Lesson Module Status

- Wall updated and emailed

- Slides –
- Properties -
- Flashcards -
- 1st minute quiz –
- Web Calendar summary –
- Web book pages –
- Commands –
- Howtos –

- Lab tested –
- Lab template in depot -
- Youtube Videos uploaded –

- VM (Classroom PC) –
- VMs (VLab) - extra gondor and arnor switches made for each pod

- Headset charged –

- Exam prep published -

[ ] Has the phone bridge been added?

[ ] Is phone being used for voice input?

[ ] Is recording on?

[ ] Share slides, multiple Putties started, Chrome, vlab192.rdp, VMware Workstation, wireshark

[ ] Disable spelling on PowerPoint

[ ] Repeat all ?'s for remote students

[ ] Remote student proxy

Course history and credits

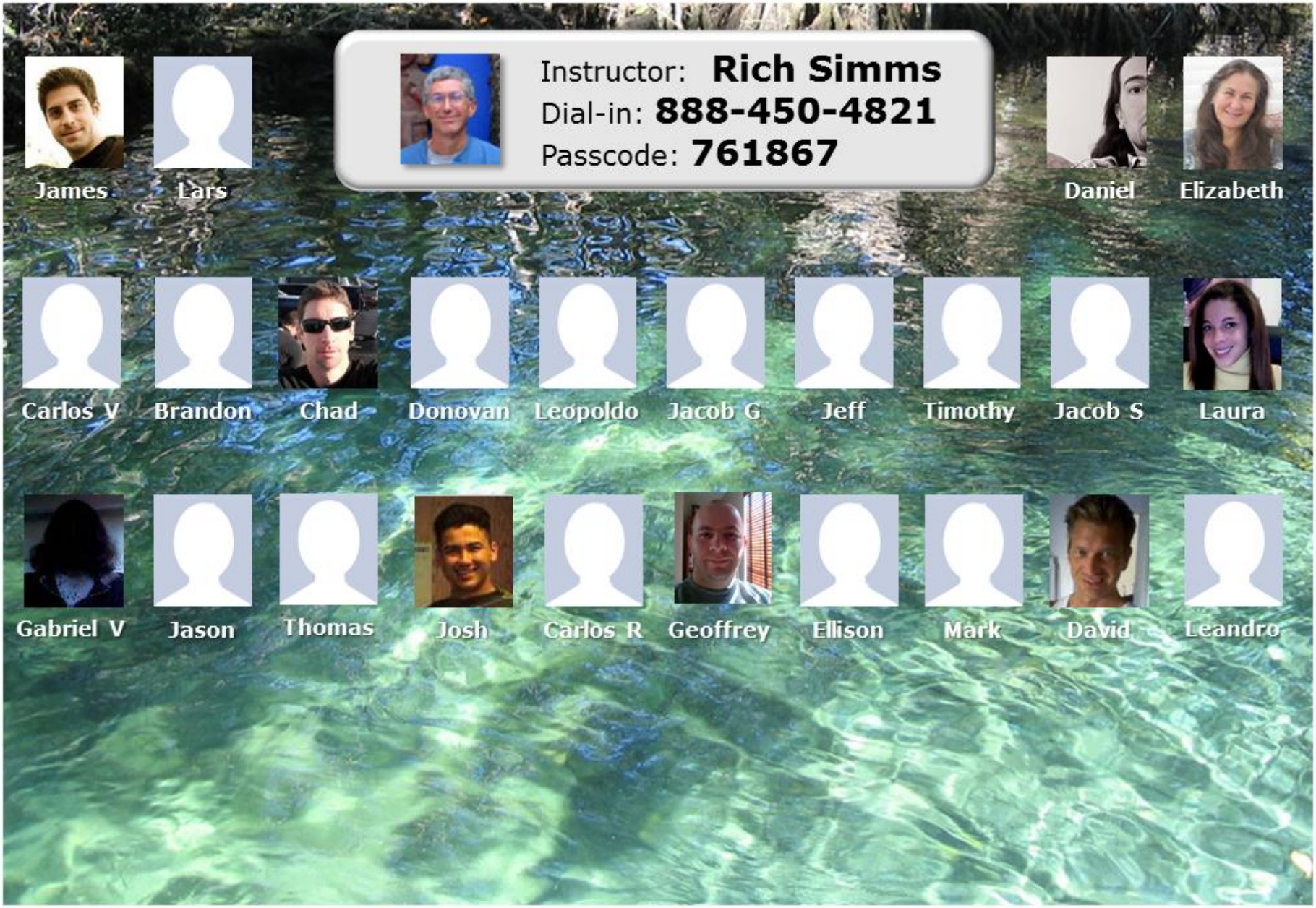
Jim Griffin

- Jim created the original version of this course
- Jim's site: http://cabrillo.edu/~jgriffin/

Rick Graziani

- Thanks to Rick Graziani for the use of some of his great network slides
- Rick's site: http://cabrillo.edu/~rgraziani/

CIS 192 – Lesson 7



Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

First Minute Quiz

Please answer these questions **in the order** shown:

email answers to: `risimms@cabrillo.edu`
within the first few minutes of class

PPP and WAN protocols

Objectives	Agenda
• Connect two computers on a serial line. • Connect two LANs together through a serial line using Point to Point protocol.	• Quiz • Questions on previous material • Housekeeping • Skills practice • SSH tunneling • TCP Wrappers • PPP • PPP Lab prep • Exam prep • Wrap

VMs for tonight
(Revert,and power up)
frodo arwen elrond

Questions on previous material

Questions?

- Previous lesson material
- Lab assignment

Housekeeping

- DHCP Lab 6 due today
- Five posts due next week
- Extra credit labs due next week
- Final Exam next week

- Wind and Layer 1 struck this past week!

