Assignment 2

CSCI353 Discussion
2-18-2016

simon woo
simonsoo@usc.edu

Topics for Today

- How to use DETER testbed:
  - create a single node experiment
- Basics of tcpdump:
  - how to use
  - wireshark and showing packet content
- how to read a pcap file:
  - code study
- how to read the live interface:
  - code study
- Parse packets and keep track of packet counts.

One node experiment

```
# This is a simple ns script. Comments start with #.
set ns [new Simulator]
source tb_compat.tcl
set nodeA [$ns node]
# Set the OS on a couple.
tb-set-node-os $nodeA Ubuntu1004-STD

$ns rtproto Static

# Go!
$ns run
```

Get Ubuntu Machine from DETER

- Download oneNode.ns from Piazza→ Resources
- **From you local machine to DETER users machine:**

```
$ssh usc353ta@users.isi.deterlab.net
[usc353ta@users ~]$ hostname users.isi.deterlab.net
```

- **From DETER users machine to nodeA**

```
[usc353ta@users ~]$ ssh
nodeA.ass2.usc353.isi.deterlab.net
```

- **Run tcpdump from nodeA**

```
usc353ta@nodea:~$ hostname
nodea.ass2.usc353.isi.deterlab.net
```

Stand on the shoulders of Packet Analyzer: your friend: **tcpdump**

```
• usc353ta@nodea:~$ sudo tcpdump > out
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
```

```
• usc353ta@nodea:~$ cat out
• 17:29:13.230892 IP pc20.isi.deterlab.net.ssh >
  users.isi.deterlab.net.21481: Flags [P.], seq
  1266068218:1266068410, ack 1262857425, win 408, options
  [nop,nop,TS val 82891 ecr 3295448492], length 192
```

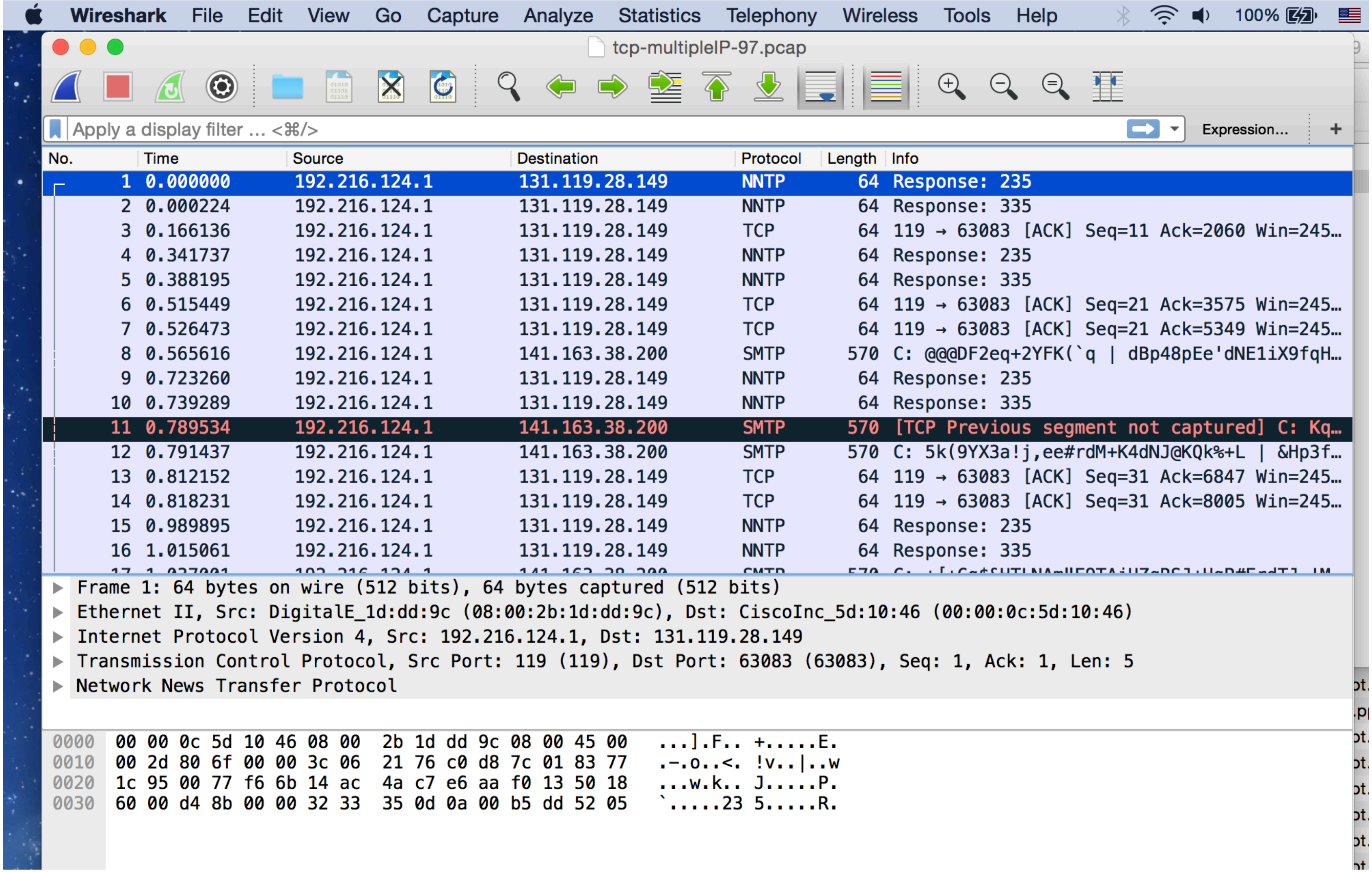
Another friend:

WIRESHARK

https://www.wireshark.org/#download

May the WireShark
Be with you

Load Sample PCAP file into wireshark

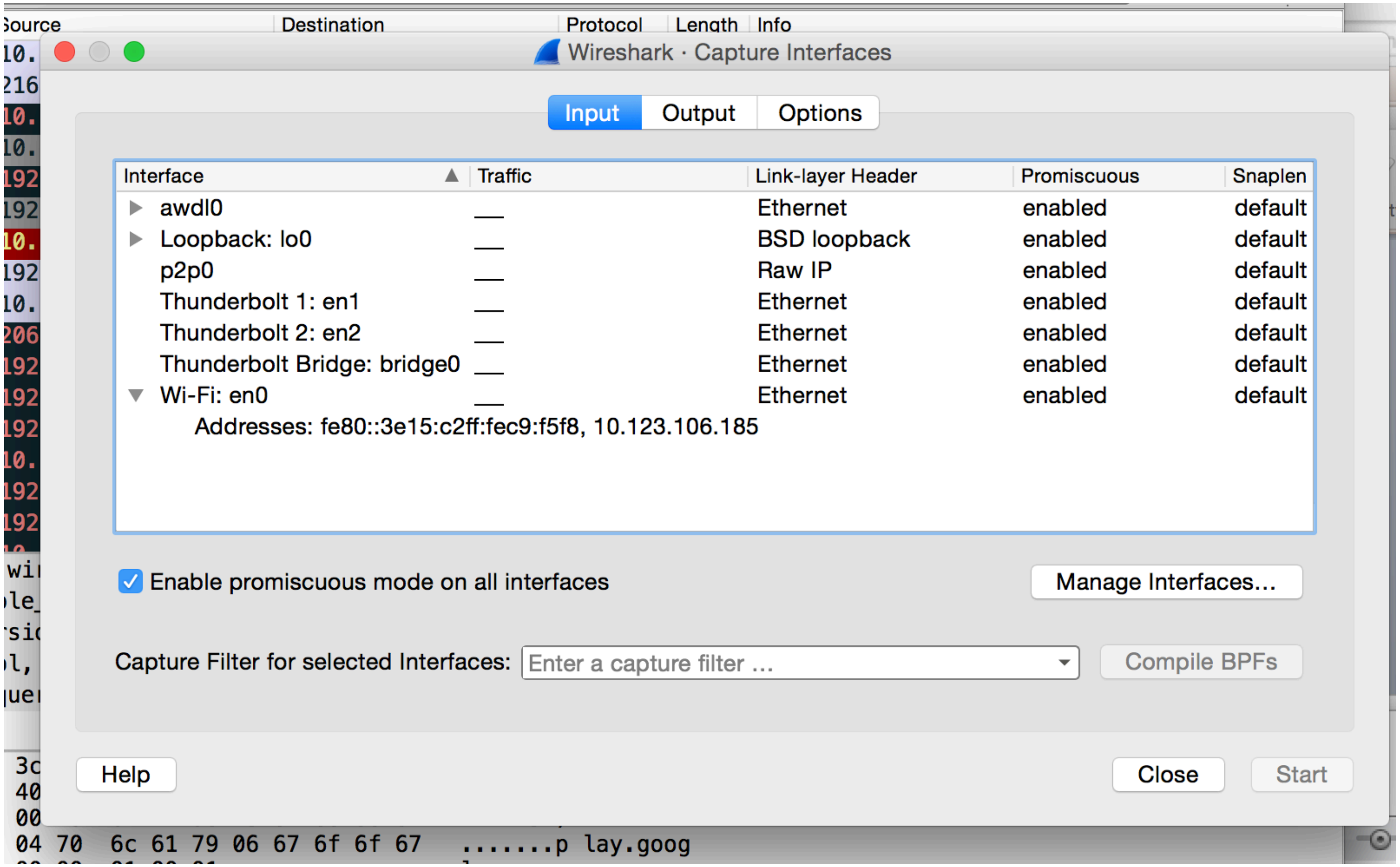


```
No.  Time      Source         Destination     Protocol Length Info
   1 0.000000  192.216.124.1  131.119.28.149  NNTP       64 Response: 235
   2 0.000224  192.216.124.1  131.119.28.149  NNTP       64 Response: 335
   3 0.166136  192.216.124.1  131.119.28.149  TCP        64 119 → 63083 [ACK] Seq=11 Ack=2060 Win=245…
   4 0.341737  192.216.124.1  131.119.28.149  NNTP       64 Response: 235
   5 0.388195  192.216.124.1  131.119.28.149  NNTP       64 Response: 335
   6 0.515449  192.216.124.1  131.119.28.149  TCP        64 119 → 63083 [ACK] Seq=21 Ack=3575 Win=245…
   7 0.526473  192.216.124.1  131.119.28.149  TCP        64 119 → 63083 [ACK] Seq=21 Ack=5349 Win=245…
   8 0.565616  192.216.124.1  141.163.38.200  SMTP      570 C: @@@DF2eq+2YFK(`q | dBp48pEe'dNE1iX9fqH…
   9 0.723260  192.216.124.1  131.119.28.149  NNTP       64 Response: 235
  10 0.739289  192.216.124.1  131.119.28.149  NNTP       64 Response: 335
  11 0.789534  192.216.124.1  141.163.38.200  SMTP      570 [TCP Previous segment not captured] C: Kq…
