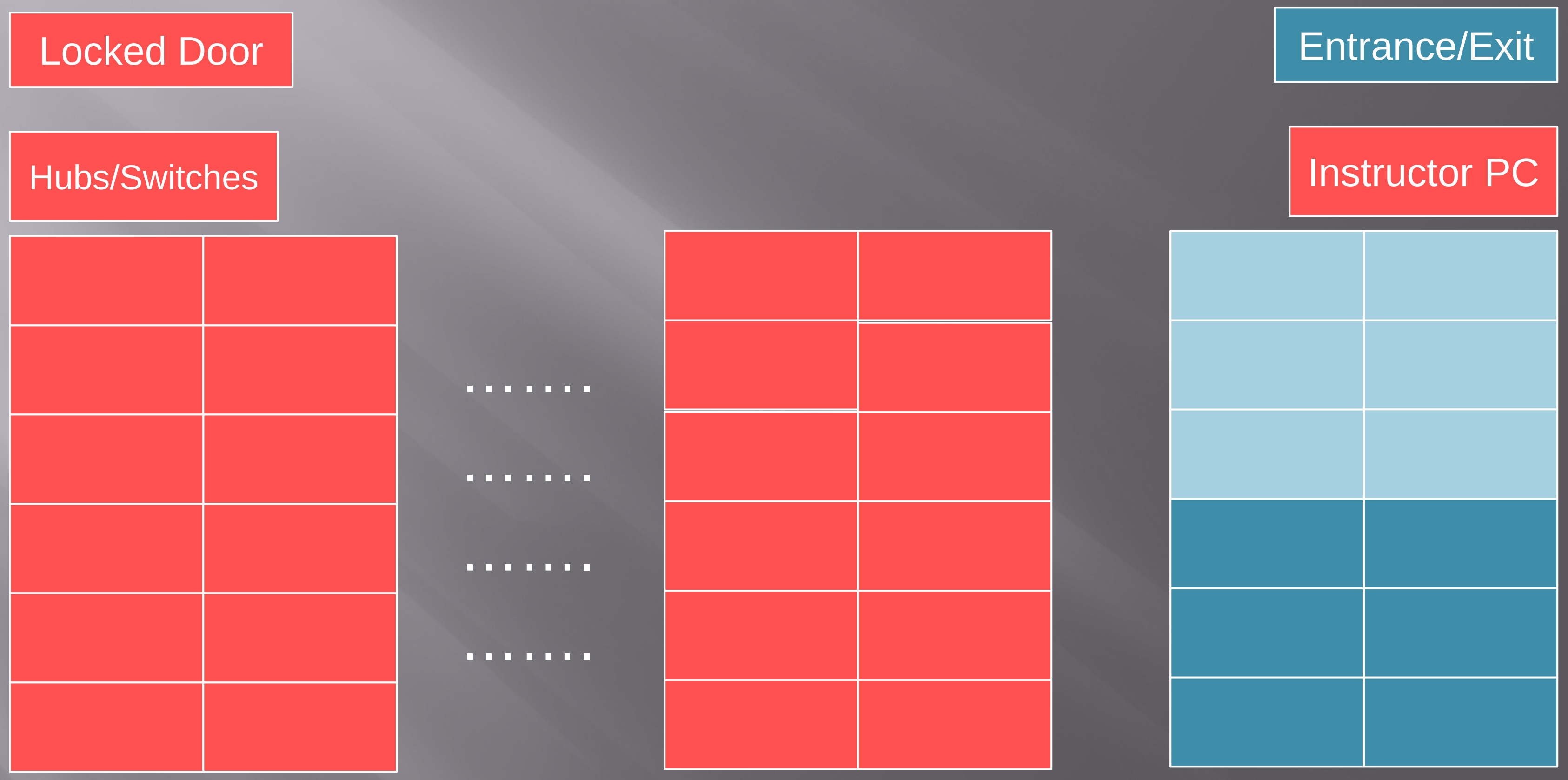
# ASSIGNMENT 1 BRIEFING

CS5321

# OS and Security Lab



Locked Door

Entrance/Exit

Hubs/Switches

Instructor PC

.......

