Institute of Computational Intelligence
Częstochowa University of Technology

The Wireshark – filters and statistics

Foundations of computer networks laboratory

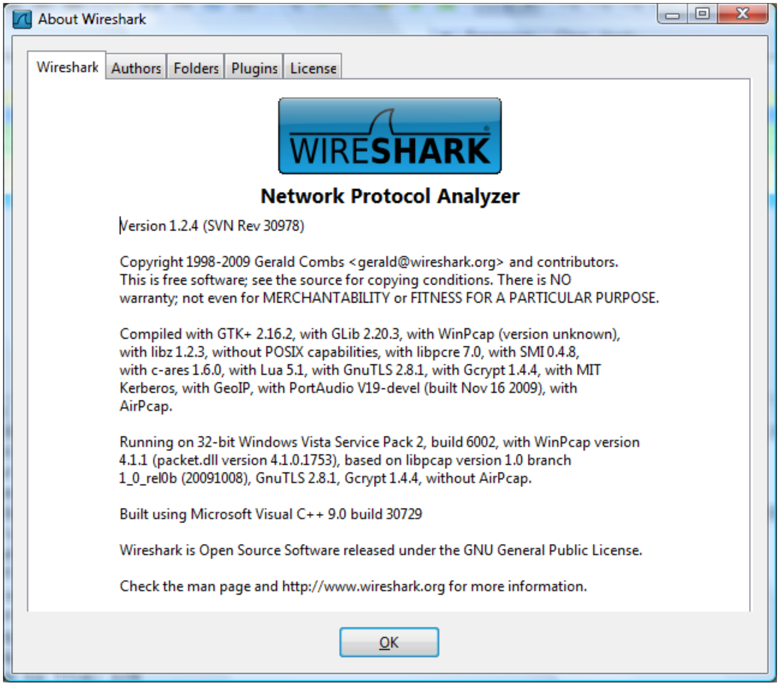
The objective of the exercises

Celem ćwiczenia jest zapoznanie się z wybranymi funkcjami analizatorów protokołów.

Introduction

The Wireshark – network protocol analyser

The Wireshark is a probably most popular software network protocol analyser. It is inter alia a result of applied open GNU licence and constantly improving its functionality. The Wireshark is a successor of Ethereal project.



Network Protocol Analyzer

Version 1.2.4 (SVN Rev 30978)

Copyright 1998-2009 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compiled with GTK+ 2.16.2, with GLib 2.20.3, with WinPcap (version unknown),
with libz 1.2.3, without POSIX capabilities, with libpcre 7.0, with SMI 0.4.8,
with c-ares 1.6.0, with Lua 5.1, with GnuTLS 2.8.1, with Gcrypt 1.4.4, with MIT
Kerberos, with GeoIP, with PortAudio V19-devel (built Nov 16 2009), with
AirPcap.

Running on 32-bit Windows Vista Service Pack 2, build 6002, with WinPcap version
4.1.1 (packet.dll version 4.1.0.1753), based on libpcap version 1.0 branch
1_0_rel0b (20091008), GnuTLS 2.8.1, Gcrypt 1.4.4, without AirPcap.

