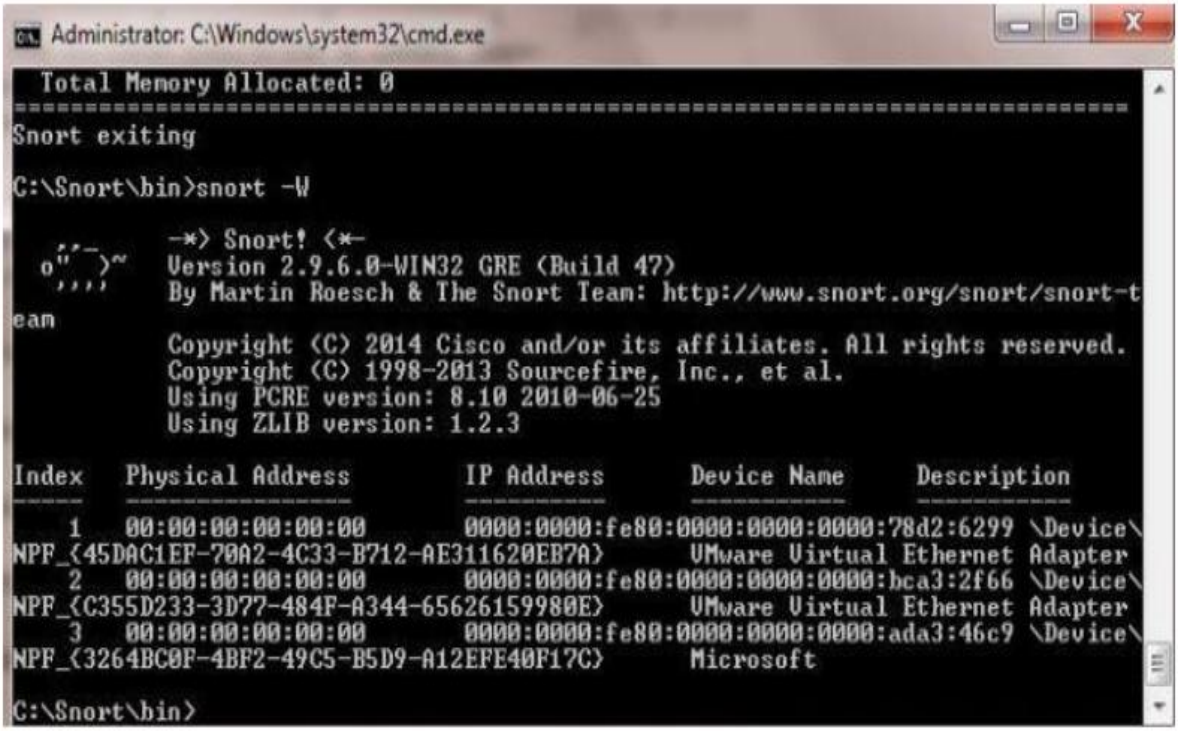
snort -dev -i 3

-i indicates the interface number. You must pick the correct interface number. In my case, it is 3.

-dev is used to run snort to capture packets on your network.

To check the interface list, use following command:

snort -W



Finding an interface

You can tell which interface to use by looking at the Index number and finding Microsoft. As you can see in the above example, the other interfaces are for VMWare.
To run snort in IDS mode, you will need to configure the file "snort.conf" according to your network environment.
To specify the network address that you want to protect in snort.conf file, look for the following line.
var HOME_NET 192.168.1.0/24 (You will normally see any here)
You may also want to set the addresses of DNS_SERVERS, if you have some on your network.

Example:
example snort
Change the RULE_PATH variable to the path of rules folder.
var RULE_PATH c:\snort\rules

path to rules
Change the path of all library files with the name and path on your system. and you must change the path of snort_dynamicpreprocessorvariable.
C:\Snort\lib\snort_dynamiccpreprocessor
You need to do this to all library files in the "C:\Snort\lib" folder. The old path might be: "/usr/local/lib/…". you will need to replace that path with your system path. Using C:\Snort\lib
Change the path of the "dynamicengine" variable value in the "snort.conf" file..

Example:
dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll

Add the paths for “include classification.config” and “include reference.config” files.
include c:\snort\etc\classification.config
include c:\snort\etc\reference.config
Remove the comment (#) on the line to allow ICMP rules, if it is commented with a #.
include $RULE_PATH/icmp.rules
You can also remove the comment of ICMP-info rules comment, if it is
commented. include $RULE_PATH/icmp-info.rules
To add log files to store alerts generated by snort, search for the “output log” test in snort.conf and add the following line:
output alert_fast: snort-alerts.ids
Comment (add a #) the whitelist $WHITE_LIST_PATH/white_list.rules and the blacklist

Change the nested_ip inner , \ to nested_ip inner #, \
Comment out (#) following lines:
#preprocessor normalize_ip4
#preprocessor normalize_tcp: ips ecn stream
#preprocessor normalize_icmp4
#preprocessor normalize_ip6
#preprocessor normalize_icmp6

Save the “snort.conf” file.
To start snort in IDS mode, run the following command:

snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 3
(Note: 3 is used for my interface card)

