# ASSIGNMENT NO: 7

**Aim:** Study of Honeypots

## INTRODUCTION:

Honeypot is a network-attached system used as a trap for cyber-attackers to detect and study the tricks and types of attacks used by hackers. It acts as a potential target on the internet and informs the defenders about any unauthorized attempt to the information system.

Honeypots are mostly used by large companies and organizations involved in cybersecurity. It helps cybersecurity researchers to learn about the different type of attacks used by attackers. It is suspected that even the cybercriminals use these honeypots to decoy researchers and spread wrong information.

A **honeynet** is a combination of two or more honeypots on a network.

- ## Types of Honeypot:

Honeypots are classified based on their deployment and the involvement of the intruder.

**Based on their deployment, honeypots are divided into:**

1. **Research honeypots**- These are used by researchers to analyse hacker attacks and deploy different ways to prevent these attacks.

2. **Production honeypots**- Production honeypots are deployed in production networks along with the server. These honeypots act as a frontend trap for the attackers, consisting of false information and giving time to the administrators to improve any vulnerability in the actual system.

- ## Honeypots are classified into:
    1. **Low interaction honeypots:** Low interaction honeypots gives very little insight and control to the hacker about the network. It simulates only the services that are frequently requested by the attackers. The main operating system is not involved in the low interaction systems and therefore it is less risky.
    2. **Medium Interaction Honeypots:** Medium interaction honeypots allows more activities to the hacker as compared to the low interaction honeypots. They can

expect certain activities and are designed to give certain responses beyond what a low-interaction honeypot would give.

3. **High Interaction honeypots:** A high interaction honeypot offers a large no. of services and activities to the hacker, therefore, wasting the time of the hackers and trying to get complete information about the hackers. These honeypots involve the real-time operating system and therefore are comparatively risky if a hacker identifies the honeypot

- ## How honeypots work:

The honeypot looks like a real computer system, with applications and data, fooling cybercriminals into thinking it's a legitimate target. For example, a honeypot could mimic a company's customer billing system - a frequent target of attack for criminals who want to find credit card numbers. Once the hackers are in, they can be tracked, and their behavior assessed for clues on how to make the real network more secure.

Honeypots are made attractive to attackers by building in deliberate security vulnerabilities. For instance, a honeypot might have ports that respond to a port scan or weak passwords. Vulnerable ports might be left open to entice attackers into the honeypot environment, rather than the more secure live network.

- ## The benefits of using honeypots:

Honeypots can be a good way to expose vulnerabilities in major systems. For instance, a honeypot can show the high level of threat posed by attacks on IoT devices. It can also suggest ways in which security could be improved.

Using a honeypot has several advantages over trying to spot intrusion in the real system. For instance, by definition, a honeypot shouldn't get any legitimate traffic, so any activity logged is likely to be a probe or intrusion attempt.

That makes it much easier to spot patterns, such as similar IP addresses (or IP addresses all coming from one country) being used to carry out a network sweep. By contrast, such tell-tale signs of an attack are easy to lose in the noise when you are looking at high levels of legitimate traffic on your core network. The big advantage of using honeypot security is that these malicious addresses might be the only ones you see, making the attack much easier to identify.

- **Use of Honeypots:**
    - **Database Attack**

A power company can set up a fake Microsoft SQL server that appears to contain a database of the locations of all the plants it uses to source the power it sells to customers.

So suppose the power company has eight hydroelectric plants, one nuclear power plant, 10 solar farms, and two coal-burning power plants that all provide power to the people the company serves. Network admins can create a fake database, host it on an SQL server, make it relatively easy to hack into, and then use this honeypot to see how hackers try to steal the information. Of course, the names of the power plants, and especially their geolocations, are all false.

In many cases, the IT team will create a system that closely parallels their real network setup. In this way, if hackers are able to get in, they can identify vulnerabilities in their actual setup.

It is important to keep in mind that honeypots in network security are designed based on your IT team's objectives. Consequently, honeypot security setups can vary drastically from one organization to another.

# CONCLUSION:

In this assignment we studied about honeypots.