Built using Microsoft Visual C++ 9.0 build 30729

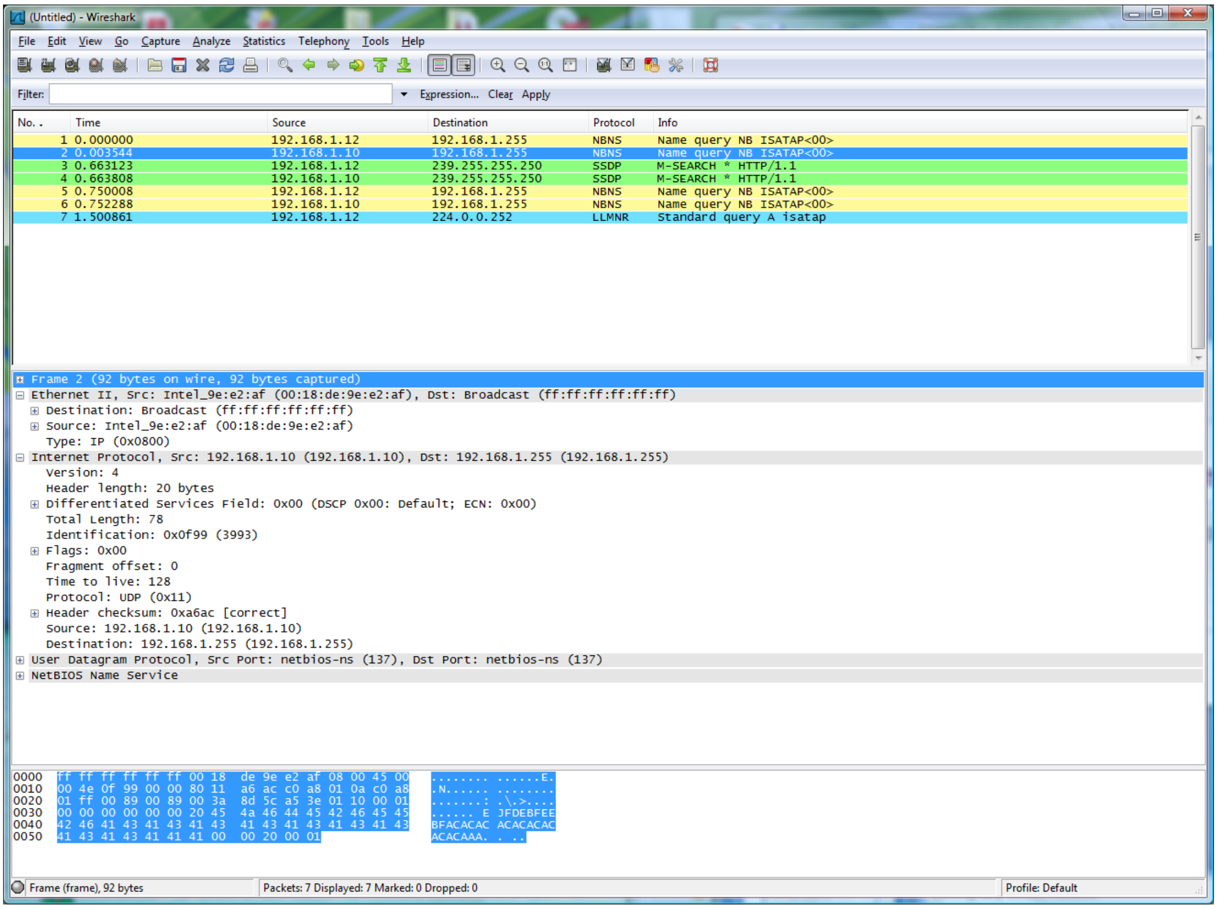
Wireshark is Open Source Software released under the GNU General Public License.

Check the man page and http://www.wireshark.org for more information.

The window of the application contains few elements. There are:

- menu,
- toolbar,
- display filter bar,
- packet list,
- packet details,
- packet bytes (hex and ASCII representation),
- status bar.

The visibility of above elements could be turned on and off individually.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
No. .  Time        Source          Destination      Protocol  Info
    1 0.000000    192.168.1.12    192.168.1.255    NBNS      Name query NB ISATAP<00>
    2 0.003544    192.168.1.10    192.168.1.255    NBNS      Name query NB ISATAP<00>
    3 0.663123    192.168.1.12    239.255.255.250  SSDP      M-SEARCH * HTTP/1.1
    4 0.663808    192.168.1.10    239.255.255.250  SSDP      M-SEARCH * HTTP/1.1
    5 0.750008    192.168.1.12    192.168.1.255    NBNS      Name query NB ISATAP<00>
    6 0.752288    192.168.1.10    192.168.1.255    NBNS      Name query NB ISATAP<00>
    7 1.500861    192.168.1.12    224.0.0.252      LLMNR     Standard query A isatap
```

