# 6. INTRODUCTION TO THE LABORATORY: SOFTWARE TOOLS

### 6.1. Wireshark - network sniffer

Wireshark (originally called Ethereal) is a freeware network sniffer. A sniffer investigates and analyzes network traffic. It gives administrators the opportunity to recognize weaknesses and vulnerabilities in the network and quickly find a way to resolve problems.



Fig. 6.1 Wireshark GUI

Wireshark is very similar to *tcpdump* (a common packet analyzer that runs from the command line), but Wireshark has a graphical front-end, and many more information sorting and filtering options. Wireshark allows the user to see all traffic being passed over the network by putting the network interface into promiscuous mode (configuration of a network card that makes the card pass all traffic it receives to the central processing unit rather than just frames addressed to it).

## 6.2. Wireshark features

Wireshark is software that "understands" the structure of different networking protocols. Thus, it is able to display the encapsulation and the fields, together with their meanings, of different packets specified by different networking protocols. Wireshark uses packet capture (pcap) to capture packets, so it can only capture the packets on the types of networks that pcap supports.

- Data can be captured "from the wire" from a live network connection or read from a file that has recorded already-captured packets.
- Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.
- Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, tshark.
- Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.
- Data display can be refined using a display filter.
- Plug-ins can be created for dissecting new protocols.
- VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding the media flow can even be played.
- Raw USB traffic can be captured with Wireshark. This feature is currently available only under Linux.

## 6.3. Tutorial

1. Download and install the software. The latest version (1.4.1 at the time of writing)isavailablefordownladfromtheofficialsitehttp://www.wireshark.org/download.html

2. Once Wireshark is installed, start it up and you'll be presented with the blank screen shown below in Figure 6.2.

| The Wireshark Network Analyzer           |                                |                               |
|--|--------------------------------|-------------------------------|
| <u>File Edit View Go Capture Analyze</u> | <u>Statistics</u> <u>H</u> elp |                               |
|  |                                | @, @, @, [1]   ₩ 12 🖲 🔹       |
| <u>F</u> ilter:                          | ▼ Expression                   | . <u>C</u> lear <u>A</u> pply |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
|  |                                |                               |
| Ready to load or capture                 | No Packets                     | Profile: Default              |

Fig. 6.2 The Wireshark screen after startup

3. To start scanning, choose Interfaces from the Capture menu. You'll see a pop-up window similar to the one below (Figure 6.3).

#### Module 5

| Description                            | IP                   | Packets | Packets/s | 's Stop       |         |         |
|--|----------------------|---------|-----------|---------------|---------|---------|
| 📕 Broadcom 440x 10/100 Integrated Cont | roller 192.168.0.100 | 0       | 0         | <u>S</u> tart | Options | Details |
| . Microsoft                            | 192.168.0.28         | 276     | 25        | Start         | Options | Details |
| MS Tunnel Interface Driver             | unknown              | 0       | 0         | <u>S</u> tart | 0ptions | Details |
| . VMware Virtual Ethernet Adapter      | 192.168.28.1         | 0       | 0         | Start         | Options | Details |
| 📕 VMware Virtual Ethernet Adapter      | 192.168.48.1         | 0       | 0         | Start         | Options | Details |

Fig. 6.3 The pop-up window

If you'd like to configure advanced options -- like capturing a file, resolving MAC addresses and DNS names, or limiting the time or size of the capture -- click the Options button corresponding to the interface you wish to configure. Many of these options can help to improve the performance of Wireshark. For example, you can adjust settings to avoid name-resolution issues, as they will otherwise slow down your capture system and generate large numbers of name queries. Time and size limits can also place limitations on unattended captures. Otherwise, simply click the Start button next to the name of the interface on which you wish to capture traffic.