# ASSIGNMENT NO.: 6

**Aim:** Configure and demonstrate use of vulnerability assessment tool like Wireshark or Snort.

## INTRODUCTION:

### ➢ What Is Wireshark?

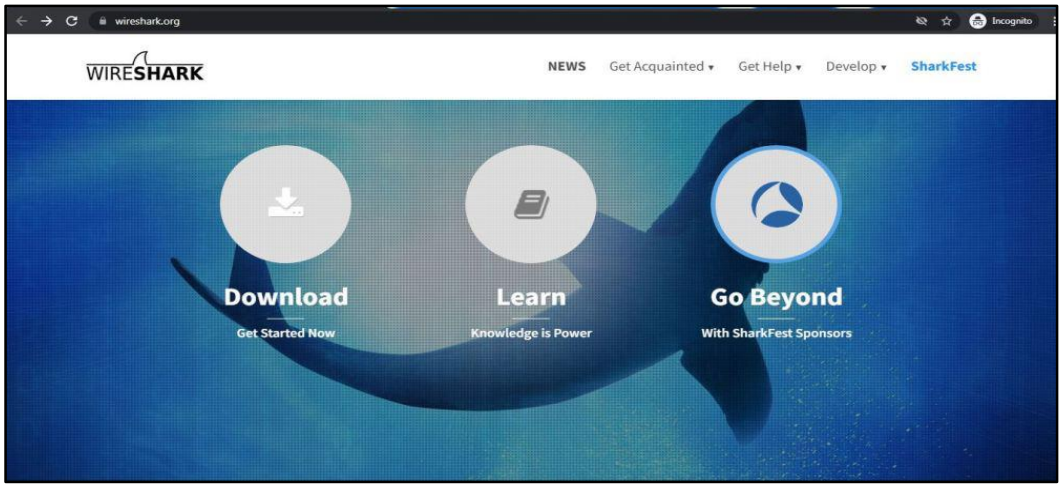
Wireshark is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

1. **Packet Capture:** Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.
2. **Filtering:** Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.
3. **Visualization:** Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

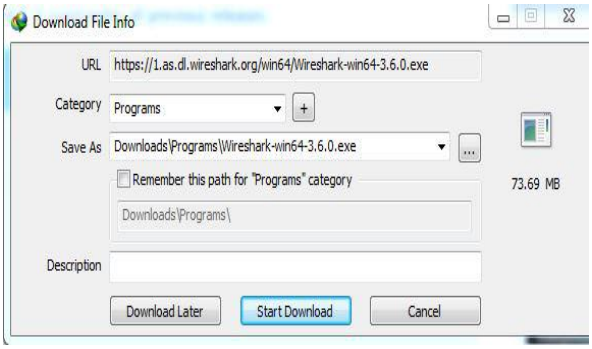
### ➢ Installing Wireshark on Windows:

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



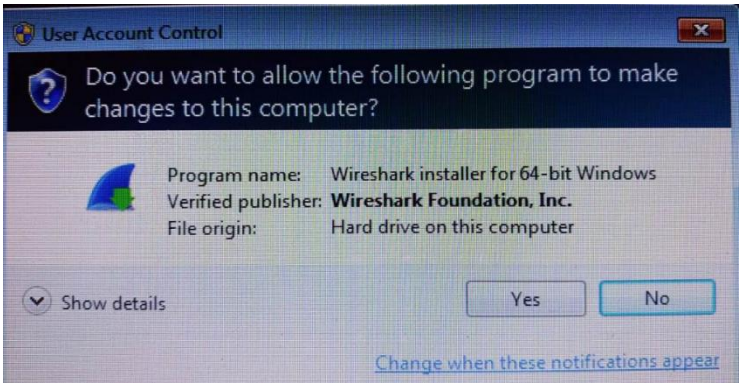
**Step 2:** Click on Download, a new webpage will open with different installers of Wireshark.