```
⊞ Frame 2 (92 bytes on wire, 92 bytes captured)
⊟ Ethernet II, Src: Intel_9e:e2:af (00:18:de:9e:e2:af), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊞ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊞ Source: Intel_9e:e2:af (00:18:de:9e:e2:af)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.1.10 (192.168.1.10), Dst: 192.168.1.255 (192.168.1.255)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 78
    Identification: 0x0f99 (3993)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0xa6ac [correct]
    Source: 192.168.1.10 (192.168.1.10)
    Destination: 192.168.1.255 (192.168.1.255)
⊞ User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
⊞ NetBIOS Name Service
```

```
0000  ff ff ff ff ff ff 00 18  de 9e e2 af 08 00 45 00   ........ ......E.
0010  00 4e 0f 99 00 00 80 11  a6 ac c0 a8 01 0a c0 a8   .N...... ........
0020  01 ff 00 89 00 89 00 3a  8d 5c a5 3e 01 10 00 01   .......: .\.>....
0030  00 00 00 00 00 00 20 45  4a 46 44 45 42 46 45 45   ...... E JFDEBFEE
0040  42 46 41 43 41 43 41 43  41 43 41 43 41 43 41 43   BFACACAC ACACACAC
0050  41 43 41 43 41 41 41 00  00 20 00 01               ACACAAA. . ..
```

Frame (frame), 92 bytes | Packets: 7 Displayed: 7 Marked: 0 Dropped: 0 | Profile: Default

Capturing could be began in various ways, e.g. showing the list of available network interfaces and pressing **Start** button in chosen interface line or **Option** button then **Start** button in the Option window. The second way allows setting some parameters of capturing.

The very important aspect in Wireshark applying is the filters. There are display filters and capture filters.

Filters

Filters – relationships operators

eq, ==	equal
ne, !=	not equal
gt, >	greater
lt, <	lower
ge, >=	greater or equal
le, <=	lower or equal

Filters – logic operators

and, &&	logical AND
or, ||	logical OR
not, !	logical NOT

Filters - protocols

arp, dns, tcp, udp, ip, ipv6, irc, idp, ipx, http, pop, smtp, ftp, gnutella, image-jfif, kerberos, l2tp, netlogon, smb...

Filters – protocol fields (eth)

eth.addr ==
eth.dst ==
eth.len ==
eth.src ==
eth.trailer ==
eth.type ==

Filters – protocol fields (IP)

ip.dst eq www.mit.edu
ip.src == 192.168.1.1
ip.addr == 129.111.0.0/16
ip.fragment ==
ip.id ==
ip.len ==
ip.ttl ==

Filters – protocol fields (TCP)

tcp.port == 80
tcp.dstport ==
tcp.srcport ==
tcp.ack == numer potwierdzenia
tcp.flags == flaga 8-bit
tcp.flags.reset {ack, syn, fin, }
tcp.len == ???
tcp.window_size ==

Filters – protocol fields (UDP)

udp.checksum
udp.checksum_bad
udp.dstport
udp.length
udp.port
udp.srcport

Filters – protocol fields (HTTP)

http.cookie ==
http.host ==

Filters – protocol fields (echo)

echo.data ==
echo.request
echo.response

Examples:

Traffic of telnet service for particular host
tcp.port==23 and host==10.0.0.5

Traffic of telnet service for all host except selected one
tcp.port==23 and not host==10.0.0.5

Course of exercise

Student should capture the huge number of frames from the network without capture filter. To do this, student should start the capturing and then run various programs that use network resources and diagnostic programs, e.g. web browser, e-mail client, ftp, ping, nslookup, net, search network resources Windows Explorer.

The filter under examination should allow to find the frames, which

- has send to used (our) computer only,
- has send by (from) used (our) computer,
- apply arp protocol,
- apply icmp protocol,
- apply ftp protocol,
- apply pop protocol,
- apply smtp protocol,
- apply arp protocol,
- apply arp protocol and has send from used (our) computer
- other, proposed by students or teacher.

Then student should create the charts of traffic in the local collision domain for various protocols (http, udp, tcp, smb, icmp, arp itp.).

The raport

Students work in pairs or alone. Each team examines various display filters uses captured traffic. The results and comments should be included in the raport.