- Advisory council un-prioritized requests for courses not offered:
  - Virtualization
  - Project management
  - Mobile
  - More database

aragorn	224
arwen	212
bombadil	261
denethor	176
dwalin	178
elrohir	208
elrond	234
eomer	194
faramir	136
frodo	224
gimli	184
goldberry	144
gwaihir	183
ioreth	186
legolas	251
nazgul	137
pippin	170
samwise	162
saruman	199
strider	184
theoden	217
treebeard	160

Grades Check

(as of 12/6/2011)

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	293 or higher	A	pass
80% to 89.9%	260 to 292	B	pass
70% to 79.9%	228 to 259	C	pass
60% to 69.9%	195 to 227	D	no pass
0% to 59.9%	0 to 194	F	no pass

Remaining Points to earn

Lab 6 = 30 points
Final Exam = 60 points
Forum 2 = 20 points
Quiz 5 = 3 points

} 113 points

Extra credit maximum = 60 points

Crib Sheet Shakeout

Crib Sheet Shakeout

Login
Flashcards
Admin

CIS 192A
Previous Classes

11 days till term ends!

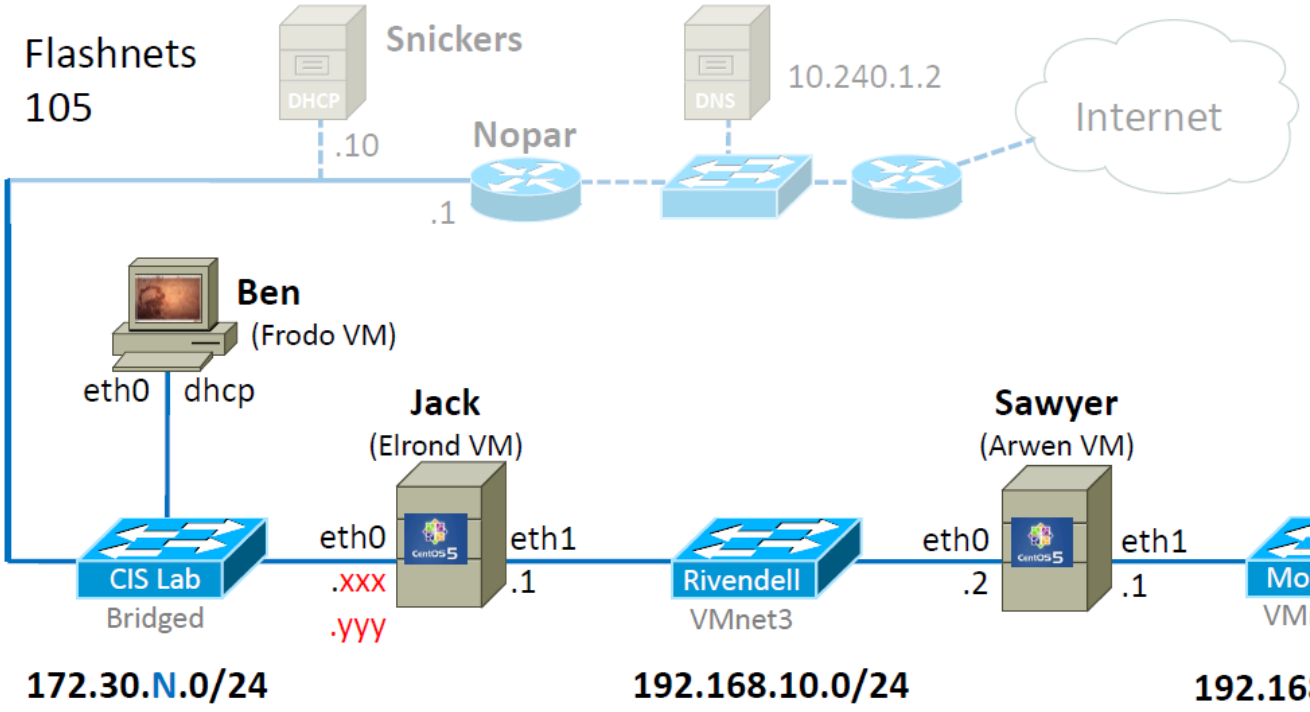
Cabrillo College
Web Advisor
Static IPs
Commands and Files
Accessing VLab

RIP Dennis Ritchie

Linux Network Commands & Files

Click on the link in the table below to see commands, configuration files and examples.

Virtual Cabling	
VMware Cabling	
Joining a Network	
Showing and Controlling Interfaces Show and Control Routes	
IPCalc - to calculate netmasks and more	
Temporary Interface Configuration Using DHCP Temporary Interface Configuration Using Static IP addresses	
Temporary Route configuration	
redhat Permanent Interface Configuration Permanent Routing Table Configuration Permanent Hostname Configuration	debian Permanent Interface Configuration Permanent Hostname Configuration
Name Resolution	
Connectivity Testing	
Making Routers	
	Packet Forwarding
Firewalls and NAT	
Firewalls Firewalls (Red Hat Family)	NAT Favorites
Network Services	
Telnet	FTP
Other	
General Linux commands - root & shutdown General Linux commands - basic inventory Installing more commands	Packet Sniffing SELinux
ARP commands	Linux hardware and driver commands

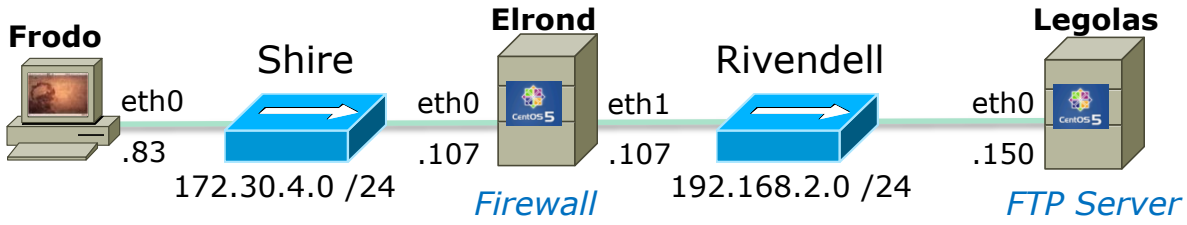


Skills
A. Snapshots
B. Cabling
C. IP Addressing
D. Interfaces
E. Routes
F. Names
G. Forwarding
H. Firewall
I. SELinux
J. Applications
K. NAT
L. tcpdump
M. Wireshark