4. The Wireshark screen will immediately begin filling up with traffic seen on the network interface, as shown below in Figure 6.4.

| Flouule J |
|-----------|
|-----------|

| 📶 (Untitled) - Wireshark              | the second s |                                   |   |  | 3   |  |  |
|---------------------------------------|--|-----------------------------------|---|--|---|--|--|
| File Edit View Go Ca                  | pture <u>A</u> nalyze <u>S</u> tatistics   | <u>H</u> elp                      |   |  |   |  |  |
|                                       |  |                                   | Image: Second |  |   |  |  |
|                                       |  |                                   |   |  | es,11.2,3,9,4.2,1.4,1,5,7,1<br>s,11,2,3,9,4,2,1,4,1,5,8,1<br>es,11,2,3,9,4,2,1,4,1,5,8,1<br>1,21,A,64,236,91,23,A,64,22<br>Len-0 MSS=1460 WS=2<br>1 Win=8192 Len=0 MSS=1460<br>=17520 Len=0<br>tin=6960 Len=0 |  |  |
| Eiter   The Expression Clear Apply    |  |                                   |   |  |   |  |  |
| No Time                               | Source   | Destination                       | Protocol  | Info   | ^   |  |  |
| 366 11.767290                         | 192.168.0.31   | 192.168.0.28                      | SNMP  | get-response SNMPv2-SMI::enterprises.11.2.3.9.4.2.1.4.1.5.7.1            |   |  |  |
| 367 11.768865                         | 192.168.0.28   | 192.168.0.31                      | SNMP  | <pre>get-request SNMPv2-SMI::enterprises.11.2.3.9.4.2.1.4.1.5.8.1.</pre> |   |  |  |
| 369 11.775952                         | 192.168.0.31   | 192.168.0.28                      | SNMP  | get-response SNMPV2-SMI::enterprises.11.2.3.9.4.2.1.4.1.5.8.1            |   |  |  |
| 384 12 311862                         | 192.108.0.20   | 192.168.0.28                      | DNS   | Standard query response A 64 236 91 21 A 64 236 91 23 A 64 23            |   |  |  |
| 385 12, 312727                        | 192,168,0,28   | 64.236.91.21                      | TCP   | 56606 > http [SYN] Seg=0 win=8192 Len=0 MSS=1460 WS=2                    |   |  |  |
| 386 12.361495                         | 64.236.91.21   | 192.168.0.28                      | TCP   | http > 56606 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460              |   |  |  |
| 387 12.361583                         | 192.168.0.28   | 64.236.91.21                      | TCP   | 56606 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0                           |   |  |  |
| 388 12.361805                         | 192.168.0.28   | 64.236.91.21                      | HTTP  | GET / HTTP/1.1   |   |  |  |
| 389 12.413166                         | 64.236.91.21   | 192.168.0.28                      | TCP   | http > 56606 [ACK] Seq=1 Ack=845 Win=6960 Len=0                          |   |  |  |
| 390 12.413011                         | 64.236.91.21   | 192.168.0.28                      | TCP   | [ICP segment of a reassembled PDU]                                       |   |  |  |
| 591 12.414580                         | 04.250.91.21   | 192.108.0.28                      | TCP   | [TCP segment of a reassembled Pb0]                                       | -   |  |  |
| *                                     |  |                                   | 111   |  |   |  |  |
| 🗄 Frame 384 (167 by                   | tes on wire, 167 b   | ytes captured)                    |   |  | -   |  |  |