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

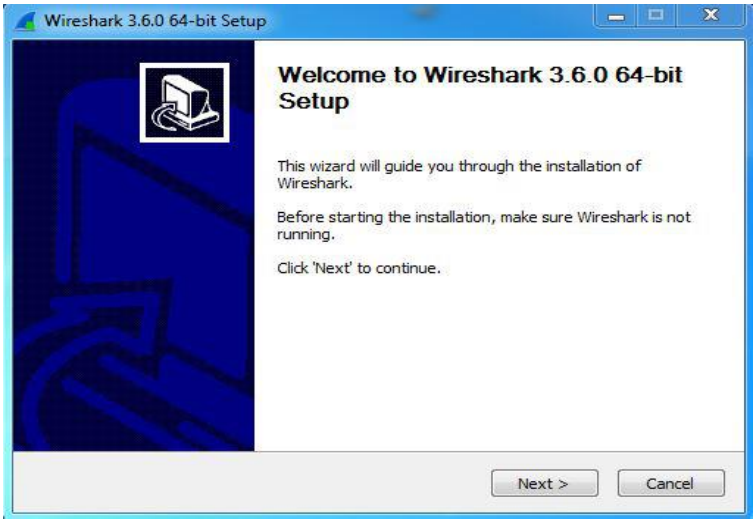



**Step 4:** Now check for the executable file in downloads in your system and run it.

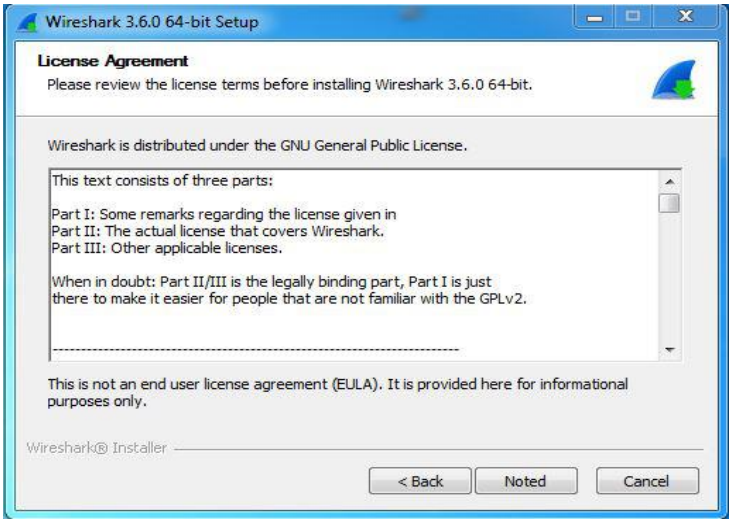

**Step 5:** It will prompt confirmation to make changes to your system. Click on Yes.



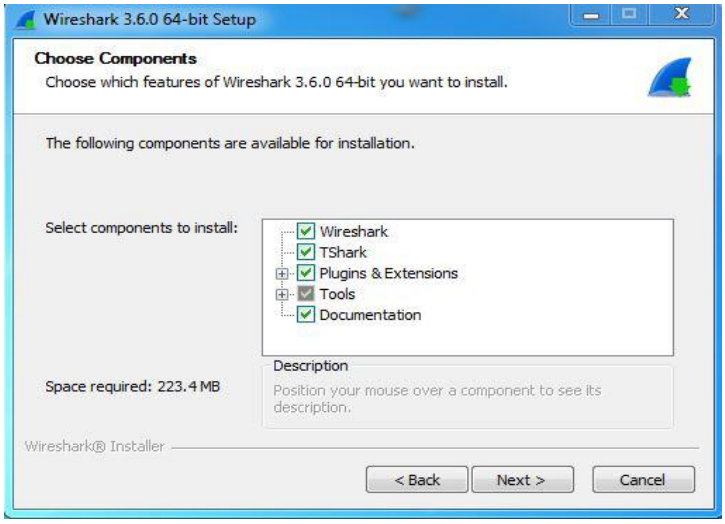

**Step 6:** Setup screen will appear, click on Next.



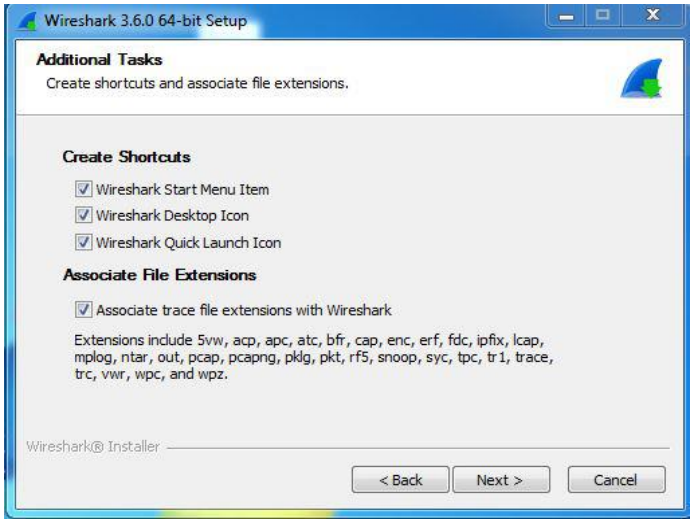

**Step 7:** The next screen will be of License Agreement, click on Noted.



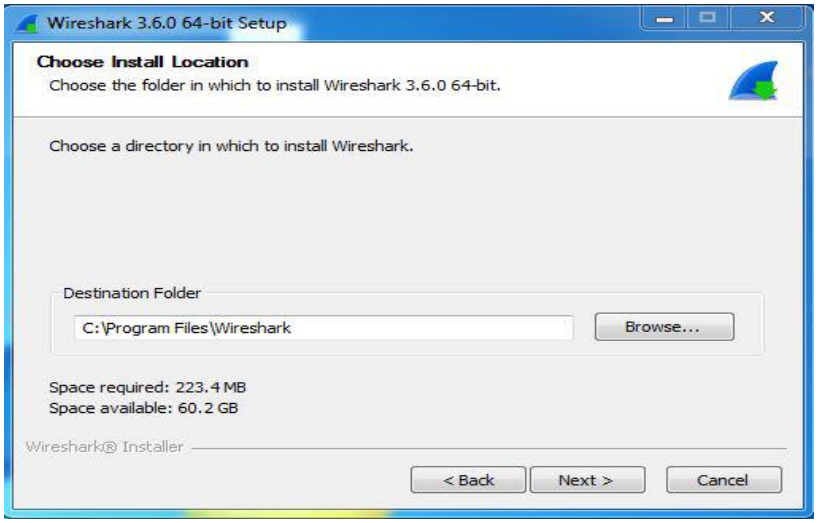

**Step 8:** This screen is for choosing components, all components are already marked so don't change anything just click on the Next button.