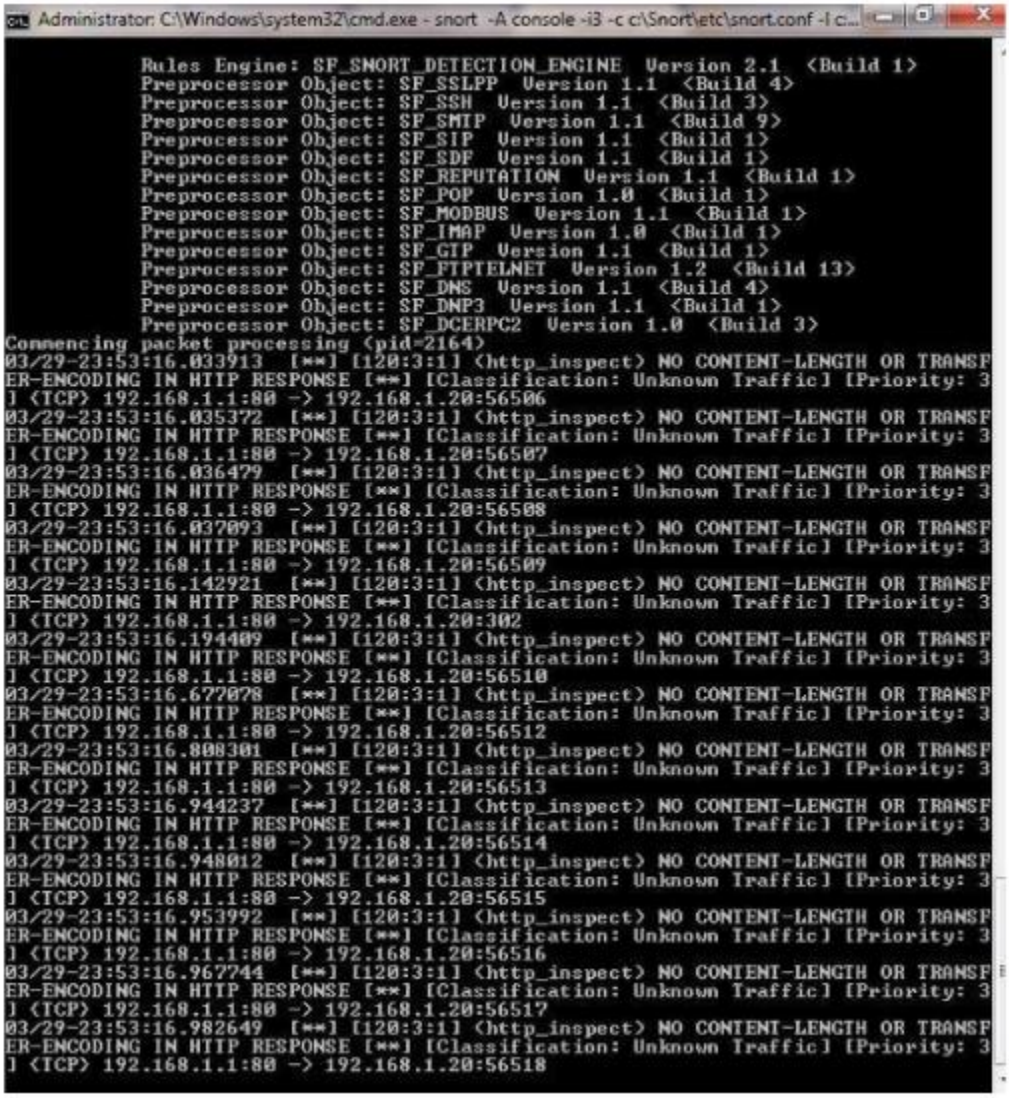
If a log is created, select the appropriate program to open it. You can use WordPard or NotePad++ to read the file.

To generate Log files in ASCII mode, you can use following command while running snort in IDS mode:
snort -A console -i3 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

Scan the computer that is running snort from another computer by using PING or NMap (ZenMap).

After scanning or during the scan you can check the snort-alerts.ids file in the log folder to insure it is logging properly. You will see IP address folders appear.

Snort monitoring traffic –



RESULT:

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

VIVA QUESTIONS :

1. Which tool is used for intrusion detection?
2. Which detection method is used in intrusion detection system?
3. What are the 3 types of intrusion detection systems?
4. What is IDS and IPS tools?
5. How did you set up the IDS using the chosen tool?
6. How does the IDS handle false positives and false negatives?
7. What are types of IPS?

EX.No:9

OBJECTIVE :

The main objective is to explore about Network monitoring tools which is an essential part of network management.

INTRODUCTION :

It involves using various tools to monitor a system network and determine slowness and weak connections, among other issues. Knowing more about these tools can help you understand them better and use the right ones that suit your requirements. In this article, we define what network monitoring tools are, provide details about various tools and discuss about some tips that can help you choose the right tool for your requirements.

What Are Network Monitoring Tools?

Network monitoring tools are software that you can use to evaluate network connections. These software programs can help you monitor a network connection and identify network issues, which may include failing network components, slow connection speed, network outage or unidentifiable connections. Network management and monitoring tools can also help you resolve these issues or establish solutions that prevent specific issues from occurring in the future.

Network Monitoring Tools

Here are eight monitoring tools along with their descriptions and features:

1. SolarWinds Network Performance Monitor

SolarWinds Network Performance Monitor is a multi-vendor monitoring tool. It allows users to monitor multiple vendors' networks at the same time. It also provides network insights for thorough visibility into the health of the networks. Some prominent features include network availability monitoring, intelligent network mapping, critical path visualisation, performance analysis and advanced alerting. SolarWinds also allows users to track VPN tunnel status. It prompts when a VPN tunnel is available to help users ensure a stable connection between sites. SolarWinds provides a seven-day free trial, after which users can choose a preferred subscription plan.

2. Auvik

Auvik is a network monitoring and management tool. It offers a quick implementation process that helps users to set up the tool easily. It also has a clean user interface that makes it easy to navigate and use. The tool provides in-depth network visibility that enables faster troubleshooting for network issues. Users can automate network visibility using Auvik. It provides real-time updates on network issues and configuration changes.

3. Datadog Network Monitoring

Datadog Network Monitoring offers services for on-premises devices and cloud networks. A highlighting feature of this tool is the visualisations. It offers various graphical representations of all the network connections on a system. It also allows users to track key metrics like network latency, connection churn and transmission control protocol (TCP) retransmits. Users can monitor the health of a network connection at different endpoints at the application, IP address, port or process ID layers. Other prominent features include automated log collection and user interface monitoring.

4. Paessler PRTG Network Monitor

Paessler's network connection monitoring tool provides a clean user interface and network visibility on multiple devices. Users can track the health of different connection types like local area networks (LAN), wide area network (WAN), servers, websites, applications and services. The tools also integrate with various technologies, which makes it easier to use it for different types of applications. It provides distribute monitoring, allowing users to track network connections on devices in different locations. The tool also provides apps for mobile platforms that can help users to track network health on mobile phones.

5. ManageEngine OpManager

ManageEngine OpManager is a good network monitoring and managing tool for users that prefer in-depth view of network health and issues. This tool provides over 2000 network performance monitors that allow users to track and monitor their connections and perform detailed analyses on issues. It also provides over 200 dashboard widgets that can help users customise their dashboard to their own suitability. Other features include CPU, memory and disk utilisation monitoring on local and virtual machines. It also allows setting network performance threshold and notifies the user in case of a violation.