| 🕑 Ethernet II, Src:                   | sparklan_04:d0:9e  | (00:0e:8e:04:d0:9e), I            | Dst: HonHaiP  | Pr_26:66:a2 (00:1c:26:26:66:a2)  |   |  |  |
| Internet Protocol                     | , Src: 192.168.0.1   | (192.168.0.1), Dst: 1             | 92.168.0.28   | (192.168.0.28)   |   |  |  |
| H User Datagram Pro                   | tocol. Src Port: d   | omain (53), Dst Port:             | 62872 (62872  | ກໍ   |   |  |  |
| Domain Name System                    | em (response)  |                                   |   |  |   |  |  |
| [Request In: 38                       | 21]  |                                   |   |  |   |  |  |
| Time: 0 035771                        | 000 cocondel   |                                   |   |  |   |  |  |
| [11me. 0.023771                       | ourfite  |                                   |   |  |   |  |  |
| Transaction ID:                       | UXCTIT   | 2                                 |   |  |   |  |  |
| H Flags: 0x8180 (                     | standard query res   | ponse, No error)                  |   |  | =   |  |  |
| Questions: 1                          |  |                                   |   |  |   |  |  |
| Answer RRs: 6                         |  |                                   |   |  |   |  |  |
| Authority RRs:                        | 0  |                                   |   |  |   |  |  |
| Additional RRs:                       | : 0  |                                   |   |  |   |  |  |
| 🗉 Queries                             |  |                                   |   |  |   |  |  |
| www.cnn.com:                          | type A, class IN   |                                   |   |  |   |  |  |
| Name: www.c                           | nn.com   |                                   |   |  |   |  |  |
| Type: A (Ho                           | st address)  |                                   |   |  |   |  |  |
| Class: TN (                           | (0x0001)   |                                   |   |  |   |  |  |
|                                       | (0/0001)   |                                   |   |  |   |  |  |
|                                       | tuno A class Th  | addr 61 336 01 31                 |   |  |   |  |  |
| www.crin.com:                         | cype A, CIASS IN,  | auur 04.230.91.21                 |   |  | -   |  |  |
| 0000 00 1c 26 26 6                    | 6 a2 00 0e 8e 04 d   | 10 9e 08 00 45 00 8               | &f  | F  | 13  |  |  |
| 0010 00 99 00 00 4                    | 0 00 40 11 b8 e6 0   | 0 a8 00 01 c0 a8                  | .@.@.   |  | 1   |  |  |
| 0020 00 1c 00 35 f                    | 5 98 00 85 98 5a d   | f 1f 81 80 00 01                  | 5Z  |  | -   |  |  |
| 0030 00 06 00 00 0                    | 0 00 03 77 77 77 0   | 03 63 6e 6e 03 63                 | w ww.cn   | n. c   |   |  |  |
| 0050 b7 00 04 40 e                    | r = 00 01 c0 0c 00 0   | 00 01 00 00 00 00 0m.             | а г   |  |   |  |  |
| 0060 b7 00 04 40 e                    | c 5b 17 c0 0c 00 0   | 01 00 01 00 00 00                 | @. [  |  |   |  |  |
| 0070 b7 00 04 40 e                    | c 10 14 c0 0c 00 0   | 01 00 01 00 00 00                 | @   |  | -   |  |  |
| This is a response to the DNS         | query in this fr Packets   | 1273 Displayed: 909 Marked: 0 Dro | opped: 0  | Profile: Default   |   |  |  |
| · · · · · · · · · · · · · · · · · · · |  |                                   |   |  | - 110   |  |  |

