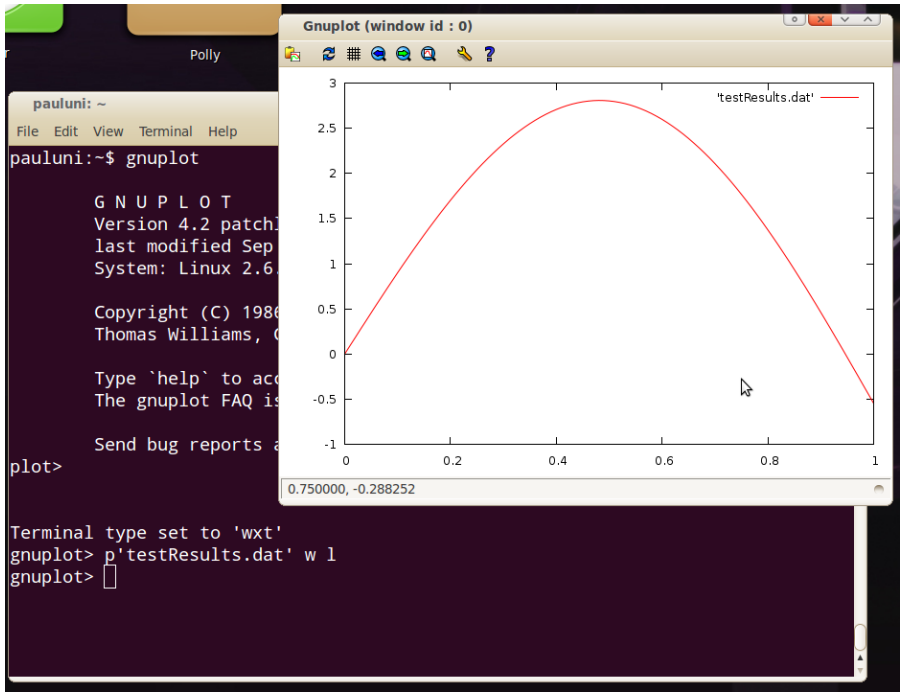
GNUplot & PCAP

Discussion Section – May 22

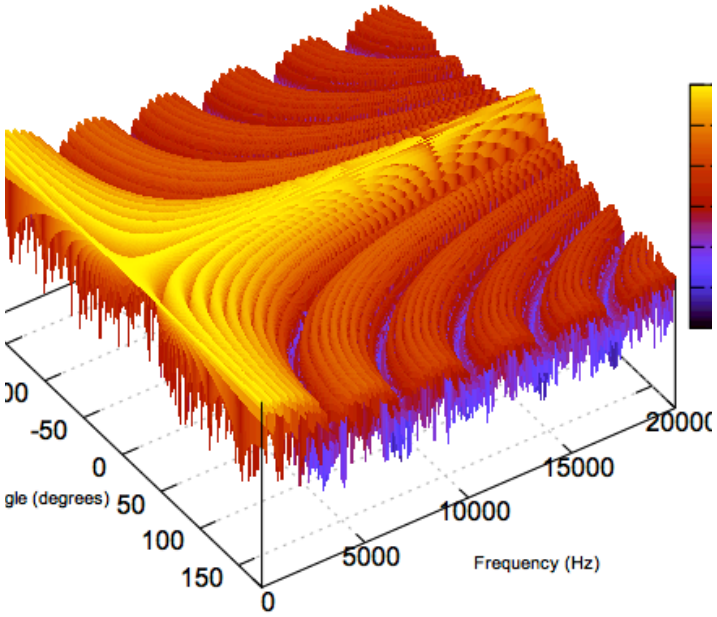
What is GNUplot?