Classroom Consultant Teams
1. Tim, Jacob S
2. Ellison, Dave
3. Brandon, Chad, Leo
4. Carlos R, Jason, Josh
5. Jacob G, Jeff

Online Consultant Teams
1. Carlos V, Laura, Gabriel
2. Geoffrey, Daniel, Leandro
3. Lars, Elizabeth, James, Mark

Wireshark Socket Spotting



Socket for commands

Client	Server
172.30.4.83	192.168.2.150
42855	21

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
42571	20

***Active Mode** is when server initiates new connection for data transfer*

```
ftp> get legolas
local: legolas remote: legolas
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for legolas (18 bytes).
226 File send OK.
18 bytes received in 0.04 secs (0.5 kB/s)
```

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Wi
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [ACK] Se
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, AC
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=82 Ack=263 Win=5856 Len=0

PORT command to listen on 166, 75 = A64B = 42571

Retrieve legolas file

3 way handshake initiated by server

File transfer

4 way handshake to close connection

Wireshark Columns

Demo

sshd

sshd

The SSH server

- openssh-server package

- Red Hat Family
  - Installed by default
  - Use **rpm –qa | grep openssh-server** to check if installed
- Ubuntu
  - Not installed by default
  - Use **dpkg –l | grep openssh-server** to check if installed

sshd

Installation on Ubuntu

```
[root@sauron ~]# apt-get update
[root@sauron ~]# apt-get install openssh-server
```

Install using aptitude or apt-get

sshd

Installation on Ubuntu

```
root@sauron:~# aptitude update
Get:1 http://security.ubuntu.com intrepid-security Release.gpg [189B]
Ign http://security.ubuntu.com intrepid-security/main Translation-en_US
Hit http://us.archive.ubuntu.com intrepid Release.gpg
Ign http://us.archive.ubuntu.com intrepid/main Translation-en_US
Ign http://security.ubuntu.com intrepid-security/restricted Translation-en_US
Ign http://security.ubuntu.com intrepid-security/universe Translation-en_US
Ign http://security.ubuntu.com intrepid-security/multiverse Translation-en_US
Get:2 http://security.ubuntu.com intrepid-security Release [51.2kB]
Ign http://us.archive.ubuntu.com intrepid/restricted Translation-en_US
Ign http://us.archive.ubuntu.com intrepid/universe Translation-en_US

< snipped >

Get:20 http://us.archive.ubuntu.com intrepid-updates/multiverse Sources [4118B]
Fetched 784kB in 8s (93.5kB/s)
Reading package lists... Done

Current status: 270 updates [+55], 24979 new [+12].
root@sauron:~#
```

sshd

Installation on Ubuntu

```
root@sauron:~# aptitude install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
The following NEW packages will be installed:
  openssh-server
0 packages upgraded, 1 newly installed, 0 to remove and 270 not upgraded.
Need to get 285kB of archives. After unpacking 782kB will be used.
Writing extended state information... Done
Get:1 http://us.archive.ubuntu.com intrepid/main openssh-server 1:5.1p1-3ubuntu1 [285kB]
Fetched 285kB in 2s (99.3kB/s)
Preconfiguring packages ...
Selecting previously deselected package openssh-server.
(Reading database ... 102936 files and directories currently installed.)
Unpacking openssh-server (from .../openssh-server_1%3a5.1p1-3ubuntu1_i386.deb) ...
Processing triggers for ufw ...
Processing triggers for man-db ...
Setting up openssh-server (1:5.1p1-3ubuntu1) ...
 * Restarting OpenBSD Secure Shell server sshd                          [ OK ]

Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
Writing extended state information... Done

root@sauron:~#
```

sshd

Daemon control on Ubuntu

```
root@sauron:~# /etc/init.d/ssh status
 * sshd is running.

root@sauron:~# /etc/init.d/ssh stop
 * Stopping OpenBSD Secure Shell server sshd                    [ OK ]

root@sauron:~# /etc/init.d/ssh start
 * Starting OpenBSD Secure Shell server sshd                    [ OK ]
```

sshd

Daemon control on Red Hat family

```
[root@arwen ~]# service sshd status
sshd (pid 4805) is running...


[root@arwen ~]# service sshd stop
Stopping sshd:                                             [   OK   ]

[root@arwen ~]# service sshd start
Starting sshd:                                             [   OK   ]
```

Firewall for sshd

CentOS Modified

```
[root@legolas ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.3.5 on Thu Feb 26 04:33:47 2009
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2883:272960]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --dport 520 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Thu Feb 26 04:33:47 2009
[root@legolas ~]#
```

