Lab 4: Packet Capture (IDS) – ARP Detection

Details

Aim: To provide define the capture of ARP information

Activities

1. The ARP protocol is important on networks, as it allows a node to determine the MAC address of a destination node on the same network. For security it is important, as it gives information on the activity on the local network. In this lab ARP packets will be captured, and then displayed for their basic information. The solution can be found at:

↳**http://www.dcs.napier.ac.uk/~bill/WinPCap4.zip**

2. The basic format of the ARP header is:

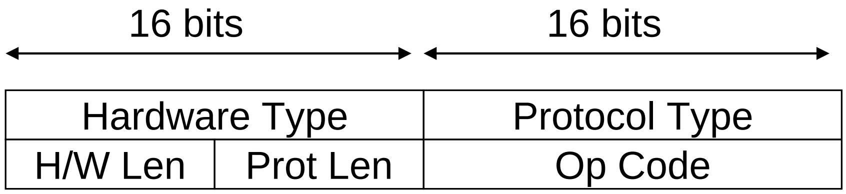

Figure 1: ARP header

Thus a program to capture the ARP packets is given next. Notice that the byte array is read for the first two bytes for the hardware type, and the next two for the protocol type [1]:

```
using System;
using Tamir.IPLib;
using Tamir.IPLib.Packets;

namespace NapierCapture
{
   public class CapturePackets
   {
      public static void Main(string[] args)
      {
         PcapDeviceList getNetConnections = SharpPcap.GetAllDevices();

         // network connection 1 (change as required)
         NetworkDevice netConn = (NetworkDevice)getNetConnections[1];
         PcapDevice device = netConn;

         // Define packet handler
         device.PcapOnPacketArrival +=
            new SharpPcap.PacketArrivalEvent(device_PcapOnPacketArrival);

         device.PcapOpen(true, 1000);
         Console.WriteLine("Network connection: {0}",device.PcapDescription);

         //Start the capturing process
         device.PcapStartCapture();

         Console.WriteLine("Press any <RETURN> to exit");
         Console.Read();

         device.PcapStopCapture();
         device.PcapClose();
      }
      private static void device_PcapOnPacketArrival(object sender, Packet packet)
      {

         if(packet is ARPPacket)
         {
               byte [] b = packet.Header;

               int type = b[1] + (b[0]<<8);

               int protocol = b[3] + (b[2]<<8);

               int opcode =  b[7] + (b[6]<<8);

            Console.WriteLine("ARP: Hardware type {0}, protocol {1}, op-code: {2}",
                       type,protocol,opcode);
      }
   }
}
```

Run the code, and ping a node on your network (one which you have not previously accessed for a while, or not at all), and examine the output:

Output of the program:

Did it detect the ARP packets:

What where the ARP types (from the op-code [2][1]):

3. Modify the code so that it displays the other fields in the ARP header.

4. Modify the code so that it displays the actual ARP type, rather than the code, Such as with:

[1] Note: For Ethernet, the **type** is normal set to 1 [2]. The **protocol** type for IP is 0x8000 (2048), and the table for the op-code is:

1	Request
2	Reply
3	Request Reverse
4	Rely Request

```
Console.Write("ARP: Hardware type {0}, protocol {1}, ",type,protocol);
if (opcode==1) Console.Write("{0}",opcode);
else if (opcode==2) …
```

References

[1] This code is based on the code wrapper for WinPCap developed by T.Gal [http://www.thecodeproject.com/csharp/sharppcap.asp].

[2] http://www.networksorcery.com/enp/protocol/arp.htm