6. Domotz

Domotz is an expansive tool that provides a list of features for monitoring network connections. It allows users to customise their network monitoring preferences. Users can write scripts the retrieve the data they wish to evaluate. It also allows connection to open ports on remote devices while ensuring network security. Users can also scan and monitor network connections globally. Domotz also allows to backup and restore network configuration for switches, firewalls and access points and alerts when there is a change in the configuration.

7. Checkmk

Checkmk is a tool that allows users to automate it completely. You can customise its operations and enable it to perform tasks automatically. It also identifies network and security components without the user requiring manual set up. For example, the tool can identify a firewall even if the user has not set it up. Its Agent Bakery feature enables users to manage agents and automate agent updating. This reduces manual effort to monitor network connections. The tool also includes over 2000 plug-ins for enhancing network monitoring.

8. Progress Whatsup Gold

Progress Whatsup Gold is a basic network monitoring software. It provides a minimal user interface with essential features like device monitoring, application monitoring, analysing network traffic and managing configurations. The tool allows users to monitor cloud devices, inspect suspicious connections, automate configuration backups and identify, and resolve bandwidth issues.

Other Tools For Network Monitoring

Here are three additional tools for network monitoring:

Fortra Intermapper: This tool enables users to monitor network connections using network maps, allowing them to get a holistic view of all the connections. It also provides various colour codes for different network status, along with real-time notifications through text, email and sound.

Nagios Core: Nagios Core is a monitoring engine that works as the primary application for all Nagios projects, including the Nagios Network Analyser. It integrates with other Nagios applications and provides users with features like a visual dashboard, custom application monitoring, automated alert system, advanced user management and network security monitoring.

Zabbix: Zabbix provides a thorough network monitoring solution with features like server monitoring, cloud monitoring, application monitoring and service monitoring. The tool also includes features like metric collection, business monitoring and root cause analyses of network issues, and allows users to establish a threshold for connection anomalies.

Tips To Choose A Network Monitoring And Management Tool

Here are some useful tips that you can consider while selecting a tool for network monitoring:

Understand the requirements

Understanding why you require network monitoring software is important in the process. Define what feature you want and for what purpose. This can help you identify the right tool for your use. It may also help you choose the correct subscription plan on paid tools.

Browse multiple tools

Once you identify the requirements, consider browsing multiple tools. Visit the websites of the tools and look for the features you require. Spend time studying the features and understand how they can be useful to your requirements. You can also identify a few tools and compare their features to each other.

Consider the budget

Some tools may be free to use, while some may require you to purchase a subscription plan. Paid tools typically offer a free trial period of up to 30 days. Once you identify which tool you may like to use, see if it is free or requires payment. If it is a paid tool, try exploring its features and efficiency during the trial period. Consider keeping a backup tool in case the tool that you choose does not fit your usage.

RESULT:

Thus the network monitoring tools was explored.

VIVA QUESTIONS :

1. What is network monitoring?
2. Which monitoring tools and approaches have you used in previous jobs?
3. What is a peer-to-peer network?
4. Name some common features found in network monitoring tools?
5. What factors affect the performance of network monitoring tools?
6. Explain the difference between active and passive monitoring.
7. How is the role of SNMP in network monitoring?

EX.No:10

OBJECTIVE :

The main objective is to study the features of firewall in providing network security and to set Firewall Security in windows.

Firewall in Windows 7

Windows 7 comes with two firewalls that work together. One is the Windows Firewall, and the other is Windows Firewall with Advanced Security (WFAS). The main difference between them is the complexity of the rules configuration. Windows Firewall uses simple rules that directly relate to a program or a service. The rules in WFAS can be configured based on protocols, ports, addresses and authentication. By default, both firewalls come with predefined set of rules that allow us to utilize network resources. This includes things like browsing the web, receiving e-mails, etc. Other standard firewall exceptions are File and Printer Sharing, Network Discovery, Performance Logs and Alerts, Remote Administration, Windows Remote Management, Remote Assistance, Remote Desktop, Windows Media Player, Windows Media Player Network Sharing Service

With firewall in Windows 7 we can configure inbound and outbound rules. By default, all outbound traffic is allowed, and inbound responses to that traffic are also allowed. Inbound traffic initiated from external sources is automatically blocked.

When we first connect to some network, we are prompted to select a network location. This feature is known as Network Location Awareness(NLA). This feature enables us to assign a network profile to the connection based on the location. Different network profiles contain different collections of firewall rules. In Windows 7, different network profiles can be configured on different interfaces. For example, our wired interface can have different profile than our wireless interface. There are three different network profiles available:

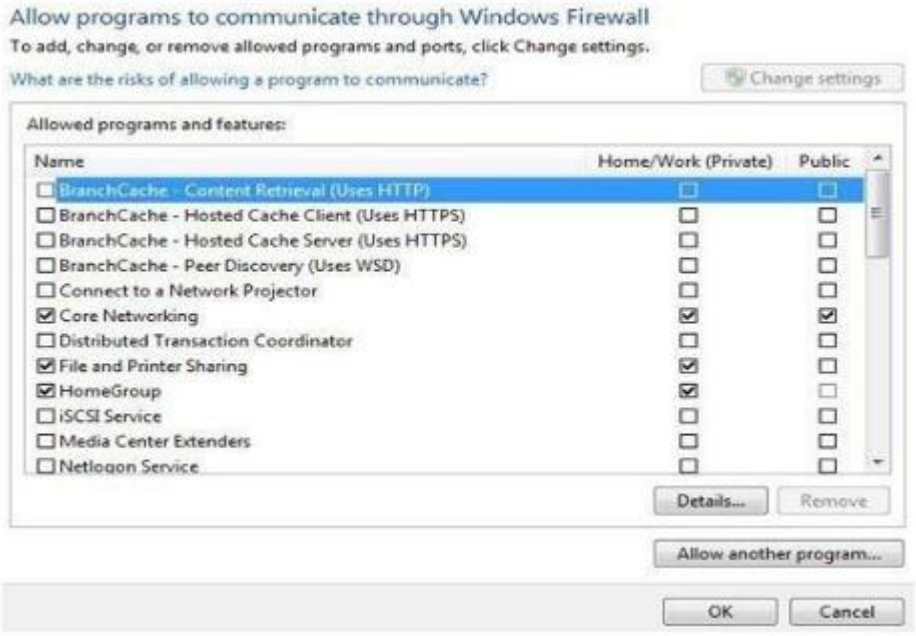
- Public
- Home/Work - private network
- Domain - used within a domain

Configuring Windows Firewall

To open Windows Firewall we can go to Start > Control Panel > Windows

Firewall.