.......

.......

.......

Legend: 🟥 Connected to Switch

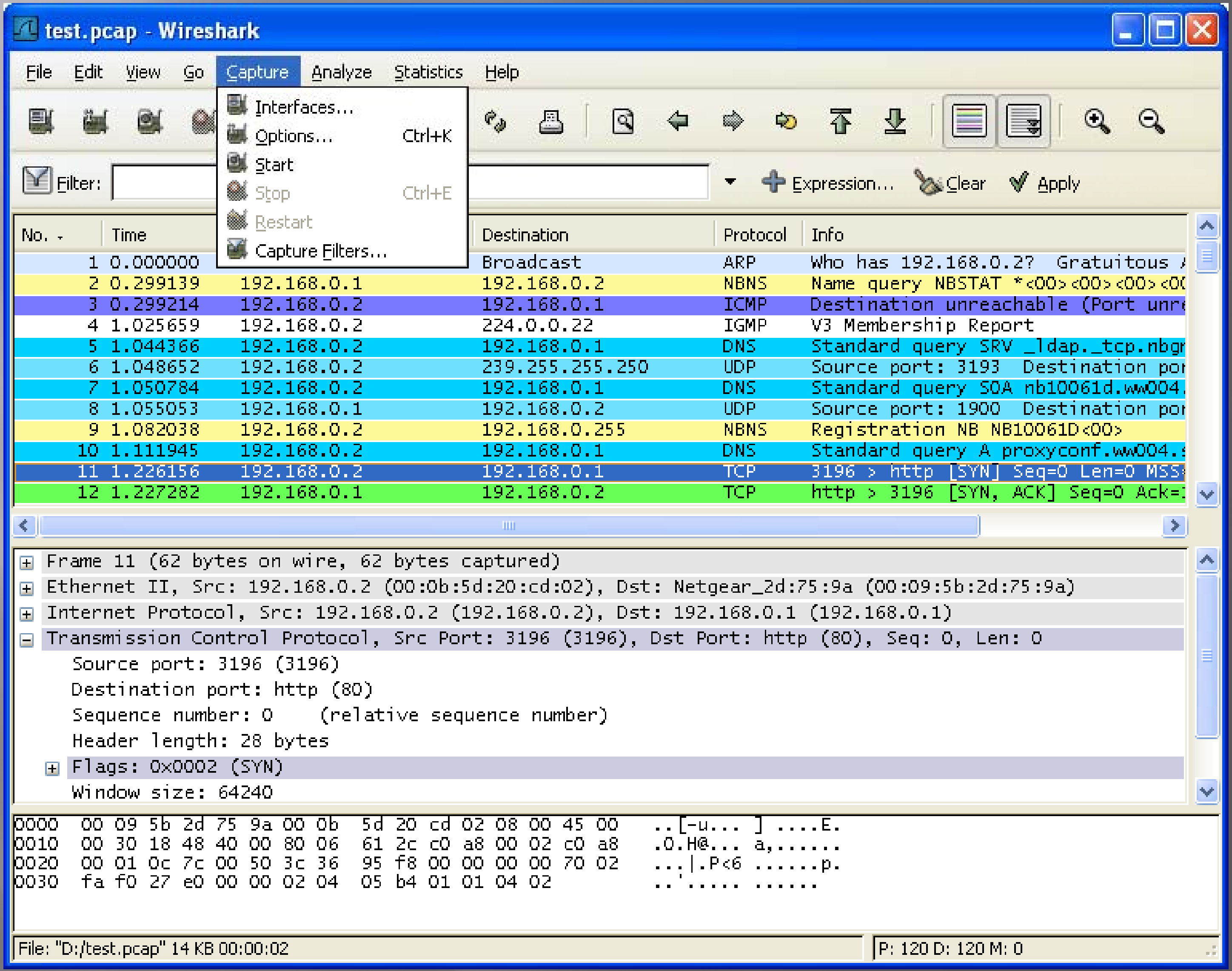
🟦 Connected to Hub1

■ Connected to Hub2

# Wireshark

- http://www.wireshark.org/
- Network Protocol Analyzer
- Sniff traffic and expose the data and protocols that pass along the wire