New connections for the SSH port are allowed

sshd

Using netstat to view listening ssh ports

```
root@sauron:~# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN

root@sauron:~# netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
root@sauron:~#
```

sshd

One SSH daemon per session

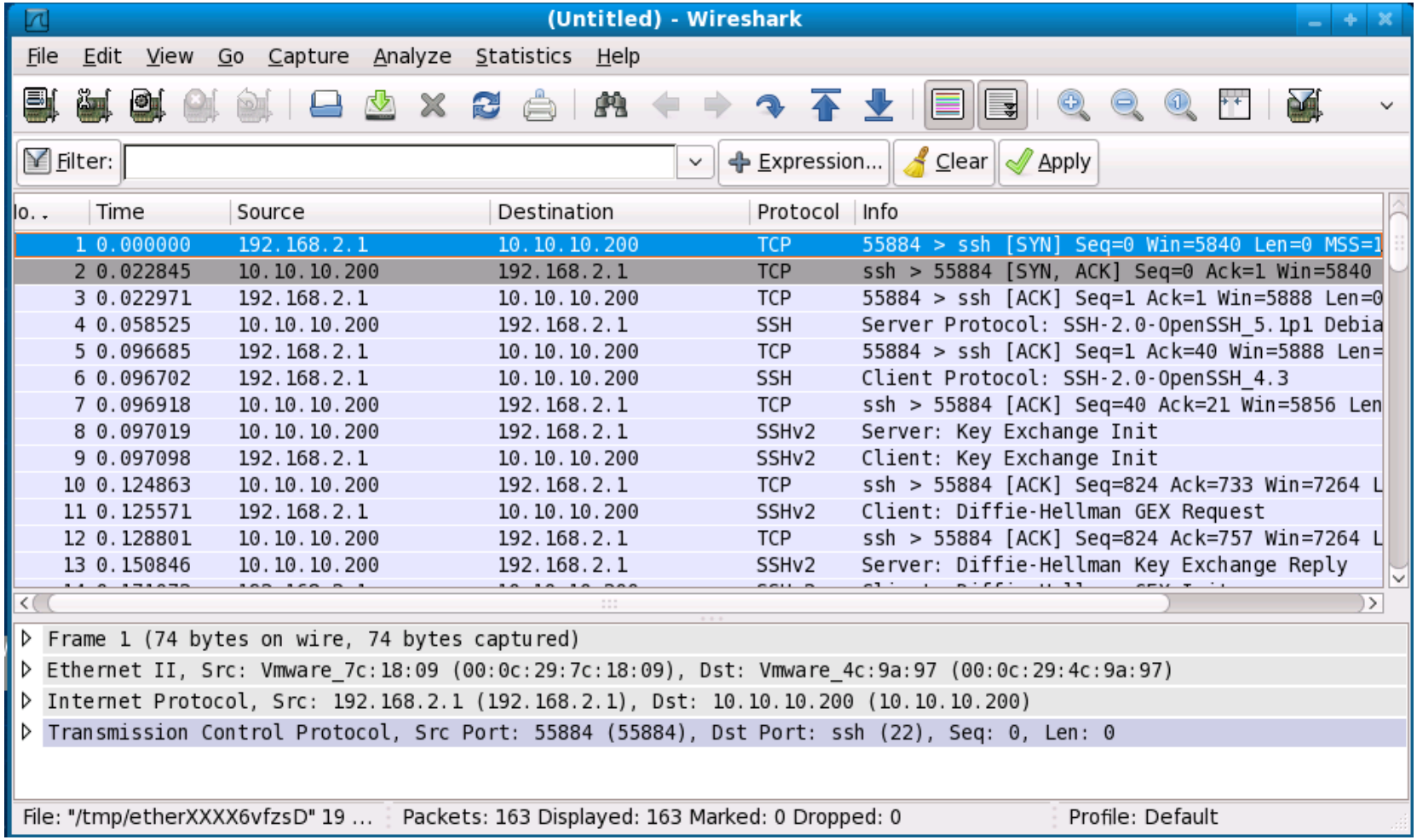
```
root@sauron:~# ps -ef | grep ssh
root      7601     1  0 13:59 ?        00:00:00 /usr/sbin/sshd
root      7607  7601  1 14:11 ?        00:00:00 sshd: root@pts/2
root      7632  7601  1 14:11 ?        00:00:00 sshd: root@pts/3
root      7658  7280  0 14:12 pts/1    00:00:00 grep ssh


root@sauron:~# who
root     tty2         2009-03-13 14:32
cis192   tty7         2009-03-15 13:16 (:0)
cis192   pts/0        2009-03-15 13:19 (:0.0)
cis192   pts/1        2009-03-15 13:19 (:0.0)
root     pts/2        2009-03-15 14:11 (legolas)
root     pts/3        2009-03-15 14:11 (arwen)
root@sauron:~#
```

sshd

Sample session

```
[root@elrond ~]# ssh cis192@sauron
The authenticity of host 'sauron (10.10.10.200)' can't be established.
RSA key fingerprint is 61:f3:89:a3:b5:a3:2a:b9:6e:f0:9b:59:f5:93:14:b8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'sauron,10.10.10.200' (RSA) to the list of known
hosts.
cis192@sauron's password:
Linux sauron 2.6.27-7-generic #1 SMP Fri Oct 24 06:42:44 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
cis192@sauron:~$ echo This is a secret!
This is a secret!
cis192@sauron:~$ exit
logout
Connection to sauron closed.
[root@elrond ~]#
```