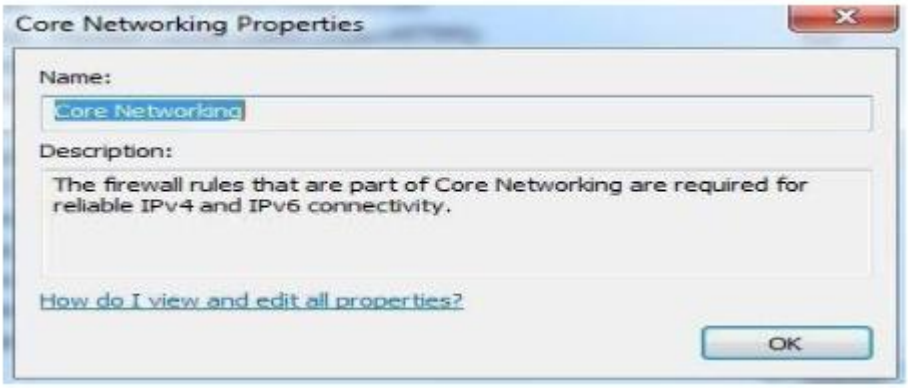
By default, Windows Firewall is enabled for both private (home or work)and public networks. It is also configured to block all connections to programs that are not on the list of allowed programs. To configure exceptions we can go to the menu on the left and select "Allow a program or feature trough Windows Firewall" option.



Exceptions

To change settings in this window we have to click the "Change settings" button. As you can see, here we have a list of predefined programs and features that can be allowed to communicate on private or public networks. For example, notice that the Core Networking feature is allowed on

both private and public networks, while the File and Printer Sharing is only allowed on private networks.
We can also see the details of the items in the list by selecting it and then clicking the Details button.



Details
If we have a program on our computer that is not in this list, we can manually add it by clicking on the "Allow another program" button.

Add a Program
Here we have to browse to the executable of our program and then click the Add button. Notice that we can also choose location types on which this program will be allowed to communicate by clicking on the "Network location types" button.

Network Locations
Many applications will automatically configure proper exceptions in Windows Firewall when we run them. For example, if we enable streaming from Media Player, it will automatically configure firewall settings to allow streaming. The same thing is if we enable Remote Desktop feature from the system properties window. By enabling Remote Desktop feature we actually create an exception in Windows Firewall.

Windows Firewall can be turned off completely. To do that we can select the "Turn Windows Firewall on or off" option from the menu on the left.

Firewall Customization

Note that we can modify settings for each type of network location (private or public). Interesting thing here is that we can block all incoming connections, including those in the list of allowed programs.

Windows Firewall is actually a Windows service. As you know, services can be stopped and started. If the Windows Firewall service is stopped, the Windows Firewall will not work.

Firewall Service

In our case the service is running. If we stop it, we will get a warning that we should turn on our Windows Firewall.

Warning

Remember that with Windows Firewall we can only configure basic firewall settings, and this is enough for most day-to-day users. However, we can't configure exceptions based on ports in Windows Firewall any more. For that we have to use Windows Firewall with Advanced Security.

How to Start & Use the Windows Firewall with Advanced Security

The Windows Firewall with Advanced Security is a tool which gives you detailed control over the rules that are applied by the Windows Firewall. You can view all the rules that are used by the Windows Firewall, change their properties, create new rules or disable existing ones. In this tutorial we will share how to open the Windows Firewall with Advanced Security, how to find your way around it and talk about the types of rules that are available and what kind of traffic they filter.

How to Access the Windows Firewall with Advanced Security

You have several alternatives to opening the Windows Firewall with Advanced Security:
One is to open the standard Windows Firewall window, by going to "Control Panel -> System and Security -> Windows Firewall". Then, click or tap Advanced settings

In Windows 7, another method is to search for the word firewall in the Start Menu search box and click the "Windows Firewall with Advanced Security" result.

In Windows 8.1, Windows Firewall with Advanced Security is not returned in search results and you need to use the first method shared above for opening it.

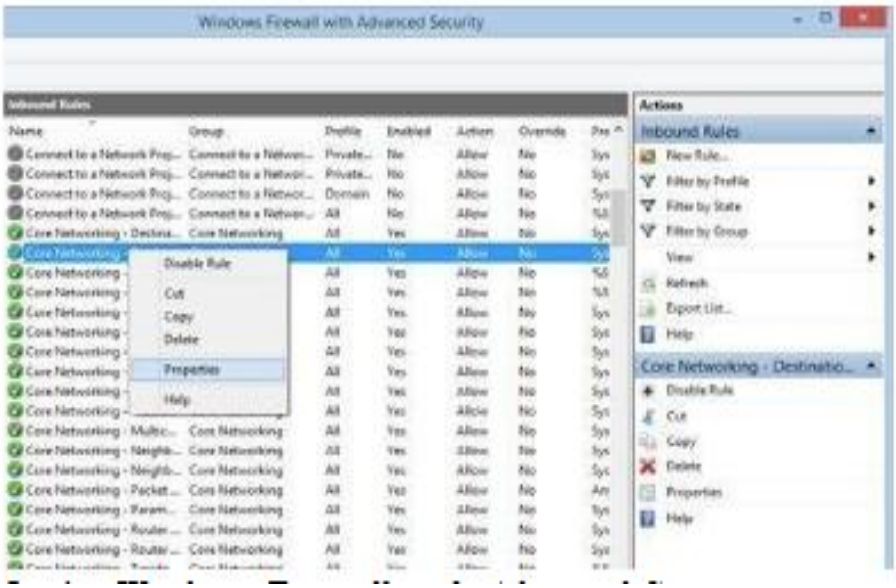
The Windows Firewall with Advanced Security looks and works the same both in Windows 7 and Windows 8.1. To continue our tutorial, we will use screenshots that were made in Windows 8.1.

What Are The Inbound & Outbound Rules?