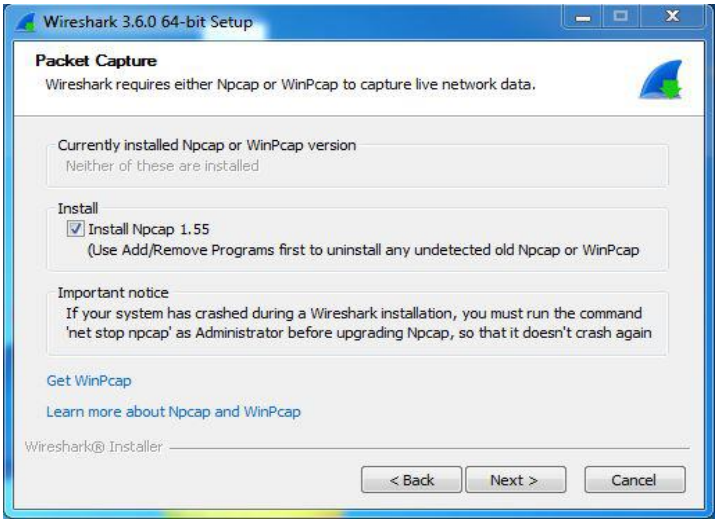

**Step 9:** This screen is of choosing shortcuts like start menu or desktop icon along with file extensions which can be intercepted by Wireshark, tick all boxes and click on Next button.




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

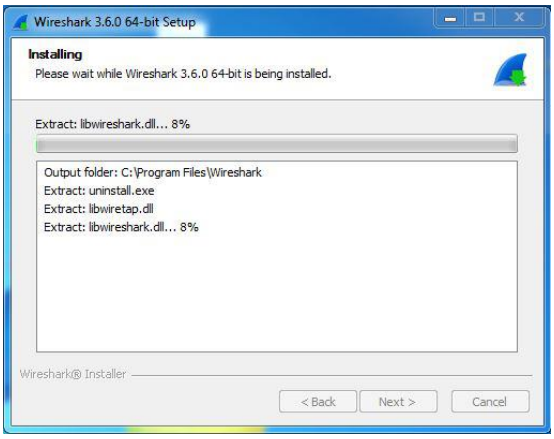



**Step 11:** Next screen has an option to install Npcap which is used with Wireshark to capture packets *pcap* means packet capture so the install option is already checked don't change anything and click the next button.