# Wireshark



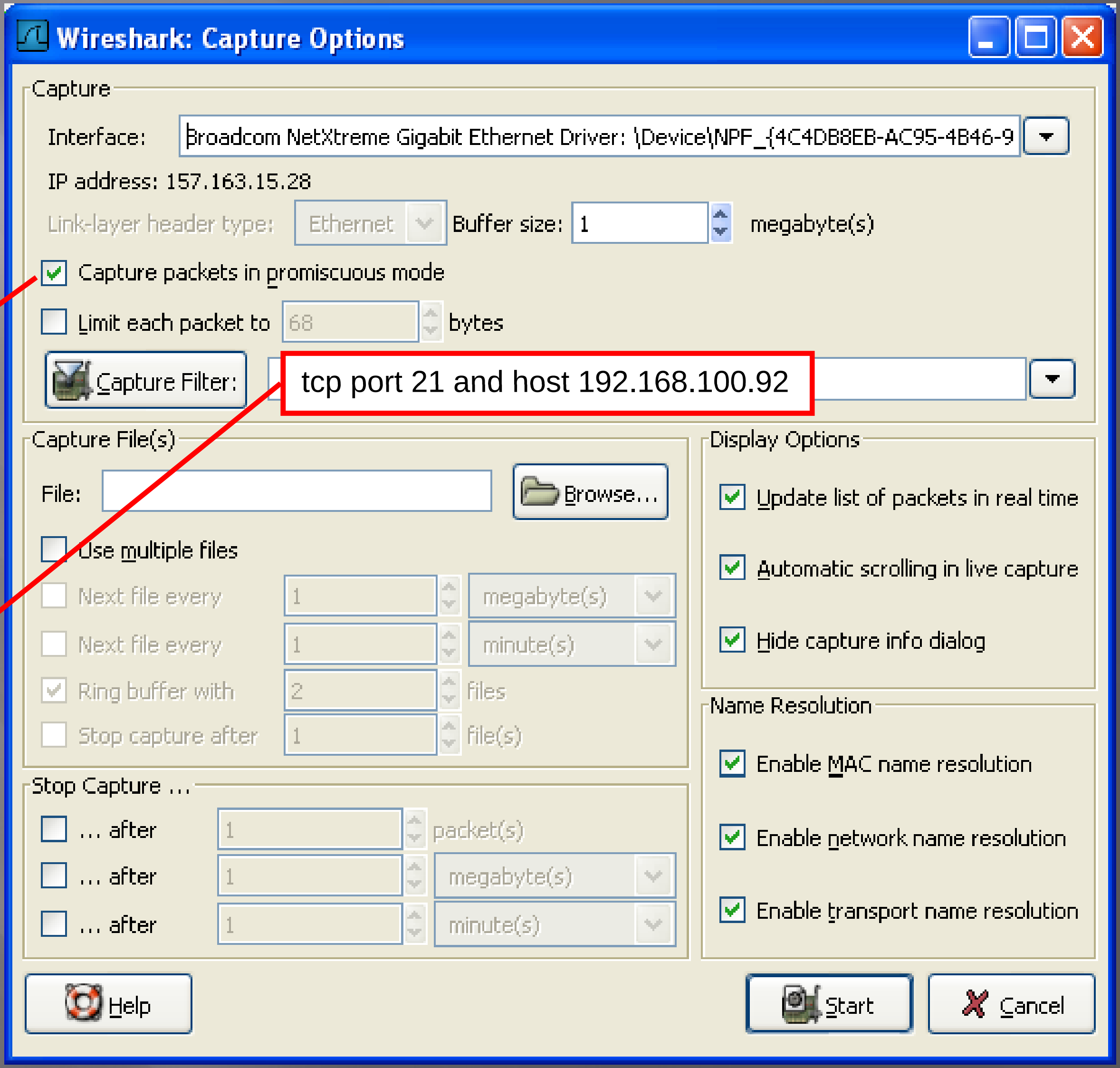
```
No. -  Time                                       Destination        Protocol  Info
    1 0.000000                                    Broadcast          ARP       Who has 192.168.0.2?  Gratuitous A
    2 0.299139     192.168.0.1                    192.168.0.2        NBNS      Name query NBSTAT *<00><00><00><00
    3 0.299214     192.168.0.2                    192.168.0.1        ICMP      Destination unreachable (Port unre
    4 1.025659     192.168.0.2                    224.0.0.22         IGMP      V3 Membership Report
    5 1.044366     192.168.0.2                    192.168.0.1        DNS       Standard query SRV _ldap._tcp.nbgr
    6 1.048652     192.168.0.2                    239.255.255.250    UDP       Source port: 3193  Destination por
    7 1.050784     192.168.0.2                    192.168.0.1        DNS       Standard query SOA nb10061d.ww004.