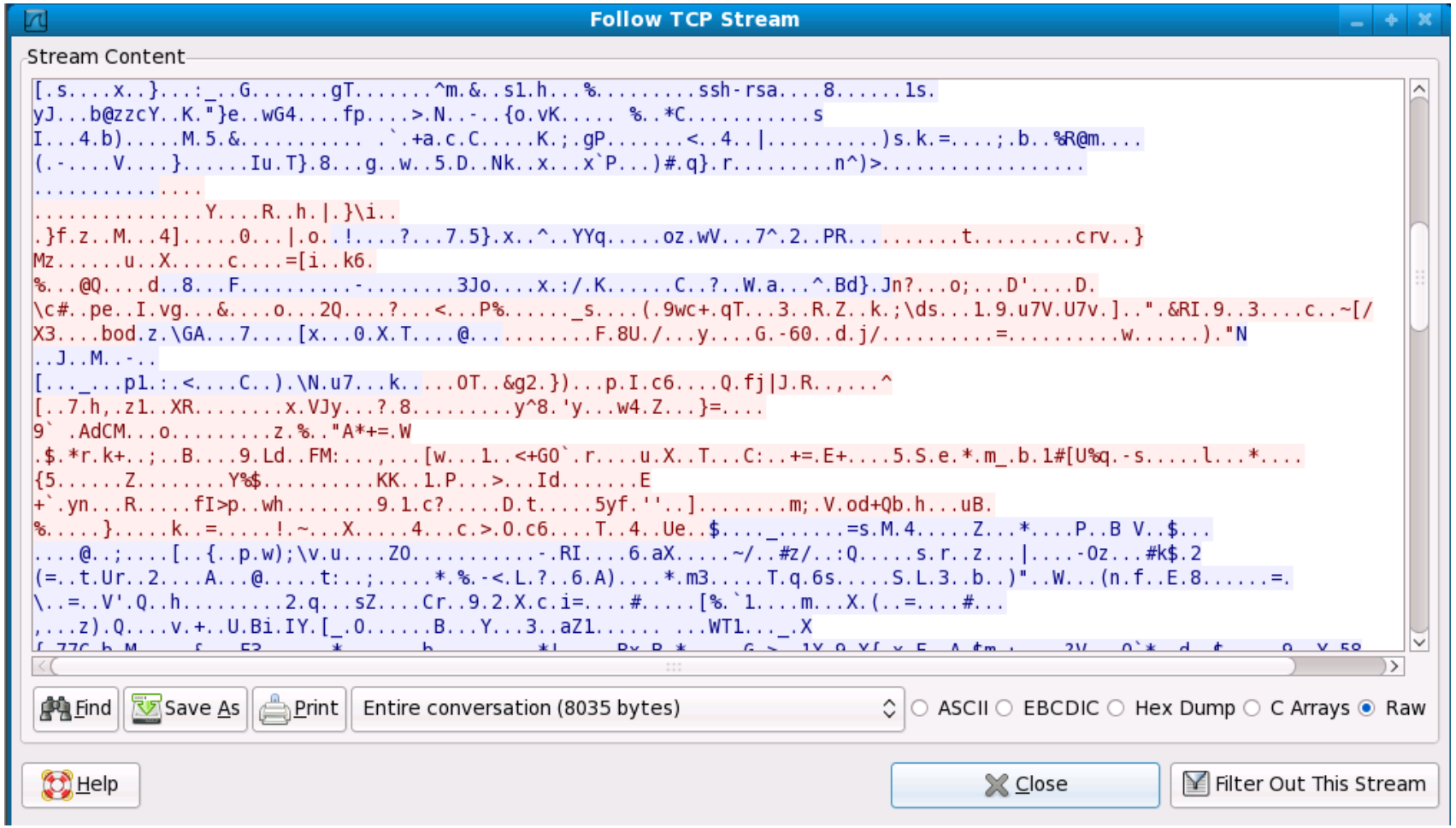
sshd



3 Way hand shake

sshd

The session is encrypted



sshd

TCP Wrappers and sshd

- sshd is compiled with TCP wrappers

```
[root@arwen ~]# type sshd
sshd is /usr/sbin/sshd
[root@arwen ~]# ldd /usr/sbin/sshd
        linux-gate.so.1 =>  (0x00146000)
        libwrap.so.0 => /usr/lib/libwrap.so.0 (0x00fb8000)
        < snipped >
        libpthread.so.0 => /lib/libpthread.so.0 (0x00185000)
[root@arwen ~]#
```

- /etc/hosts.allow – for permitted hosts
- /etc/hosts.deny – to ban hosts

sshd

TCP Wrappers and sshd example

Arwen


```
[root@arwen ~]# cat /etc/hosts.allow
sshd: frodo 192.168. 10.0.0.0/255.0.0.0
in.telnetd:  192.168.2.10  127.0.0.1
vsvtpd: frodo arwen sauron
```

For sshd, Frodo, all 192.168.x.x and all 10.x.x.x hosts are allowed

Sauron at 10.10.10.200 is included. Nosmo at 172.30.1.1 is NOT included

```
[root@arwen ~]# cat /etc/hosts.deny
ALL: ALL
```

Everyone else is denied (this includes Nosmo)

sshd

TCP Wrappers and sshd example

Arwen


```
[root@arwen ~]# cat /etc/hosts.allow
sshd: frodo 192.168. 10.0.0.0/255.0.0.0
in.telnetd:  192.168.2.10  127.0.0.1
vsftpd: frodo arwen sauron

[root@arwen ~]# cat /etc/hosts.deny
ALL: ALL
```

Sauron

```
root@sauron:~# ssh arwen
root@arwen's password:
Last login: Sun Mar 15 20:11:31 2009 from frodo
[root@arwen ~]#
```

Access permitted

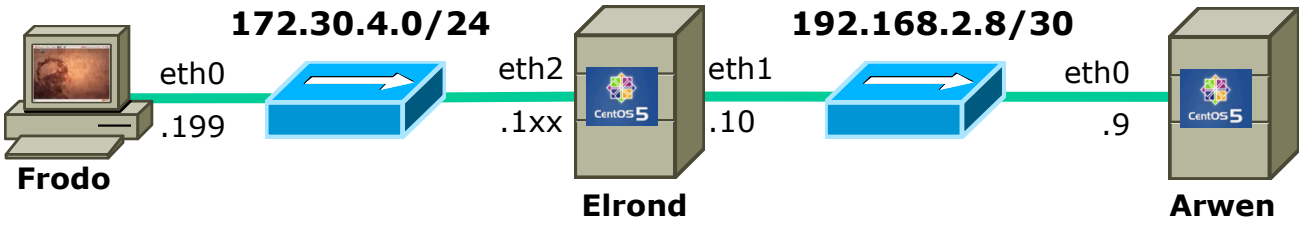
Nosmo

```
[root@nosmo root]# ssh 192.168.2.9
ssh_exchange_identification: Connection closed by remote host
[root@nosmo root]#
```

Access denied

SSH tunneling and port forwarding

SSH Port Forwarding



Public Client

Gateway and Firewall

Private Network Server

Is there a way we can tunnel an insecure protocol, like Telnet, through an SSH connection to reach a private server on our home or business network?