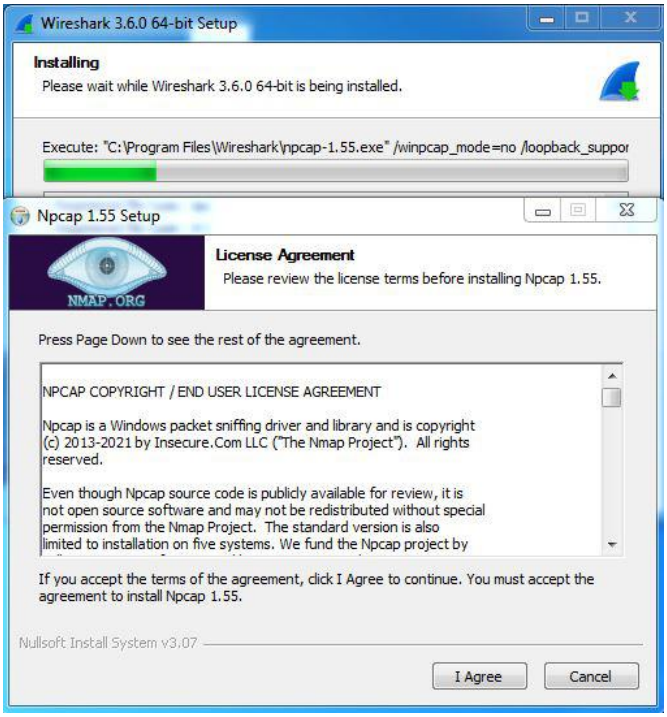


**Step 12:** Next screen is about USB network capturing so it is one's choice to use it or not, click on Install.

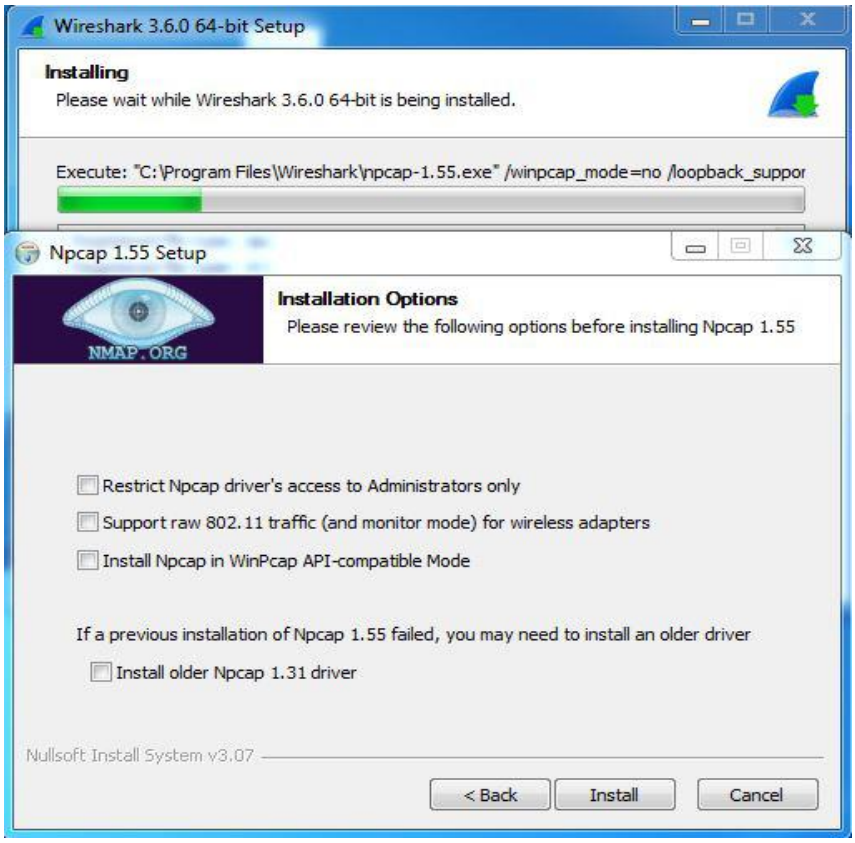

**Step 13:** After this installation process will start.



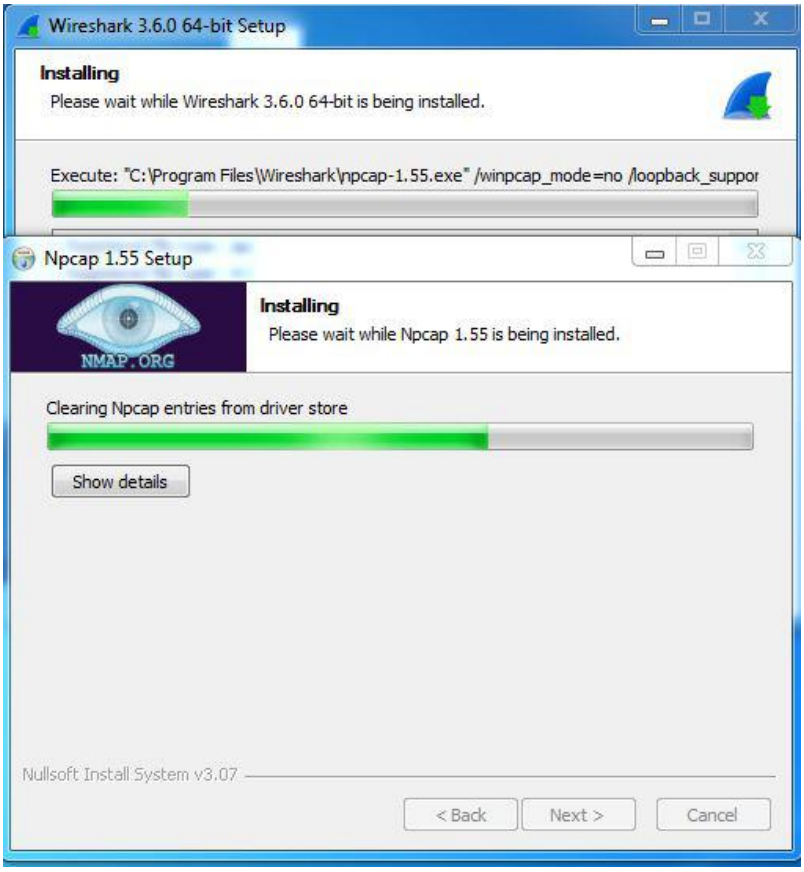

**Step 14:** This installation will prompt for Npcap installation as already checked so the license agreement of Npcap will appear to click on the *I Agree* button.