    8 1.055053     192.168.0.1                    192.168.0.2        UDP       Source port: 1900  Destination por
    9 1.082038     192.168.0.2                    192.168.0.255      NBNS      Registration NB NB10061D<00>
   10 1.111945     192.168.0.2                    192.168.0.1        DNS       Standard query A proxyconf.ww004.s
   11 1.226156     192.168.0.2                    192.168.0.1        TCP       3196 > http [SYN] Seq=0 Len=0 MSS=
   12 1.227282     192.168.0.1                    192.168.0.2        TCP       http > 3196 [SYN, ACK] Seq=0 Ack=1
```

```
⊞ Frame 11 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊟ Transmission Control Protocol, Src Port: 3196 (3196), Dst Port: http (80), Seq: 0, Len: 0
    Source port: 3196 (3196)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 64240
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

File: "D:/test.pcap" 14 KB 00:00:02 P: 120 D: 120 M: 0

# Filtering



Wireshark: Capture Options

Capture

Interface: Broadcom NetXtreme Gigabit Ethernet Driver: \Device\NPF_{4C4DB8EB-AC95-4B46-9

IP address: 157.163.15.28

Link-layer header type: Ethernet   Buffer size: 1 megabyte(s)

☑ Capture packets in promiscuous mode

□ Limit each packet to 68 bytes

Capture Filter: tcp port 21 and host 192.168.100.92

Capture File(s)

File:

□ Use multiple files

□ Next file every 1 megabyte(s)

□ Next file every 1 minute(s)

☑ Ring buffer with 2 files

□ Stop capture after 1 file(s)

Stop Capture ...

□ ... after 1 packet(s)

□ ... after 1 megabyte(s)

□ ... after 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☑ Hide capture info dialog

Name Resolution

☑ Enable MAC name resolution

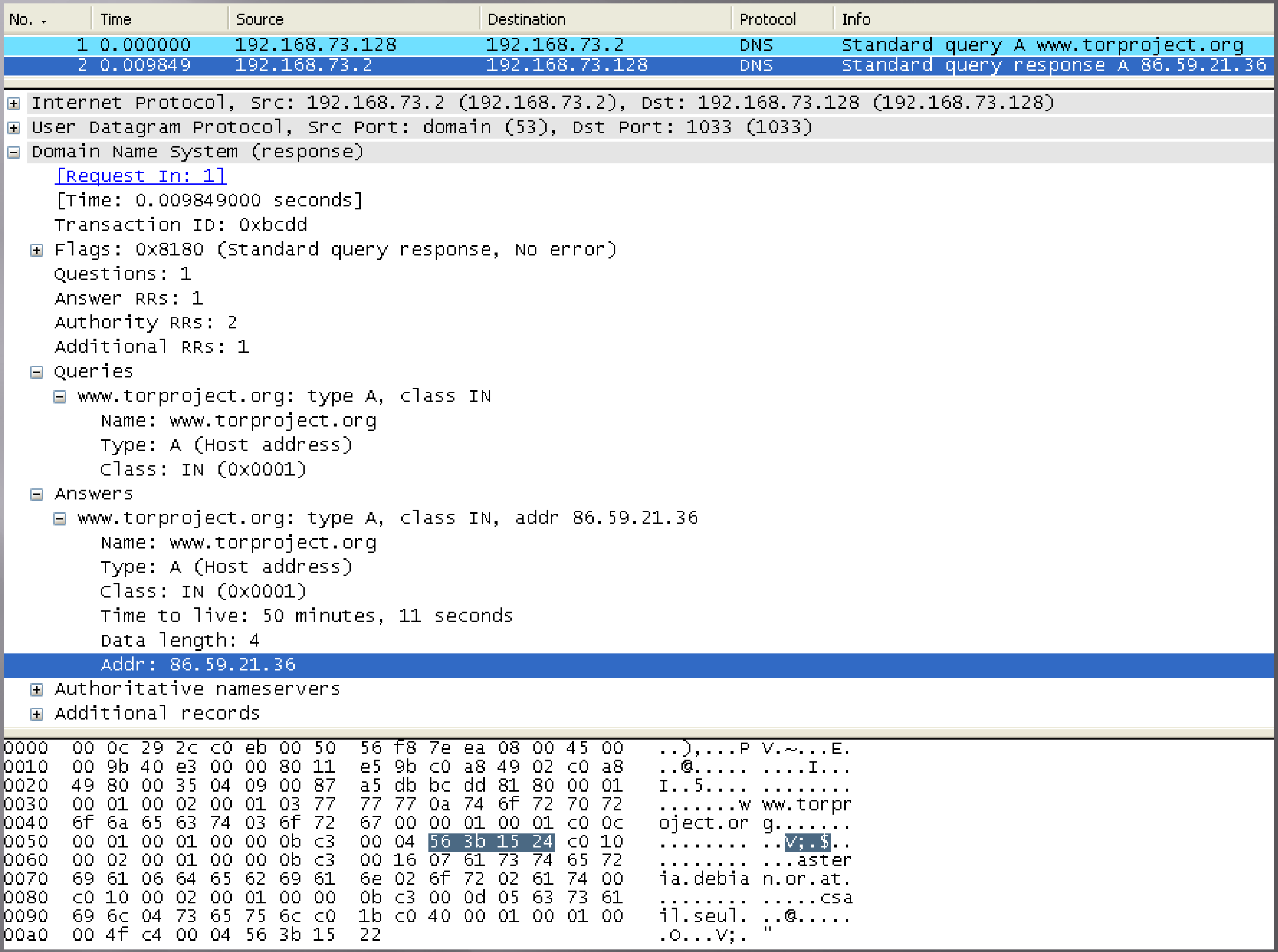
☑ Enable network name resolution

☑ Enable transport name resolution

Capture in promiscuous mode

Filter irrelevant traffic

# Understanding the Protocols