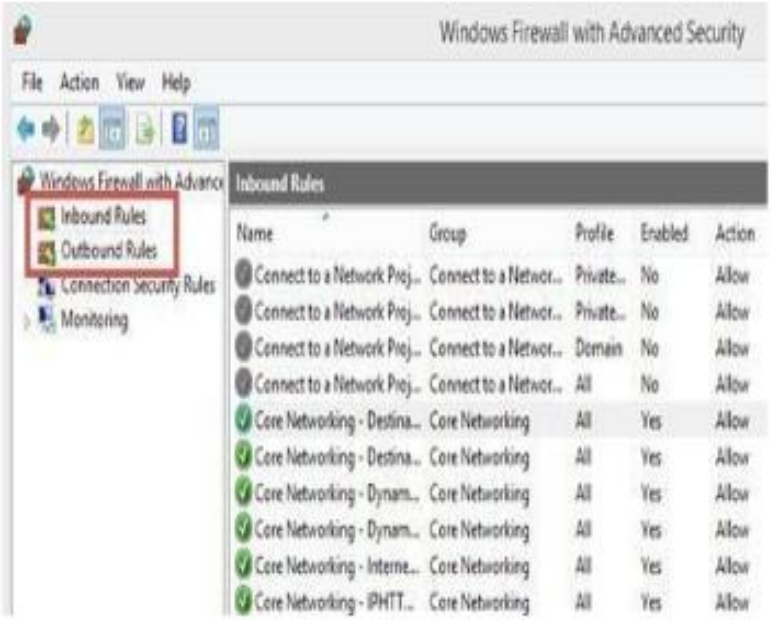
In order to provide the security you need, the Windows Firewall has a standard set of inbound and outbound rules, which are enabled depending on the location of the network you are connected to.

Inbound rules are applied to the traffic that is coming from the network and the Internet to your computer or device. Outbound rules apply to the traffic from your computer to the network or the Internet.

These rules can be configured so that they are specific to: computers, users, programs, services, ports or protocols. You can also specify to which type of network adapter (e.g. wireless, cable, virtual private network) or user profile it is applied to

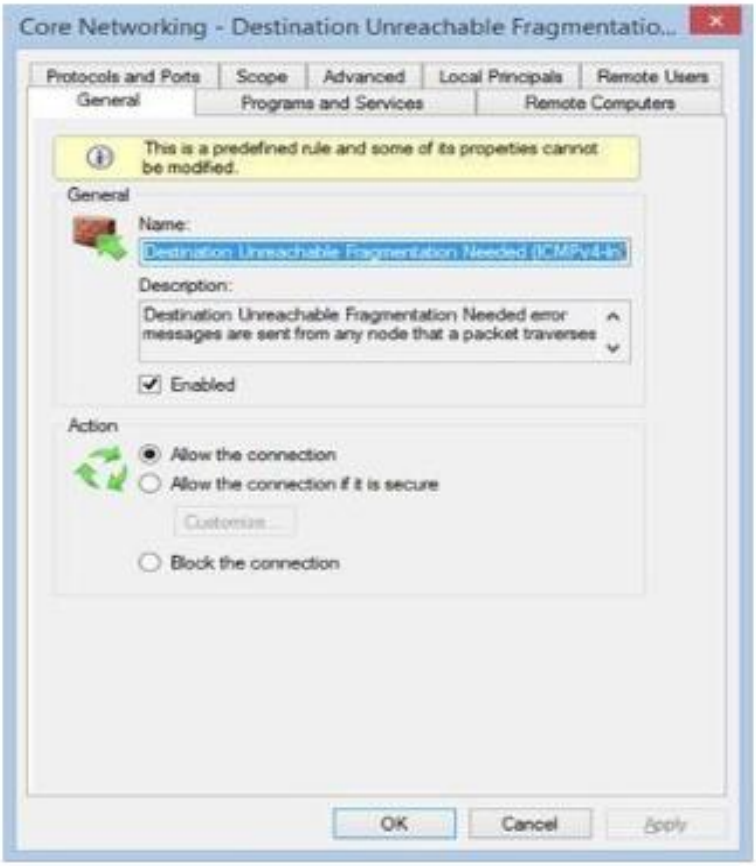


In the Windows Firewall with Advanced Security, you can access all rules and edit their properties. All you have to do is click or tap the appropriate unit in the left-side panel.



The rules used by the Windows Firewall can be enabled or disabled. The ones which are enabled or active are marked with a green check-box in the Name column. The ones that are disabled are marked with a gray check-box.

If you want to know more about a specific rule and learn its properties, right click on it and select Properties or select it and press Properties in the column on right, which lists the actions that are available for your selection.

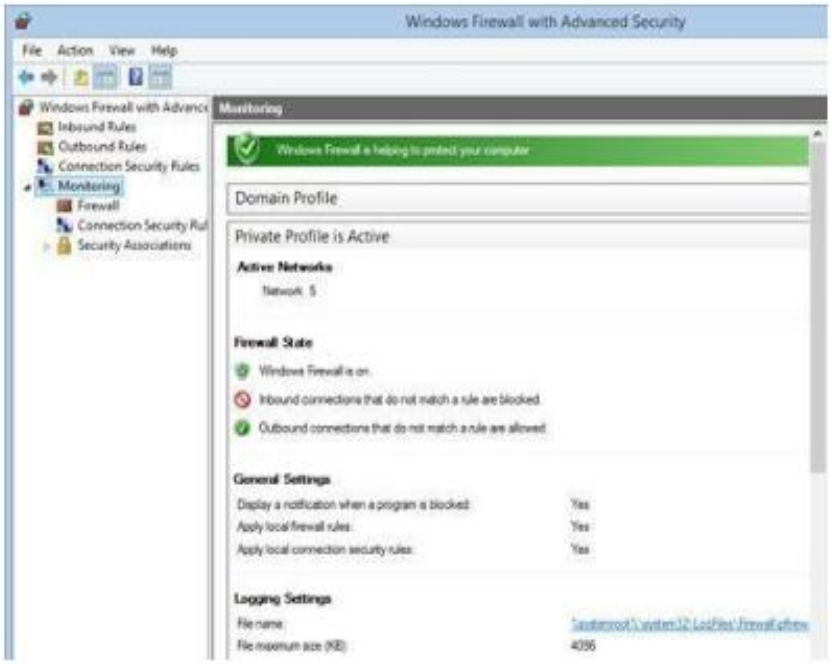


What Are The Connection Security Rules?

Connection security rules are used to secure traffic between two computers while it crosses the network. One example would be a rule which defines that connections between two specific computers must be encrypted.

Unlike the inbound or outbound rules, which are applied only to one computer, connection security rules require that both computers have the same rules defined and enabled.

If you want to see if there are any such rules on your computer, click or tap "Connection Security Rules" on the panel on the left. By default, there are no such rules defined on Windows computers and devices. They are generally used in business environments and such rules are set by the network administrator.