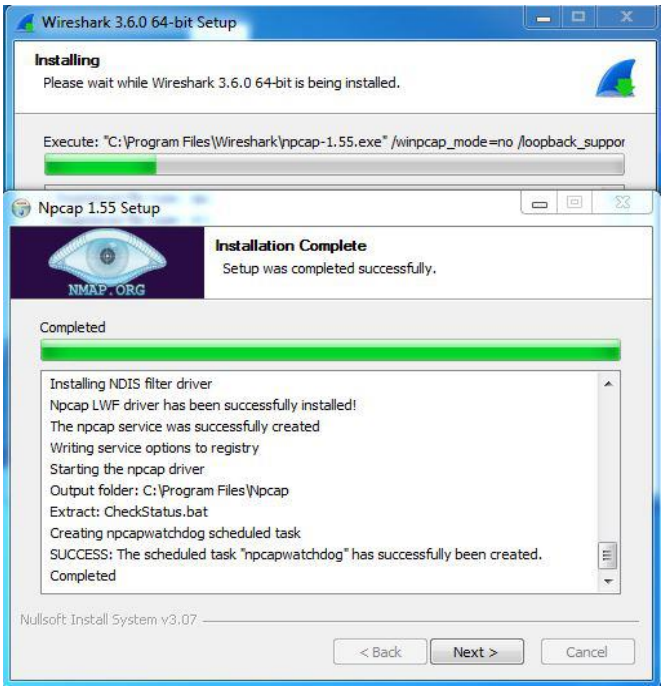
**Step 15:** Next screen is about different installing options of *npcap*, don't do anything click on Install.




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

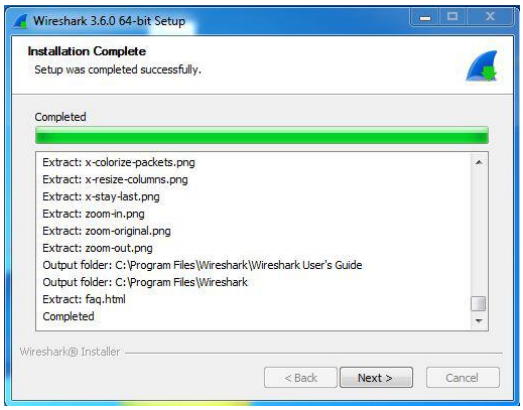


**Step 17:** After this installation process will complete click on the Next button.

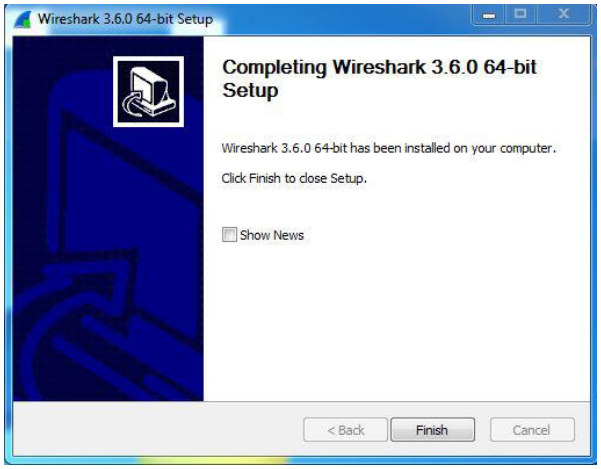



**Step 18:** Click on Finish after the installation process is complete.

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




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



## ➢ What Is Wireshark Used For?

Wireshark has many uses, including troubleshooting networks that have performance issues. Cybersecurity professionals often use Wireshark to trace connections, view the contents of suspect network transactions and identify bursts of network traffic. It's a major part of any IT pro's toolkit – and hopefully, the IT pro has the knowledge to use it.

## ➢ When Should Wireshark Be Used?

Wireshark is a safe tool used by government agencies, educational institutions, corporations, small businesses and non-profits alike to troubleshoot network issues. Additionally, Wireshark can be used as a learning tool.

1. It can't help a user who has little understanding of network protocols. No tool, no matter how cool, replaces knowledge very well. In other words, to properly use wireshark, you need to learn exactly how a network operates. That means, you need to understand things such as the three-way tcp handshake and various protocols, including tcp, udp, dhcp and icmp.
2. Wireshark can't grab traffic from all of the other systems on the network under normal circumstances. On modern networks that use devices called switches, Wireshark can only sniff traffic between your local computer and the remote system it is talking to.
3. While Wireshark can show malformed packets and apply colour coding, it doesn't have actual alerts; Wireshark isn't an intrusion detection system (ids).
4. Wireshark can't help with decryption with regards to encrypted traffic.

# CONCLUSION:

In this assignment we configured and demonstrated the use of vulnerability assessment tool like Wireshark.

# ASSIGNMENT NO.: 3

**AIM:** A person on nearby road is trying to enter into a Wi-Fi network by trying to crack the password to use the IP printer resource; write a program detect such attempt and prohibit the access/ Develop the necessary scenario by using IEEE 802.11, configure a Wi-Fi adapter and access point.