Fig. 6.4 The Wireshark interface showing network traffic

Each line in the top pane of the Wireshark window corresponds to a single packet seen on the network. The default display shows the time of the packet (relative to the initiation of the capture), the source and destination IP addresses, the protocol used and some information about the packet. You can drill down and obtain more information by clicking on a row. This causes the bottom two window panes to fill with information.

The middle pane contains drill-down details on the packet selected in the top frame. The "+" icons reveal varying levels of detail about each layer of information contained within the packet. In the example above a DNS response packet was selected. The DNS response (application layer) section of the packet is expanded to show that the original was requesting a DNS resolution for www.cnn.com, and this response is informing us that the available IP addresses include 64.236.91.21. The bottom window pane shows the contents of the packet in both hexadecimal and ASCII representations. Module 5

5. Color is your friend when analyzing packets with Wireshark. Notice in the example above that each row is color-coded. The darker blue rows correspond to DNS traffic, the lighter blue rows are UDP SNMP traffic, and the green rows signify HTTP traffic. Wireshark includes a complex color-coding scheme (which you can customize). Figure 6. 5 shows the default settings.

| New       List is processed in order until match is found         Name       String         HSRP State Change       hsrp.state != 16         Spanning Tree Topology Change       stp.type == 0x80         OSPF State Change       ospf.msg != 1         ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       arp         ICMP       icmp         TCP RST       tcp.flags.reset eq 1         TL. low or unexpected       (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)         Checksum Errors       cdp.checksum_bad==1    edp.checksum_bad==1    ip.checksum_bad==1    tp.checksum_bad==1   |      | Filter                        |  | Order        |
|--|------|-------------------------------|--|--------------|
| Istring       Name       String         HSRP State Change       hsrp.state != 8 && hsrp.state != 16         Spanning Tree Topology Change       stp.type == 0x80         OSPF State Change       ospf.msg != 1         ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       arp         ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       arp         ICMP       icmp         TCP RST       tcp.flags.reset eq 1         TTL low or unexpected       (! ip.dst == 224.00.0/4 && ip.ttl < 5)    (ip.dst == 224.00.0/24 && ip.ttl != 1)         Checksum Errors       cdp.checksum_bad==1    edp.checksum_bad==1    ip.checksum_bad==1            Disable       SMB       smb    nbss    nbns    nbixx    ipxsap    netbios         HTTP       http    tcp.port == 80         IVX       ipx    spx         Delete       Geerpc         Routing       hsrp    eigrp    ospf    bgp    cdp    vrrp    gyrp    igmp    ismp         TCP SYN/FIN       tcp.flags & 0.02    tcp.flags.fin == 1         TCP       tcp         up       up         Broadcast       eth[0] & 1  | ew   |                               | List is processed in order until match is found  |              |
| HSRP State Change       hsrp.state != 8 && hsrp.state != 16         Spanning Tree Topology Change       stp.type == 0x80         OSPF State Change       ospf.msg != 1         ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       ap         ICMP       icmp         TCP RST       tcp.flags.reset eq 1         TTL low or unexpected       (! ip.dst == 224.0.00/4 && ip.ttl < 5)    (ip.dst == 224.0.00/24 && ip.ttl != 1)  | CVV  | Name                          | String   |              |
| Edit       Spanning Tree Topology Change       stp.type == 0x80         OSPF State Change       ospf.msg != 1         ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       ap         Enable       ICMP         TCP RST       tcp.flags.reset eq 1         TTL low or unexpected       (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl = 1)  |      | HSRP State Change             | hsrp.state != 8 && hsrp.state != 16  |              |
| Edit       OSPF State Change       ospf.msg != 1         ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       arp         ICMP       icmp         TCP RST       tcp.flags.reset eq 1         TTL low or unexpected       (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)   |      | Spanning Tree Topology Change | stp.type == 0x80   | Up           |
| ICMP errors       icmp.type eq 3    icmp.type eq 5    icmp.type eq 11         ARP       arp         ICMP       icmp         TCP RST       tcp.flags.reset eq 1         TIL low or unexpected       (! ip.dst == 224.00.0/4 && ip.ttl < 5)    (ip.dst == 224.00.0/24 && ip.ttl != 1)  | it   | OSPF State Change             | ospf.msg != 1  |              |
| ARP     arp       Enable     ICMP     icmp       TCP RST     tcp.flags.reset eq 1       TL low or unexpected     (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)  |      | ICMP errors                   | icmp.type eq 3    icmp.type eq 4    icmp.type eq 5    icmp.type eq 11                        |              |