What Does the Windows Firewall with Advanced Security Monitor?

The Windows Firewall with Advanced Security includes some monitoring features as well. In the Monitoring section you can find the following information: the firewall rules that are active (both inbound and outbound),the connection security rules that are active and whether there are any active security associations.

You should note that the Monitoring section shows only the active rules for the current network location.

used to determine the operating system running on the host machine. Another feature is "boot-time filtering". This feature ensures that the firewall is working at the same time when the network interface becomes active, which was not the case in previous versions of Windows.

When we first connect to some network, we are prompted to select a network location. This feature is known as Network Location Awareness (NLA). This feature enables us to assign a network profile to the connection based on the location. Different network profiles contain different collections of firewall rules. In Windows 7, different network profiles can be configured on different interfaces. For example, our wired interface can

have different profile than our wireless interface. There are three different network profiles available:

- Public
- Home/Work - private network
- Domain - used within a domain

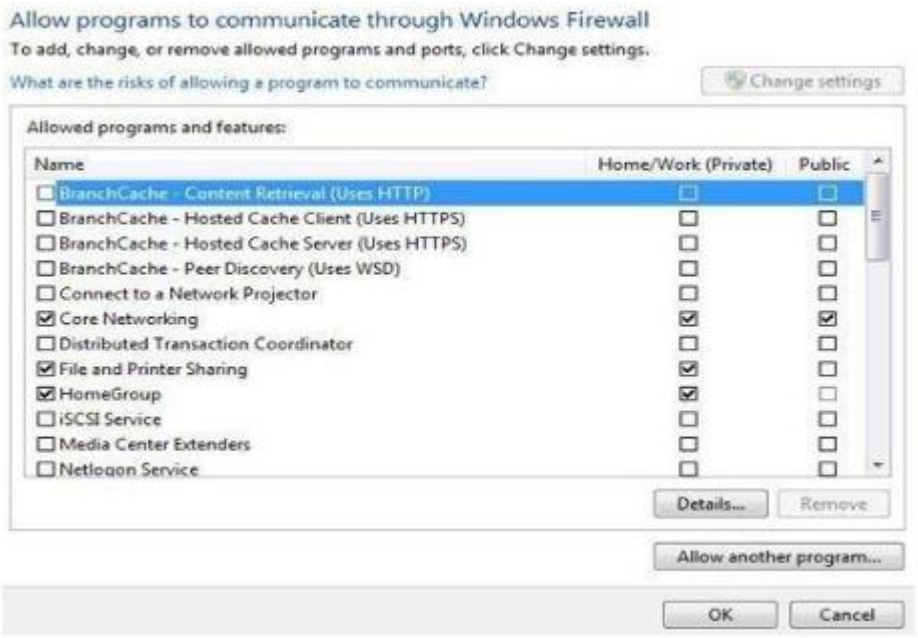
We choose those locations when we connect to a network. We can always change the location in the Network and Sharing Center, in Control Panel. The Domain profile can be automatically assigned by the NLA service when we log on to an Active Directory domain. Note that we must have administrative rights in order to configure firewall in Windows 7.

2.1.1 Configuring Windows Firewall

To open Windows Firewall we can go to **Start > Control Panel >**

Windows Firewall.

By default, Windows Firewall is enabled for both private (home or work) and public networks. It is also configured to block all connections to programs that are not on the list of allowed programs. To configure exceptions we can go to the menu on the left and select "Allow a program or feature trough Windows Firewall" option.



Exceptions

To change settings in this window we have to click the "Change settings" button. As you can see, here we have a list of predefined programs and features that can be allowed to communicate on private or public networks. For example, notice that the Core Networking feature is allowed on both private and public networks, while the File and Printer Sharing is only allowed on private networks. We can also see the details of the items in the list by selecting it and then clicking the Details button.

Details

If we have a program on our computer that is not in this list, we can

manually add it by clicking on the "Allow another program" button.

Add a Program

Here we have to browse to the executable of our program and then click the Add button. Notice that we can also choose location types on which this program will be allowed to communicate by clicking on the "Network location types" button.

Network Locations

Many applications will automatically configure proper exceptions in Windows Firewall when we run them. For example, if we enable streaming from Media Player, it will automatically configure firewall settings to allow streaming. The same thing is if we enable Remote Desktop feature from the system properties window. By enabling Remote Desktop feature we actually create an exception in Windows Firewall.

Windows Firewall can be turned off completely. To do that we can select the "Turn Windows Firewall on or off" option from the menu on the left.

Firewall Customization

Note that we can modify settings for each type of network location (private or public). Interesting thing here is that we can block all incoming connections, including those in the list of allowed programs.

Windows Firewall is actually a Windows service. As you know, services can be stopped and started. If the Windows Firewall service is stopped, the Windows Firewall will not work.

Firewall Service

In our case the service is running. If we stop it, we will get a warning thatwe should turn on our Windows Firewall.

Warning

Remember that with Windows Firewall we can only configure basic firewall settings, and this is enough for most day-to-day users. However, we can't configure exceptions based on

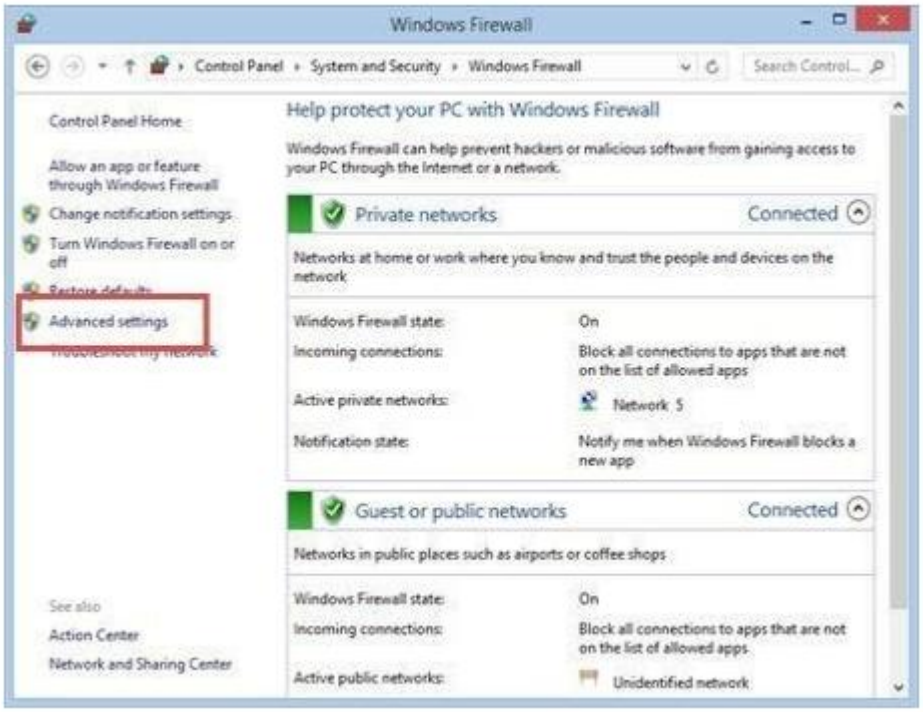
ports in Windows Firewall any more. For that we have to use Windows Firewall with Advanced Security.