**Program:**

Sniff.py

```
#!/usr/bin/env python
import sys
from scapy.all import *
count =0
k=0
found=0

Matrix = [[0 for x in xrange(6)] for x in xrange(6)]
def packet_handler(p):
    global Matrix
    global k
    global count
    global found
    if p.haslayer(Dot11) and p.type==0 and p.subtype==11:
        print(p.show())
        #check adress of AP
        if p.addr2 not in Matrix and p.addr2!="00:90:4c:91:00:03":
            Matrix[k][0]=str(p.addr2)
            Matrix[k][1]=1
            k=k+1

        for i in range(len(Matrix)):

            if Matrix[i][0]==str(p.addr2):
                Matrix[i][1]+=1
                if Matrix[i][1]>3:
                    print "Authentication requests = "+ str(Matrix[i][1])+" Malicious user
detected with MAC ID "+str(Matrix[i][0])
                    found=1
                    return


                #print "Address of the Client is ="+str(p.addr2)

        #count=count+1
```

```
        #print "This is packet number"+str(count)
        #print p.addr1
    #for i in range(len(Matrix)):
    #    print "Packet address "+str(Matrix[i][0])+" count is "+str(Matrix[i][1])
sniff(iface="mon0", prn=packet_handler)
```

**Output:**

```
root@ubuntu:/home/hemlata# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:"dd-wrt"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:90:4C:91:00:03
          Bit Rate=11 Mb/s   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-38 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:61   Missed beacon:0

eth0      no wireless extensions.

root@ubuntu:/home/hemlata#

root@ubuntu:/home/hemlata# iw dev wlan0 interface add mon0 type monitor
root@ubuntu:/home/hemlata# iwconfig
lo        no wireless extensions.

mon0      IEEE 802.11bg  Mode:Monitor  Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

wlan0     IEEE 802.11bg  ESSID:"dd-wrt"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:90:4C:91:00:03
          Bit Rate=54 Mb/s   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-40 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:63   Missed beacon:0

eth0      no wireless extensions.


root@ubuntu:/home/hemlata# ifconfig mon0 up
root@ubuntu:/home/hemlata# iwconfig
lo        no wireless extensions.
```

```
mon0      IEEE 802.11bg  Mode:Monitor  Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

wlan0     IEEE 802.11bg  ESSID:"dd-wrt"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:90:4C:91:00:03
          Bit Rate=54 Mb/s   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=66/70  Signal level=-44 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:89   Missed beacon:0

eth0      no wireless extensions.

root@ubuntu:/home/hemlata#



root@ubuntu:/home/hemlata# python auth_sniff1.py
WARNING: No route found for IPv6 destination :: (no default route?)
###[ RadioTap dummy ]###
  version   = 0
  pad       = 0
  len       = 18
  present   = Flags+Rate+Channel+dBm_AntSignal+Antenna+b14
  notdecoded= '\x00\x02\x85\t\xa0\x00\xda\x01\x00\x00'
###[ 802.11 ]###
     subtype   = 11L
     type      = Management
     proto     = 0L
     FCfield   =
     ID        = 14849
     addr1     = 00:90:4c:91:00:03
     addr2     = 34:be:00:c6:1a:20
     addr3     = 00:90:4c:91:00:03
     SC        = 15792
     addr4     = None
###[ 802.11 Authentication ]###
        algo      = open
        seqnum    = 1
        status    = success
###[ 802.11 Information Element ]###
           ID        = vendor
           len       = 9
           info      = '\x00\x10\x18\x02\x00\x00\x04\x00\x00'
None
Address =34:be:00:c6:1a:20
###[ RadioTap dummy ]###
  version   = 0
```

```
  pad       = 0
  len       = 18
  present   = Flags+Rate+Channel+dBm_AntSignal+Antenna+b14
  notdecoded= '\x00\x02\x85\t\xa0\x00\xd0\x01\x00\x00'
###[ 802.11 ]###
     subtype   = 11L
     type      = Management
     proto     = 0L
     FCfield   =
     ID        = 14849
     addr1     = 34:be:00:c6:1a:20
     addr2     = 00:90:4c:91:00:03
     addr3     = 00:90:4c:91:00:03
     SC        = 2320
     addr4     = None
###[ 802.11 Authentication ]###
        algo      = open
        seqnum    = 2
        status    = success
###[ 802.11 Information Element ]###
           ID        = vendor
           len       = 9
           info      = '\x00\x10\x18\x02\x01\xf0\x00\x00\x00'
None
###[ RadioTap dummy ]###
  version   = 0
  pad       = 0
  len       = 18
  present   = Flags+Rate+Channel+dBm_AntSignal+Antenna+b14
  notdecoded= '\x00\x02\x85\t\xa0\x00\xde\x01\x00\x00'
###[ 802.11 ]###
     subtype   = 11L
     type      = Management
     proto     = 0L
     FCfield   =
     ID        = 14849
     addr1     = 00:90:4c:91:00:03
     addr2     = 34:be:00:c6:1a:20
     addr3     = 00:90:4c:91:00:03
     SC        = 16512
     addr4     = None
###[ 802.11 Authentication ]###
        algo      = open
        seqnum    = 1
        status    = success
###[ 802.11 Information Element ]###
           ID        = vendor
           len       = 9
           info      = '\x00\x10\x18\x02\x00\x00\x04\x00\x00'
None
```

```
Address =34:be:00:c6:1a:20
###[ RadioTap dummy ]###
  version   = 0
  pad       = 0
  len       = 18
  present   = Flags+Rate+Channel+dBm_AntSignal+Antenna+b14
  notdecoded= '\x00\x02\x85\t\xa0\x00\xd2\x01\x00\x00'
###[ 802.11 ]###
     subtype   = 11L
     type      = Management
     proto     = 0L
     FCfield   =
     ID        = 14849
     addr1     = 34:be:00:c6:1a:20
     addr2     = 00:90:4c:91:00:03
     addr3     = 00:90:4c:91:00:03
     SC        = 25392
     addr4     = None
###[ 802.11 Authentication ]###
        algo      = open
        seqnum    = 2
        status    = success
###[ 802.11 Information Element ]###
           ID        = vendor
           len       = 9
           info      = '\x00\x10\x18\x02\x01\xf0\x00\x00\x00'
None
###[ RadioTap dummy ]###
  version   = 0
  pad       = 0
  len       = 18
  present   = Flags+Rate+Channel+dBm_AntSignal+Antenna+b14
  notdecoded= '\x00\x02\x85\t\xa0\x00\xde\x01\x00\x00'
###[ 802.11 ]###
     subtype   = 11L
     type      = Management
     proto     = 0L
     FCfield   =
     ID        = 14849
     addr1     = 00:90:4c:91:00:03
     addr2     = 34:be:00:c6:1a:20
     addr3     = 00:90:4c:91:00:03
     SC        = 17520
     addr4     = None
###[ 802.11 Authentication ]###
        algo      = open
        seqnum    = 1
        status    = success
###[ 802.11 Information Element ]###
           ID        = vendor
```

```
        len       = 9
        info      = '\x00\x10\x18\x02\x00\x00\x04\x00\x00'
None
Address =34:be:00:c6:1a:20
Authentication requests = 4 Malicious user detected with MAC ID 34:be:00:c6:1a:20
```