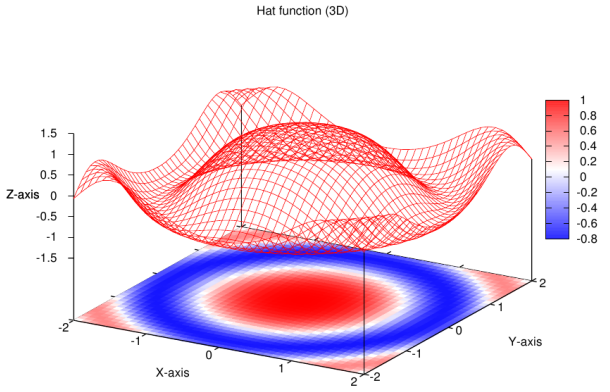
- It's an interactive plotting environment
- Can be directly used from Terminal (popular method)
- Or can be imported as a module in Python

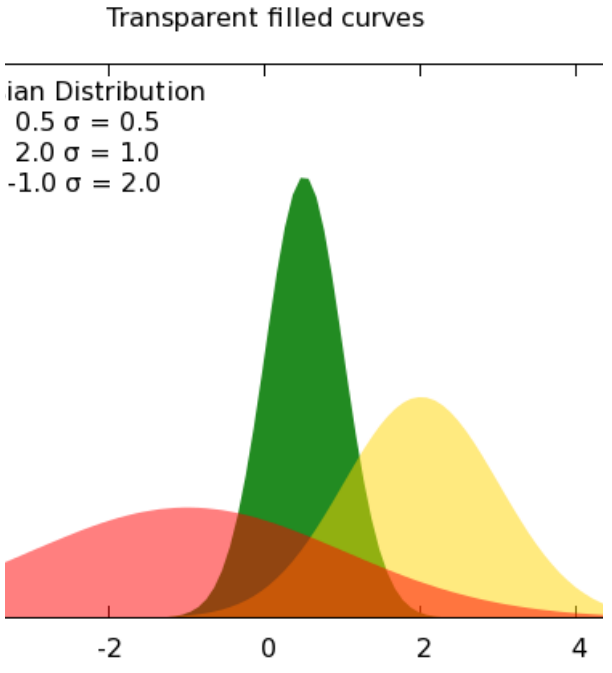


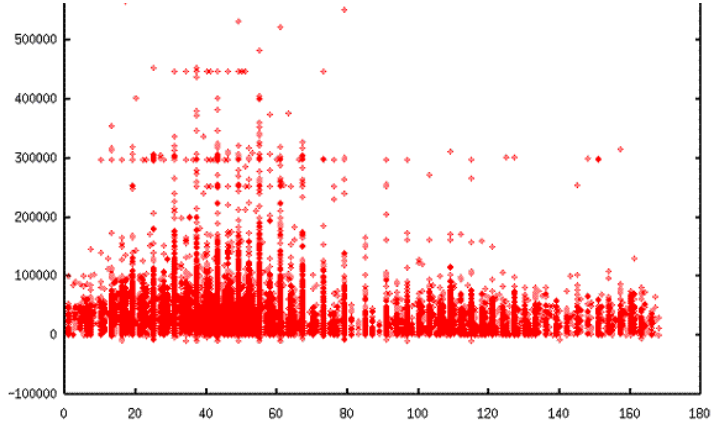
What can you do with GNUplot?

> Graph these beautiful plots!









Basic Features

- Provides math, string and time functions
- Provides block structure if/while/do
- Can select a column of data from a data file by matching a label, or by index.
- Installing GNUplot:
- Windows get .exe from sourceforge.net/projects/gnuplot/files
- Mac get .tar.gz from sourceforge.net/projects/gnuplot/files
- open shell, go to download directory, type ”configure”, ”make”, ”sudo make install”
- Linux use package management system
- To start, type “gnuplot” in terminal/cmd
- To quit, type “quit”