  12 0.791437  192.216.124.1  141.163.38.200  SMTP      570 C: 5k(9YX3a!j,ee#rdM+K4dNJ@KQk%+L | &Hp3f…
  13 0.812152  192.216.124.1  131.119.28.149  TCP        64 119 → 63083 [ACK] Seq=31 Ack=6847 Win=245…
  14 0.818231  192.216.124.1  131.119.28.149  TCP        64 119 → 63083 [ACK] Seq=31 Ack=8005 Win=245…
  15 0.989895  192.216.124.1  131.119.28.149  NNTP       64 Response: 235
  16 1.015061  192.216.124.1  131.119.28.149  NNTP       64 Response: 335
```

```
▶ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
▶ Ethernet II, Src: DigitalE_1d:dd:9c (08:00:2b:1d:dd:9c), Dst: CiscoInc_5d:10:46 (00:00:0c:5d:10:46)
▶ Internet Protocol Version 4, Src: 192.216.124.1, Dst: 131.119.28.149
▶ Transmission Control Protocol, Src Port: 119 (119), Dst Port: 63083 (63083), Seq: 1, Ack: 1, Len: 5
▶ Network News Transfer Protocol
```

```
0000  00 00 0c 5d 10 46 08 00  2b 1d dd 9c 08 00 45 00   ...].F.. +.....E.
0010  00 2d 80 6f 00 00 3c 06  21 76 c0 d8 7c 01 83 77   .-.o..<. !v..|..w
0020  1c 95 00 77 f6 6b 14 ac  4a c7 e6 aa f0 13 50 18   ...w.k.. J.....P.
0030  60 00 d4 8b 00 00 32 33  35 0d 0a 00 b5 dd 52 05   `.....23 5.....R.
```