## CONCLUSION:

In this assignment we studied about configuring Wi-Fi adapter and its access point.

# ASSIGNMENT NO. : 2

**Aim:** Implement a program to generate and verify CAPTCHA image.

## INTRODUCTION:

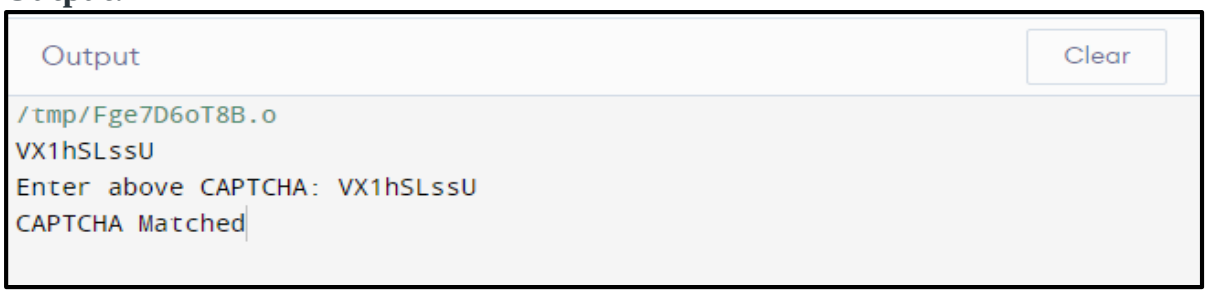
- A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a test to determine whether the user is human or not.
- The task is to generate unique CAPTCHA every time and to tell whether the user is human or not by asking user to enter the same CAPTCHA as generated automatically and checking the user input with the generated CAPTCHA.
- The set of characters to generate CAPTCHA are stored in a character array chrs[] which contains (a-z, A-Z, 0-9), therefore size of chrs[] is 62.
- To generate a unique CAPTCHA every time, a random number is generated using rand() function (rand()%62) which generates a random number between 0 to 61 and the generated random number is taken as index to the character array chrs[] thus generates a new character of captcha[] and this loop runs n (length of CAPTCHA) times to generate CAPTCHA of given length.

## Program:

```
import java.util.*;
import java.io.*;
class GFG
{
        static boolean checkCaptcha(String captcha, String user_captcha)
         {
                return captcha.equals(user_captcha);
         }
         static String generateCaptcha(int n)
         {
                Random rand = new Random(62);
                String chrs =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
                String captcha = "";
                  while (n-->0){
                        int index = (int)(Math.random()*62);
                        captcha+=chrs.charAt(index);
                  }
                  return captcha;
         }
         public static void main(String[] args)throws IOException
         {
                BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in));
```

```
                String captcha = generateCaptcha(9);
                 System.out.println(captcha);
                System.out.println("Enter above CAPTCHA: ");
                 String usr_captcha = reader.readLine();
                if (checkCaptcha(captcha, usr_captcha))
                        System.out.println("CAPTCHA Matched");
                 else
                        System.out.println("CAPTCHA Not Matched");
        }
}
```

**Output:**




## CONCLUSION:

In this assignment we studied how to generate and verify CAPTCHA image.