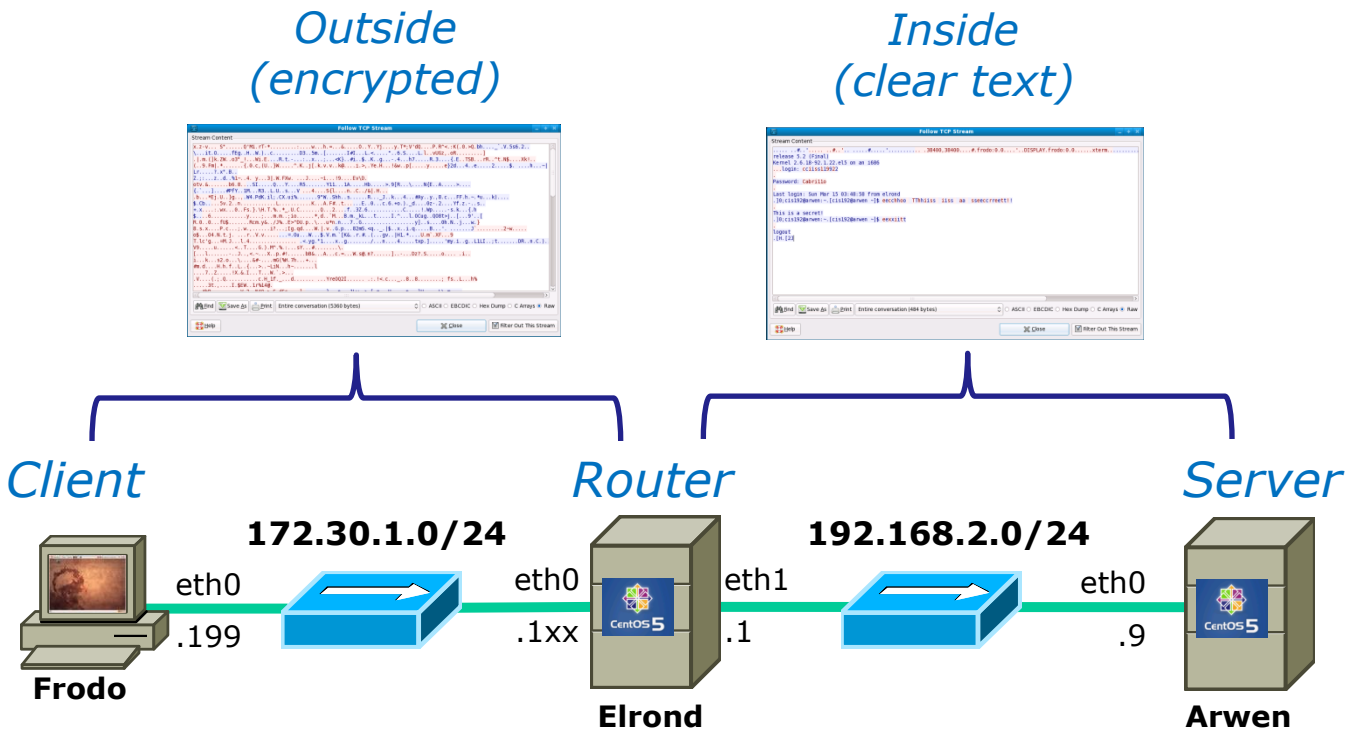
SSH Port Forwarding

-L [bind_address:]port:host:hostport

Specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side. This works by allocating a socket to listen to port on the local side, optionally bound to the specified bind_address. Whenever a connection is made to this port, the connection is forwarded over the secure channel, and a connection is made to host port hostport from the remote machine. Port forwardings can also be specified in the configuration file. IPv6 addresses can be specified with an alternative syntax: [bind_address/]port/host/hostport or by enclosing the address in square brackets. Only the superuser can forward privileged ports. By default, the local port is bound in accordance with the GatewayPorts setting. However, an explicit bind_address may be used to bind the connection to a specific address. The bind_address of \u201clocalhost\u201d indicates that the listening port be bound for local use only, while an empty address or \u2018*\u2019 indicates that the port should be available from all inter\u2010 faces.

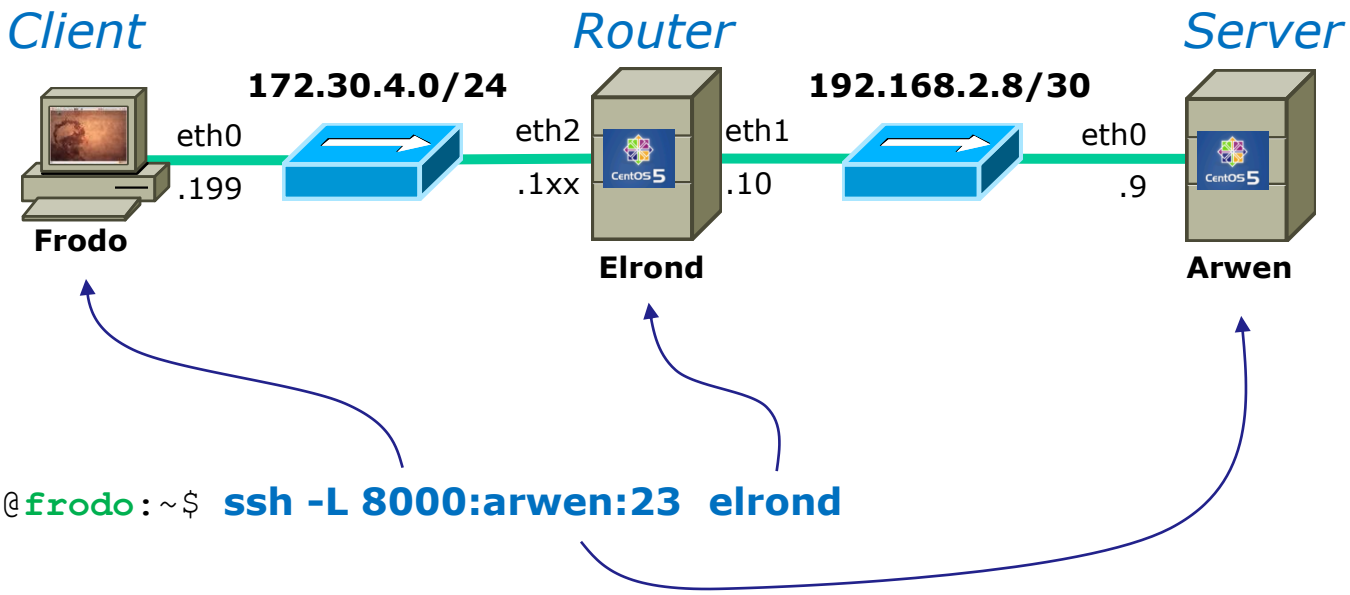
From the man page on ssh ... is that enough documentation for you?