Capture Live Traffic from wireshark



Interface	Traffic	Link-layer Header	Promiscuous	Snaplen
▶ awdl0		Ethernet	enabled	default
▶ Loopback: lo0		BSD loopback	enabled	default
p2p0		Raw IP	enabled	default
Thunderbolt 1: en1		Ethernet	enabled	default
Thunderbolt 2: en2		Ethernet	enabled	default
Thunderbolt Bridge: bridge0		Ethernet	enabled	default
▼ Wi-Fi: en0		Ethernet	enabled	default
Addresses: fe80::3e15:c2ff:fec9:f5f8, 10.123.106.185				

☑ Enable promiscuous mode on all interfaces

Capture Filter for selected Interfaces: Enter a capture filter …

```
04 70  6c 61 79 06 67 6f 6f 67   .......p lay.goog
```

More example on tcpdump(2)

Only Read Specific Protocol:

```
>>sudo tcpdump -r  tcp-oneIP-200.pcap  -ntttt tcp
```

Specify the packet size:

```
>> sudo tcpdump -r  tcp-oneIP-200.pcap  -ntttt greater 1024
```

More example on tcpdump(2)

- **Specify Src/Dst**

```
>>sudo tcpdump -r  tcp-oneIP-200.pcap  -ntttt src 10.1.1.2
```

```
>> sudo tcpdump -r  tcp-oneIP-200.pcap  -ntttt dst 10.1.2.3
```

Or both

More example on tcpdump(3)

- **Specify Src/Dst**

```
>>sudo tcpdump -r  tcp-oneIP-200.pcap  -ntttt src 10.1.1.2
```

```
>> sudo tcpdump -r  tcp-oneIP-200.pcap  -ntttt dst 10.1.2.3
```

Or both

More example on tcpdump(4)

Display unique src/dst pairs with count (Just copy and paste)

```
>> sudo tcpdump -r  tcp-multipleIP-193.pcap -n -c 5 ip |
awk '{ print gensub(/(.*)\..*/,"\\1","g",$3), $4, gensub(/
(.*)\..*/,"\\1","g",$5) }'
```

```
    7 192.216.124.1 131.119.28.149
    1 192.216.124.1 141.163.38.200
    2 192.216.124.1 131.119.28.149
    2 192.216.124.1 141.163.38.200
    4 192.216.124.1 131.119.28.149
```

Tcpdump: Live packet capture

```
>> sudo tcpdump -i eth1 > out
```

C/C++ code with pcap file

- **Programming with pcap**

http://www.tcpdump.org/pcap.html

- **Get packet sniffer sample for live traffic capture**

  http://www.tcpdump.org/sniffex.c

Example: sniffex.c

Compile:

```
usc353ta@nodea:~$ gcc -lpcap sniffex.c -o sniffex
```

Execute:

```
usc353ta@nodea:~$ ./sniffex
usc353ta@nodea:~$ ./sniffex eth1
usc353ta@nodea:~$ sudo ./sniffex eth1
```

UDP Packet Parser Example

- **Download**

https://www.google.com/url?
sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=0
ahUKEwjZi4ej8YLLAhVU2WMKHQyBDyMQFghXMA
k&url=http%3A%2F%2Finst.eecs.berkeley.edu
%2F~ee122%2Ffa07%2Fprojects%2Fp2files
%2Fpacket_parser.c&usg=AFQjCNEtfq303v8Ia_otW
-lHhfa0riEKgA&sig2=9xsDHZqoAmcCwFF9VzceSw

UDP Packet Parser

Compile

```
usc353ta@nodea:~$ gcc -lpcap pparser.c -o pparser
```

Execute

```
usc353ta@nodea:~$ ./pparser
program requires one argument, the trace file to dump

usc353ta@nodea:~$ ./pparser udp-multipleIP-7.pcap
874057320.532083 UDP src_port=1719 dst_port=53 length=39
874057322.693869 UDP src_port=1732 dst_port=53 length=39
874057322.853588 UDP src_port=53 dst_port=53 length=39
874057323.958002 UDP src_port=53 dst_port=53 length=39
874057327.841628 UDP src_port=1725 dst_port=53 length=39
```

tcpdump tutorial

- http://www.thegeekstuff.com/2010/08/tcpdump-command-examples/
-
- http://stackoverflow.com/questions/681011/getting-the-number-of-packets-in-a-pcap-capture-file
-
- http://man.he.net/man8/tcpdump
-
- https://danielmiessler.com/study/tcpdump/

pcap programing w/ c

- http://www.tcpdump.org/pcap.html