```
No. -     Time          Source             Destination        Protocol   Info
      1 0.000000     192.168.73.128     192.168.73.2        DNS       Standard query A www.torproject.org
      2 0.009849     192.168.73.2       192.168.73.128      DNS       Standard query response A 86.59.21.36
```

```
⊞ Internet Protocol, Src: 192.168.73.2 (192.168.73.2), Dst: 192.168.73.128 (192.168.73.128)
⊞ User Datagram Protocol, Src Port: domain (53), Dst Port: 1033 (1033)
⊟ Domain Name System (response)
    [Request In: 1]
    [Time: 0.009849000 seconds]
    Transaction ID: 0xbcdd
  ⊞ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 2
    Additional RRs: 1
  ⊟ Queries
    ⊟ www.torproject.org: type A, class IN
        Name: www.torproject.org
        Type: A (Host address)
        Class: IN (0x0001)
  ⊟ Answers
    ⊟ www.torproject.org: type A, class IN, addr 86.59.21.36
        Name: www.torproject.org
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 50 minutes, 11 seconds
        Data length: 4
        Addr: 86.59.21.36
  ⊞ Authoritative nameservers
  ⊞ Additional records
```

```
0000  00 0c 29 2c c0 eb 00 50  56 f8 7e ea 08 00 45 00   ..),...P V.~...E.
0010  00 9b 40 e3 00 00 80 11  e5 9b c0 a8 49 02 c0 a8   ..@..... ....I...
0020  49 80 00 35 04 09 00 87  a5 db bc dd 81 80 00 01   I..5.... ........
0030  00 01 00 02 00 01 03 77  77 77 0a 74 6f 72 70 72   .......w ww.torpr
0040  6f 6a 65 63 74 03 6f 72  67 00 00 01 00 01 c0 0c   oject.or g.......
0050  00 01 00 01 00 00 0b c3  00 04 56 3b 15 24 c0 10   ........ ..V;.$..
0060  00 02 00 01 00 00 0b c3  00 16 07 61 73 74 65 72   ........ ...aster
0070  69 61 06 64 65 62 69 61  6e 02 6f 72 02 61 74 00   ia.debia n.or.at.
0080  c0 10 00 02 00 01 00 00  0b c3 00 0d 05 63 73 61   ........ .....csa
0090  69 6c 04 73 65 75 6c c0  1b c0 40 00 01 00 01 00   il.seul. ..@.....
00a0  00 4f c4 00 04 56 3b 15  22                        .O...V;.  "
```

# Programming with pcap

- **pcap_open_live()**
  - Obtain a packet capture descriptor
- **pcap_setfilter()**
  - Specify a filter program
- **pcap_loop()**
  - Allows you to specify a callback function to process the sniffed packets
- **pcap_inject()**
  - Inject packet into the network

# Programming with pcap

```
/* Define the device */
dev = “eth2”;

/* Open the session in promiscuous mode */
handle = pcap_open_live(dev, BUFSIZ, 1, -1, errbuf);

/* Compile and apply the filter */
pcap_compile(handle, &fp, filter_exp, 1, netp);
pcap_setfilter(handle, &fp);

/* Starts sniffing */
pcap_loop(handle, -1 , my_callback, args);
```

# Programming with pcap

```
void my_callback(u_char *args, const struct pcap_pkthdr* pkthdr,
    const u_char *packet)
{
    // everytime a packet is sniffed, this function will be invoked.
    // do your packet processing in this function
    // pkthdr->caplen gives the captured packet length
    // packet is the packet that was captured
    …
    pcap_inject(…)  // send your crafted packet
    …
}
```

# Useful Information

- PCAP Tutorial
  - http://yuba.stanford.edu/~casado/pcap/section1.html
- Get MAC address for an interface in Linux
  - http://stackoverflow.com/questions/1519585/how-to-get-mac-address-for-an-interface-in-linux-using-a-c-program
- IP header checksum
  - http://web.eecs.utk.edu/~cs594np/unp/checksum.html
- C for Java Programmers
  - http://www.cs.vu.nl/~jason/college/dictaat.pdf
- Bitwise operators
  - http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/BitOp/bitwise.html

# Useful Functions

- inet_aton()
  - http://linux.die.net/man/3/inet_aton
- htons(), htonl(), ntohs(), ntohl()
  - http://linux.die.net/man/3/htons

# ARP Spoofing MADNESS

- Many students will be performing ARP spoofing in the lab.
- Buggy code during development
  - Wrong source/destination
  - Malformed ARP packets
  - Infinite loop injecting ARP packets continuously into the network
  - …

# ARP Spoofing MADNESS

- If someone is sending packets that causes problem to your program:
  - Consider changing your program to ignore these “problematic” packets
  - Wait for the person to stop his program before you run your program
  - Politely ask and discuss with the person the problem
    - Discuss and help each other (but do not copy)
- During demo/evaluation, you will be the only attacker. ☺

# Caution!

- Hacking is a serious offence!
- Computer Misuse Act
  - Addresses computer crimes and provides for stiff penalties for the violation of the law.
- Do not run your ARP spoofing or DNS hijack code outside of CS5321 Lab
  - Do not do your homework at your workplace

# Q & A