SSH Port Forwarding



In this example we will tunnel a telnet session through an encrypted SSH connection.

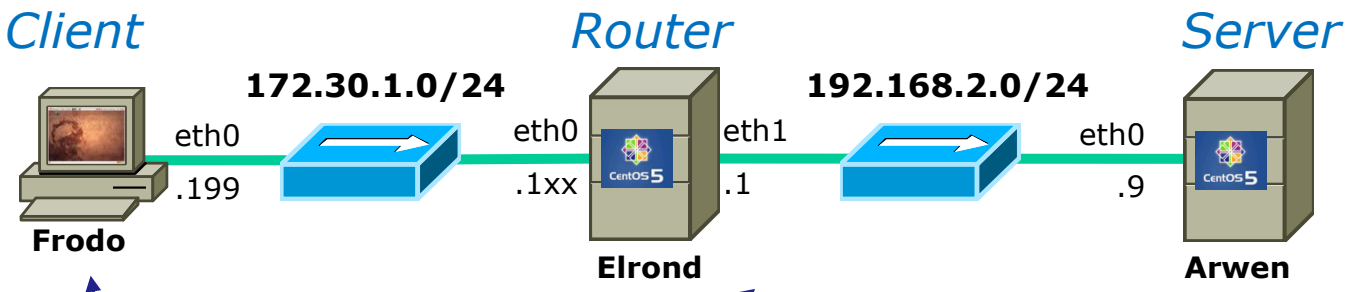
SSH Port Forwarding



Any connection made to port 8000 on Frodo will get forwarded to port 23 on Arwen via Elrond.