Set the terminal type

gnuplot> set terminal postscript
gnuplot> set terminal png
gnuplot> set output "plot.png"

```
[ResNet-10-180:~ vivekadarsh$ gnuplot

        G N U P L O T
        Version 5.0 patchlevel 6    last modified 2017-03-18

        Copyright (C) 1986-1993, 1998, 2004, 2007-2017
        Thomas Williams, Colin Kelley and many others

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type set to 'unknown'
[gnuplot> set terminal

Available terminal types:
          canvas  HTML Canvas object
             cgm  Computer Graphics Metafile
         context  ConTeXt with MetaFun (for PDF documents)
           corel  EPS format for CorelDRAW
         domterm  DomTerm terminal emulator with embedded SVG
            dumb  ascii art for anything that prints text
             dxf  dxf-file for AutoCad (default size 120x80)
           eepic  EEPIC -- extended LaTeX picture environment
             emf  Enhanced Metafile format
           emtex  LaTeX picture environment with emTeX specials
        epslatex  LaTeX picture environment using graphicx package
             fig  FIG graphics language for XFIG graphics editor
             gif  GIF images using libgd and TrueType fonts
            hpgl  HP7475 and relatives [number of pens] [eject]
            jpeg  JPEG images using libgd and TrueType fonts
           latex  LaTeX picture environment
             lua  Lua generic terminal driver
              mf  Metafont plotting standard
              mp  MetaPost plotting standard
            pcl5  HP Designjet 750C, HP Laserjet III/IV, etc. (many options)
             png  PNG images using libgd and TrueType fonts
Press return for more:
      postscript  PostScript graphics, including EPSF embedded files (*.eps)
         pslatex  LaTeX picture environment with PostScript \specials
           pstex  plain TeX with PostScript \specials
        pstricks  LaTeX picture environment with PSTricks macros
             qms  QMS/QUIC Laser printer (also Talaris 1200 and others)
             svg  W3C Scalable Vector Graphics
          tek40xx  Tektronix 4010 and others; most TEK emulators
          tek410x  Tektronix 4106, 4107, 4109 and 420X terminals
          texdraw  LaTeX texdraw environment
             tgif  TGIF X11 [mode] [x,y] [dashed] ["font" [fontsize]]
             tikz  TeX TikZ graphics macros via the lua script driver
         tkcanvas  Tk canvas widget
             tpic  TPIC -- LaTeX picture environment with tpic \specials
          unknown  Unknown terminal type - not a plotting device
            vttek  VT-like tek40xx terminal emulator
            xterm  Xterm Tektronix 4014 Mode
gnuplot>
```

Function	Returns
abs(x)	absolute value of x, |x|
acos(x)	arc-cosine of x
asin(x)	arc-sine of x
atan(x)	arc-tangent of x
cos(x)	cosine of x, x is in radians.
cosh(x)	hyperbolic cosine of x, x is in radians
erf(x)	error function of x
exp(x)	exponential function of x, base e
inverf(x)	inverse error function of x
invnorm(x)	inverse normal distribution of x
log(x)	log of x, base e
log10(x)	log of x, base 10
norm(x)	normal Gaussian distribution function
rand(x)	pseudo-random number generator
sgn(x)	1 if x > 0, -1 if x < 0, 0 if x=0
sin(x)	sine of x, x is in radians
sinh(x)	hyperbolic sine of x, x is in radians
sqrt(x)	the square root of x
tan(x)	tangent of x, x is in radians
tanh(x)	hyperbolic tangent of x, x is in radians

Supported functions

Try this syntax out:

gnuplot> plot (sin(x))

gnuplot> splot sin(x*y/20)

gnuplot> plot sin(x) title 'Sine Function', tan(x) title 'Tangent'

Plotting Data

```
Syntax:
       plot {[ranges]}
            {[function] | {"[datafile]" {datafile-modifiers}}}
            {axes [axes] } { [title-spec] } {with [style] }
            {, {definitions,} [function] ...}
```

- gnuplot> plot "example1.dat" using 1:2 title 'Column', \
  "example2.dat" using 1:3 title 'Beam'

```
Create a title:                    > set title "Force-Deflection Data"
Put a label on the x-axis:         > set xlabel "Deflection (meters)"
Put a label on the y-axis:         > set ylabel "Force (kN)"
Change the x-axis range:           > set xrange [0.001:0.005]
Change the y-axis range:           > set yrange [20:500]

Have Gnuplot determine ranges:     > set autoscale
Move the key:                      > set key at 0.01,100
Delete the key:                    > unset key
Put a label on the plot:           > set label "yield point" at 0.003,

Remove all labels:                 > unset label
Plot using log-axes:               > set logscale
Plot using log-axes on y-axis:     > unset logscale; set logscale y
Change the tic-marks:              > set xtics (0.002,0.004,0.006,0.008)
Return to the default tics:        > unset xtics; set xtics auto
```

Input *.dat file

```
# Force-Deflection data for a beam and a bar
# Deflection    Col-Force       Beam-Force
0.000             0               0
0.001           104              51
0.002           202             101
0.003           298             148
0.0031          290             149
0.004           289             201
0.0041          291             209
0.005           310             250
0.010           311             260
0.020           280             240
```