How to Start & Use the Windows Firewall with Advanced Security

The Windows Firewall with Advanced Security is a tool which gives you detailed control over the rules that are applied by the Windows Firewall. You can view all the rules that are used by the Windows Firewall, change their properties, create new rules or disable existing ones. In this tutorial we will share how to open the Windows Firewall with Advanced Security, how to find your way around it and talk about the types of rules that are available and what kind of traffic they filter. How to Access the Windows Firewall with Advanced Security

You have several alternatives to opening the Windows Firewall with Advanced Security:

One is to open the standard Windows Firewall window, by going to "Control Panel -> System and Security -> Windows Firewall". Then, click or tap Advanced settings.



In Windows 7, another method is to search for the word firewall in the Start Menu search box and click the "Windows Firewall with Advanced Security" result.

In Windows 8.1, Windows Firewall with Advanced Security is not returned in search results and you need to use the first method shared above foropening it.

The Windows Firewall with Advanced Security looks and works the same both in Windows 7 and Windows 8.1. To continue our tutorial, we will use screenshots that were made in Windows 8.1.

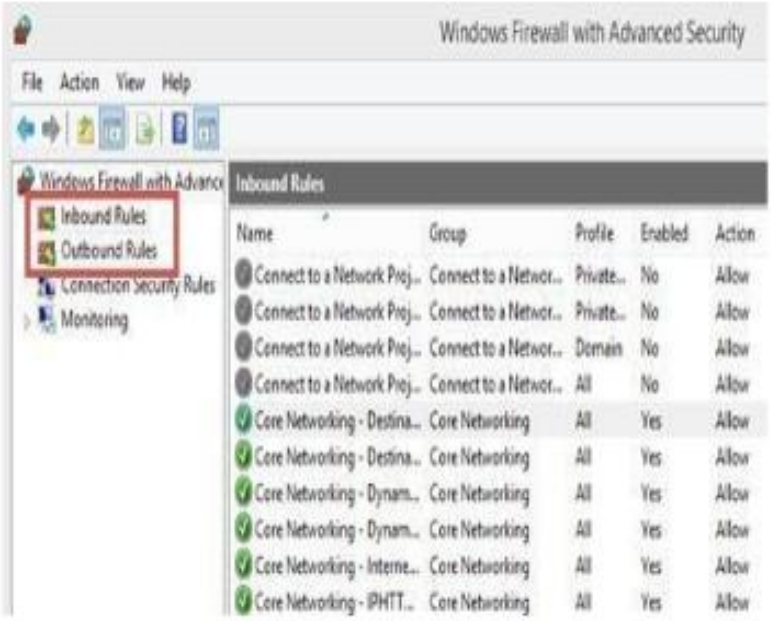
What Are The Inbound & Outbound Rules?

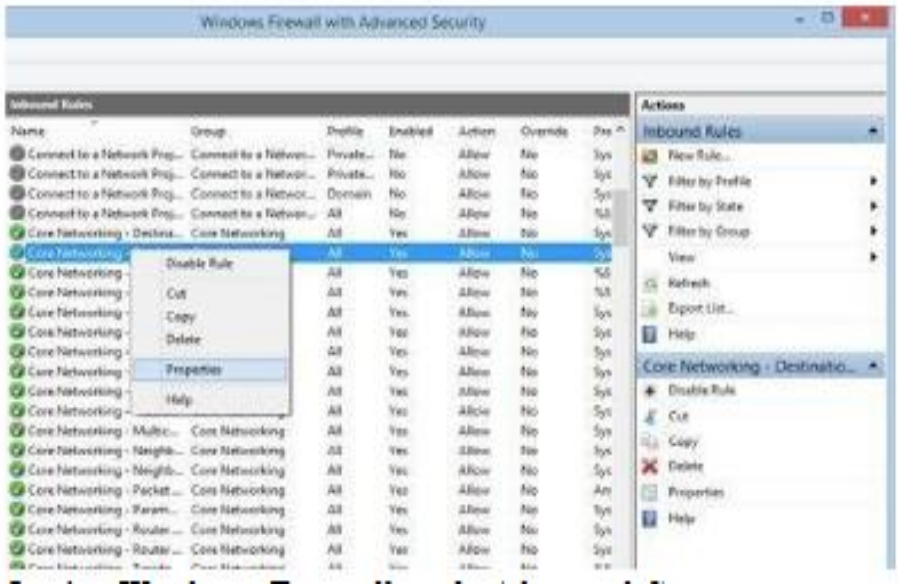
In order to provide the security you need, the Windows Firewall has a standard set of inbound and outbound rules, which are enabled depending on the location of the network you are connected to.

Inbound rules are applied to the traffic that is coming from the network and the Internet to your computer or device. Outbound rules apply to the traffic from your computer to the network or the Internet.