| ICMP       icmp         TCP RST       tcp.flags.reset eq.1         TTL low or unexpected       (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)  |      | ARP                           | arp  |              |
| TCP RST         tcp.flags.reset eq 1           TTL low or unexpected         (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)  | able | ICMP                          | icmp   |              |
| TTL low or unexpected         (! ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)           Checksum Errors         cdp.checksum_bad==1    edp.checksum_bad==1    ip.checksum_bad==1    ip.c  |      | TCP RST                       | tcp.flags.reset eq 1   |              |
| Checksum Errors       cdp.checksum_bad==1    edp.checksum_bad==1    ip.checksum_bad==1    ip   |      | TTL low or unexpected         | (!ip.dst == 224.0.0.0/4 && ip.ttl < 5)    (ip.dst == 224.0.0.0/24 && ip.ttl != 1)            | Move         |
| SMB         smb    nbss    nbns    nbpx    ipxsap    netbios           HTTP         http    tcp.port == 80           IPX         ipx    spx           Delete         DCERPC           Routing         hsrp    eigrp    ospf    bgp    cdp    vrrp    gvrp    igmp    ismp           TCP         Ktp.flags & 0x02    tcp.flags.fin == 1           TCP         tcp           UDP         udp           Broadcast         eth[0] & 1  |      | Checksum Errors               | cdp.checksum_bad==1    edp.checksum_bad==1    ip.checksum_bad==1    tcp.checksum_bad==1    = | selected fil |
| HTTP         http    tcp.port == 80           IPX         ipx    spx           Delete         IpX           ge         CERPC           Routing         hsrp    eigrp    ospf    bgp    cdp    vrrp    gvrp    igmp    ismp           TCP SYN/FIN         tcp.flags &0 x02    tcp.flags.fin == 1           TCP         tcp           UDP         udp           Broadcast         eth[0] &1  |      | SMB                           | smb    nbss    nbns    nbipx    ipxsap    netbios  | up or dow    |
| IPX         ipx    spx           Delete         DCERPC         dcerpc           Routing         hsrp    eigrp    ospf    bgp    cdp    vrrp    gvrp    igmp    ismp         TCP SYN/FIN           tcp.flags & 0x02    tcp.flags.fin == 1         TCP           mport         UDP         udp           Broadcast         eth[0] & 1         v  |      | НТТР                          | http    tcp.port == 80   |              |
| Delete         DCERPC         dcerpc           ige         hsrp    eigrp    ospf    bgp    cdp    vrrp    gvrp    igmp    ismp           xport         TCP SYN/FIN         tcp.flags.& 0x02    tcp.flags.fin == 1           TCP         tcp           UDP         udp           Broadcast         eth[0] & 1   |      | IPX                           | ipx    spx   |              |
| ge         Routing         hsrp    eigrp    ospf    bgp    cdp    vrrp    gwrp    igmp    ismp           rge         TCP SYN/FIN         tcp.flags & 0x02    tcp.flags.fin == 1           rcp         tcp           udp         Broadcast         eth[0] & 1   | lete | DCERPC                        | dcerpc   |              |
| TCP SYN/FIN         tcp.flags & 0x02    tcp.flags.fin == 1           TCP         tcp           UDP         udp           Broadcast         eth[0] & 1  |      | Routing                       | hsrp    eigrp    ospf    bgp    cdp    vrrp    gvrp    igmp    ismp                          |              |
| TCP tcp<br>mport UDP udp Broadcast eth[0] & 1  | ort  | TCP SYN/FIN                   | tcp.flags & 0x02    tcp.flags.fin == 1   | -            |
| UDP         udp           Broadcast         eth[0] & 1   |      | ТСР                           | tcp  | <u>D</u> own |
| Clear The second | ort  | UDP                           | udp  |              |
| <u>Cical</u> m   |      | Broadcast                     | eth[0] & 1   |              |
|  |      |                               |  |              |

Fig. 6. 5 The Wireshark default settings

# 6.1. Further reading and references

I've put these in alphabetical order

Are they References or Further reading or Bibliography

If the date is the publication date it is usually enough just to put the year.

It isn't necessary to give the ISBN – although it helps.

Chappell, L. & Combs, G. (March 31, 2010). Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide. Protocol Analysis Institute, dba

Chappell University. p. 800. This means "page 800" If you want to say the book has 800 pages you need to write "pp. 800. ISBN 1893939995

Orebaugh, A., Ramirez, G. & Beale, J. (February 14, 2007). *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress. p. 448.As above – pp. 448. Also who is the publisher? or is it available from the website you write below? ISBN 1597490733. http://www.syngress.com/hacking-and-penetration-testing/Wireshark-amp-Ethereal-Network-Protocol-Analyzer-Toolkit/

Sanders, C. (May 23, 2007). *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*. No Starch Press. p. 192. ISBN 1593271492. http://nostarch.com/packet.htm As above.

Author: Tomasz Miklis