Packet Capturing

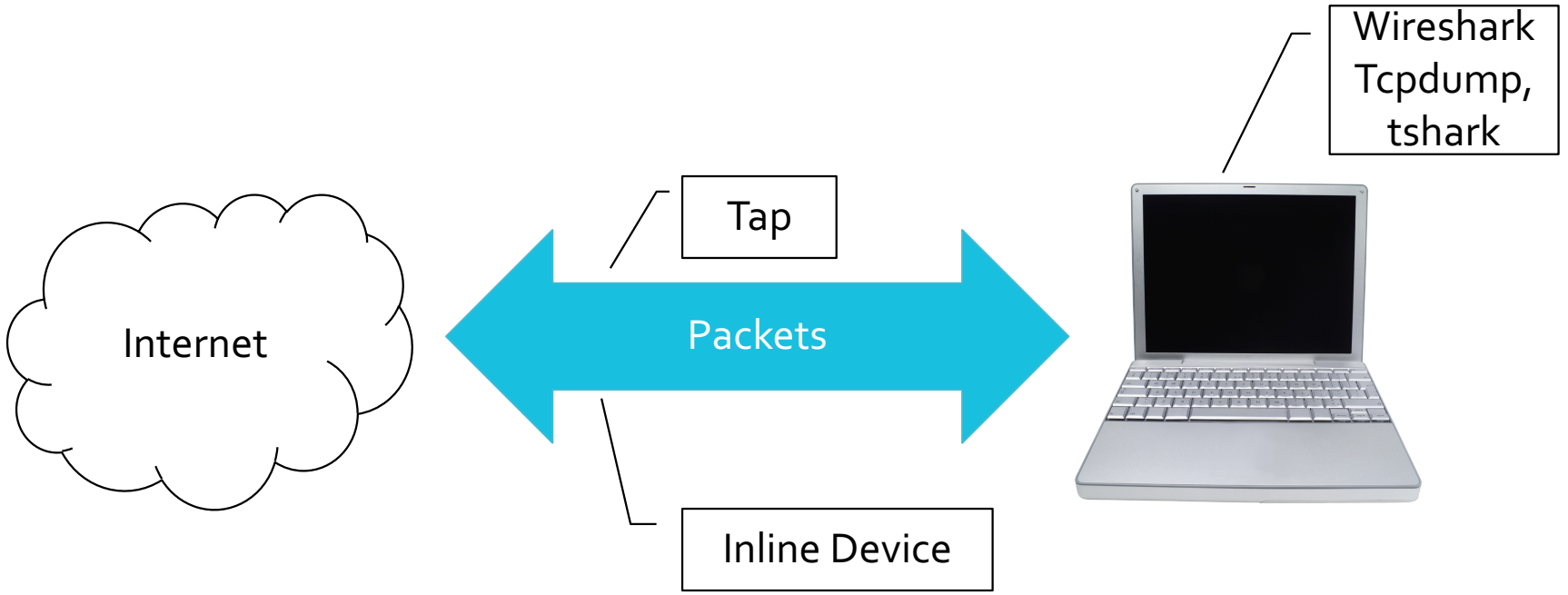
- PCAP == **P**acket **Cap**ture
- Complete record of network activity
  - Layers 2 – 7
- Most common format is `libpcap`
  - Open-source
  - Available on Unix and Windows
  - C library, bindings in many languages
  - Others proprietary formats not covered

Who Uses PCAP?

- **Researchers**: access to raw data
- **Administrators**: debug network problems
- **Analysts**: characterize malware activity
- **Incident Responders**: follow malware

- What would **you** use it for?

Collecting PCAP



Demo time!
Wireshark & tcpdump

Useful Resources

- Goto gnuplot homepage: http://www.gnuplot.info
- Goto gnuplot demos: http://www.gnuplot.info/screenshots/index.html#demos
- Goto gnuplot tutorial: http://www.gnuplot.info/help.html
- Gnuplot documentation (Most useful!): http://www.gnuplot.info/docs_5.0/gnuplot.pdf
- Some of the content have been taken from http://www.usm.uni-muenchen.de/people/puls/lessons/intro_general/gnuplot/gnuplot_for_beginners.pdf
- And lastly, YouTube is your friend! There are tons of tutorials available for GNUplot