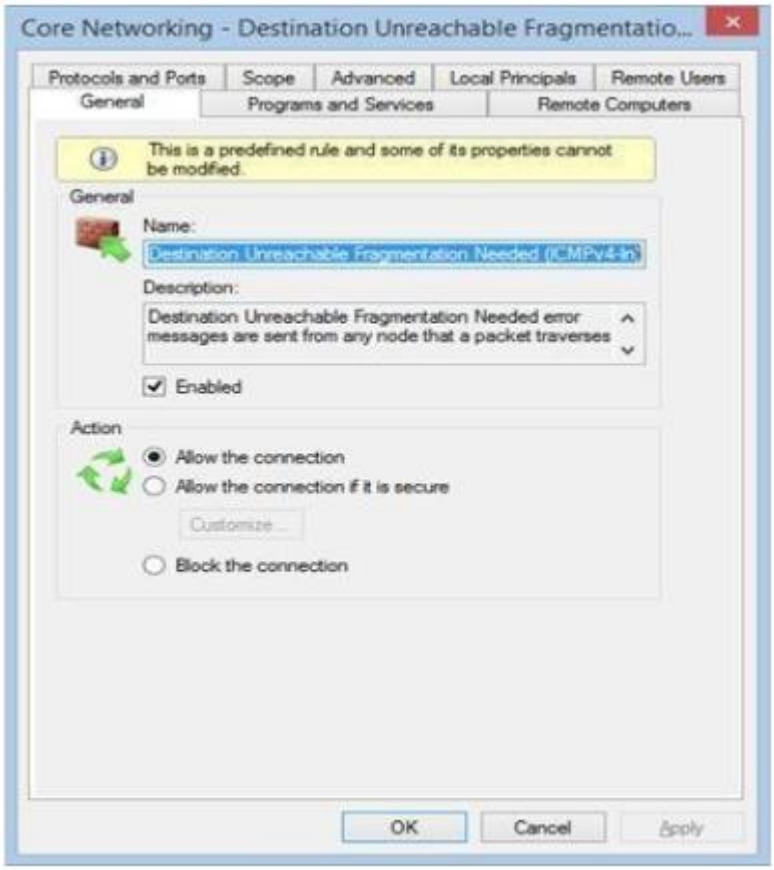
These rules can be configured so that they are specific to: computers, users, programs, services, ports or protocols. You can also specify to which type of network adapter (e.g. wireless, cable, virtual private network) or user profile it is applied to.

In the Windows Firewall with Advanced Security, you can access all rules and edit their properties. All you have to do is click or tap the appropriate unit in the left-side panel.





The rules used by the Windows Firewall can be enabled or disabled. The ones which are enabled or active are marked with a green check-box in the Name column. The ones that are disabled are marked with a gray check-box. If you want to know more about a specific rule and learn its properties, right click on it and select Properties or select it and press Properties in the column on right, which lists the actions that are available for your Selection.

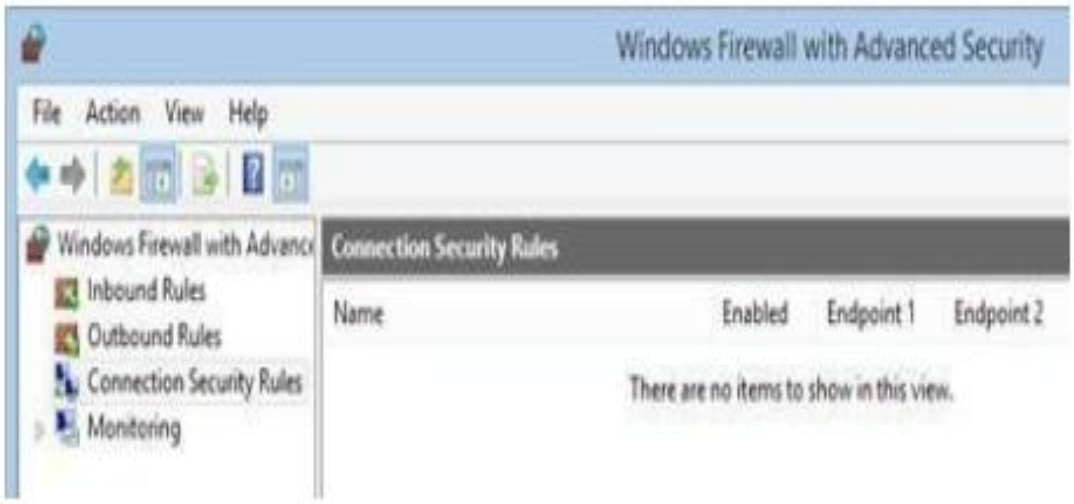


2.1.1.1 What Are The Connection Security Rules?

Connection security rules are used to secure traffic between two computers while it crosses the network. One example would be a rule which defines that connections between two specific computers must be encrypted.

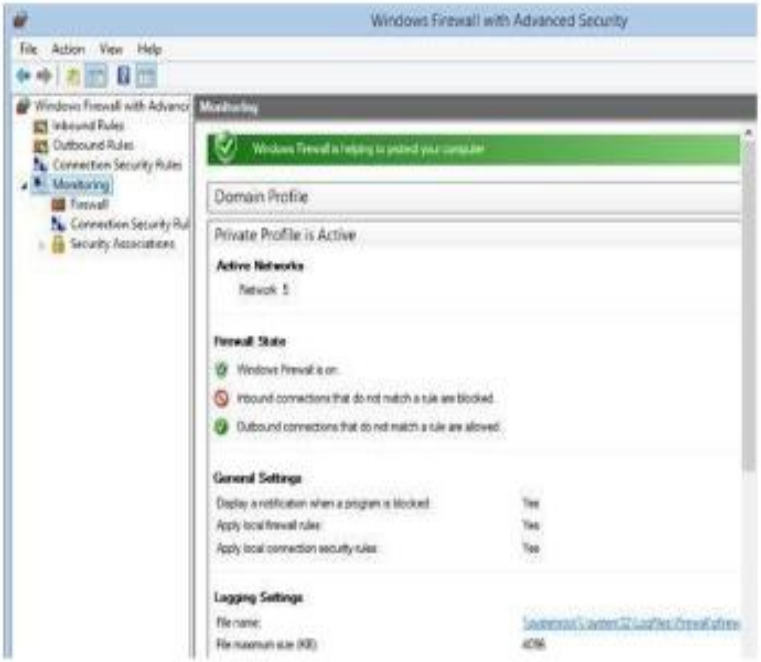
Unlike the inbound or outbound rules, which are applied only to one computer, connection security rules require that both computers have thesame rules defined and enabled.

If you want to see if there are any such rules on your computer, click or tap "Connection Security Rules" on the panel on the left. By default, there are no such rules defined on Windows computers and devices. They are generally used in business environments and such rules are set by the network administrator.



2.1.1.2 What Does the Windows Firewall with Advanced Security Monitor?

The Windows Firewall with Advanced Security includes some monitoring features as well. In the Monitoring section you can find the following information: the firewall rules that are active (both inbound and outbound), the connection security rules that are active and whether there are any active security associations



You should note that the Monitoring section shows only the active rules for the current network location.

RESULT:

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