The portion of the connection between Frodo and Elrond will be encrypted

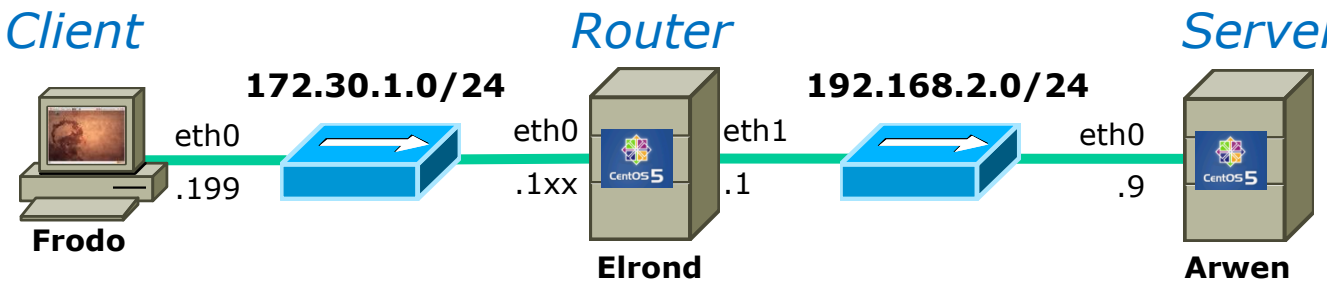
SSH Port Forwarding

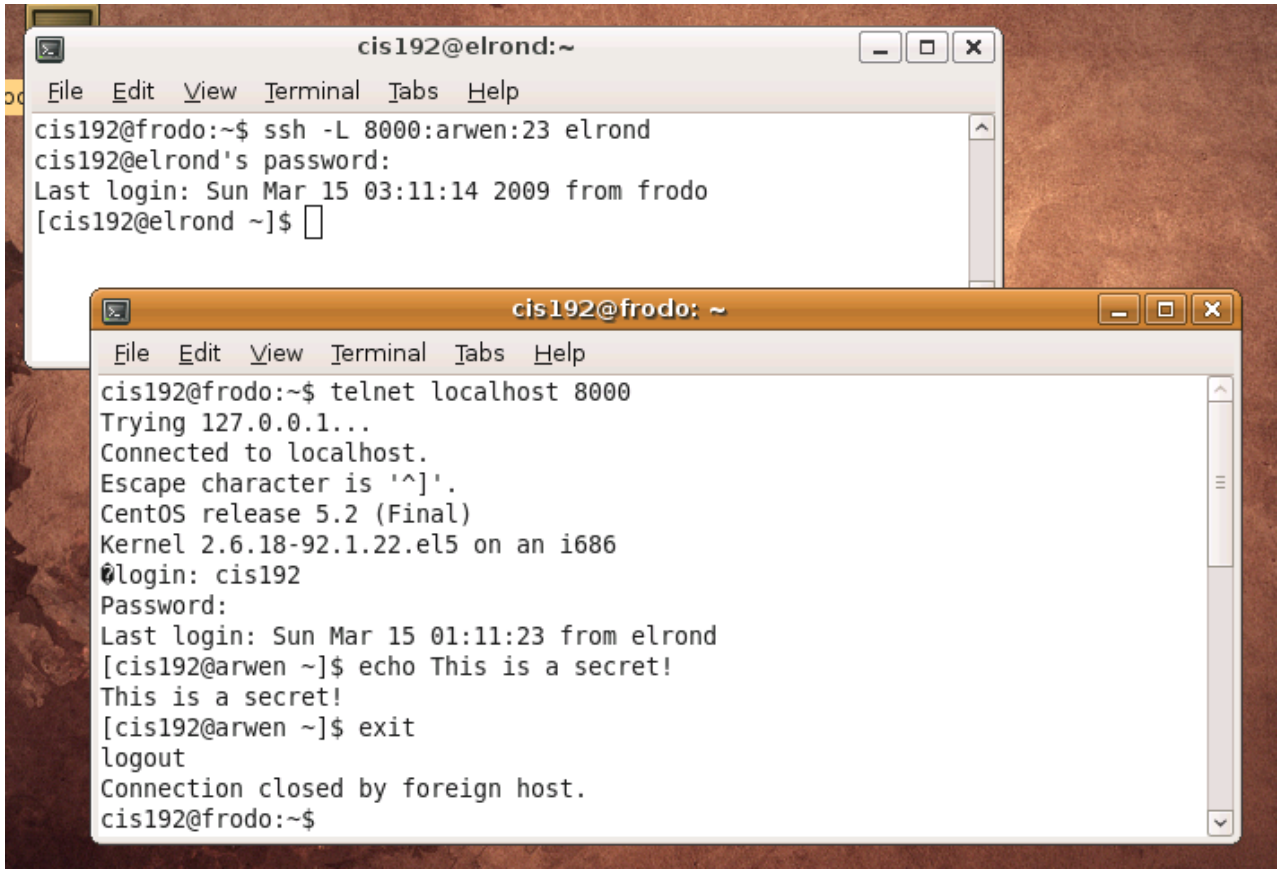



```
cis192@frodo:~$ ssh -L 8000:192.168.2.9:23   172.30.1.107
```

Same as before just using IP addresses instead of names in /etc/hosts.

SSH Port Forwarding

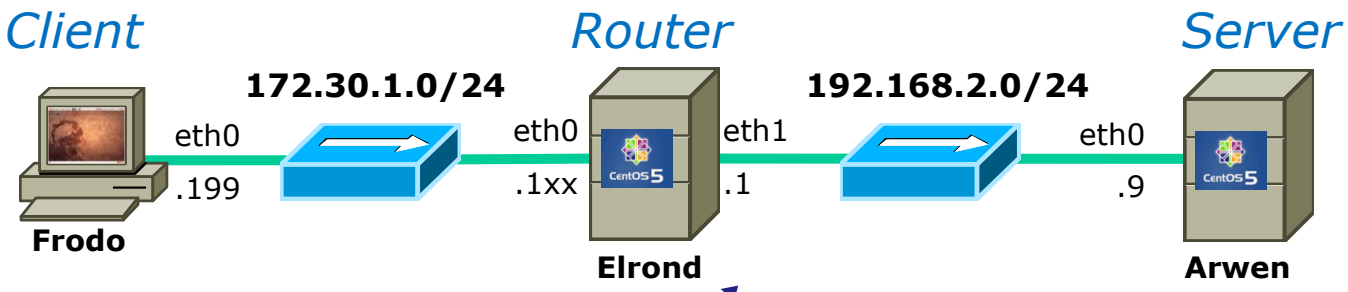




Requires one Frodo terminal to setup SSH port forwarding

And another Frodo terminal to make the Telnet connection

SSH Port Forwarding

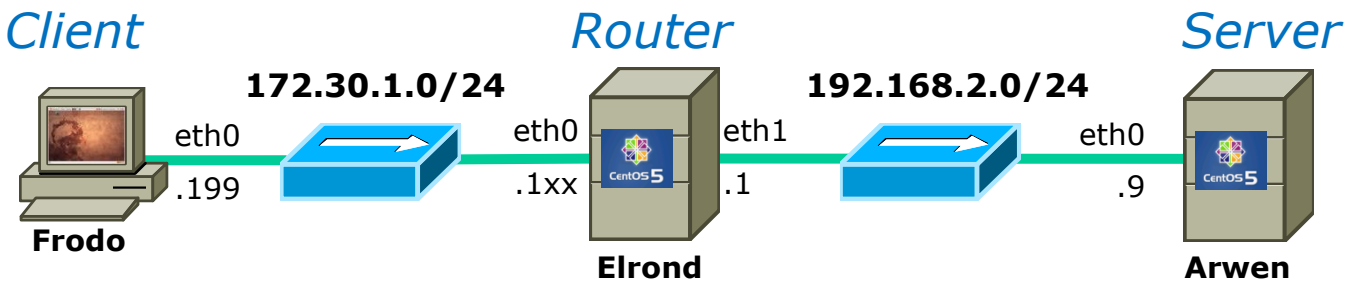



```
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
cis192@elrond's password:
Last login: Sun Mar 15 03:11:14 2009 from frodo
[cis192@elrond ~]$
[cis192@elrond ~]$
[cis192@elrond ~]$ exit
logout

Connection to elrond closed.

cis192@frodo:~$
```

Port forwarding enabled

Port forwarding disabled

SSH Port Forwarding




```
cis192@frodo:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
login: cis192
Password:
Last login: Sun Mar 15 03:48:58 from elrond
[cis192@arwen ~]$ echo This is a secret!
This is a secret!
[cis192@arwen ~]$ exit
logout


Connection closed by foreign host.
cis192@frodo:~$
```

On a different terminal on Frodo:

Telnet "to yourself" at port 8000 and notice you end up on Arwen!

SSH Port Forwarding

Frodo

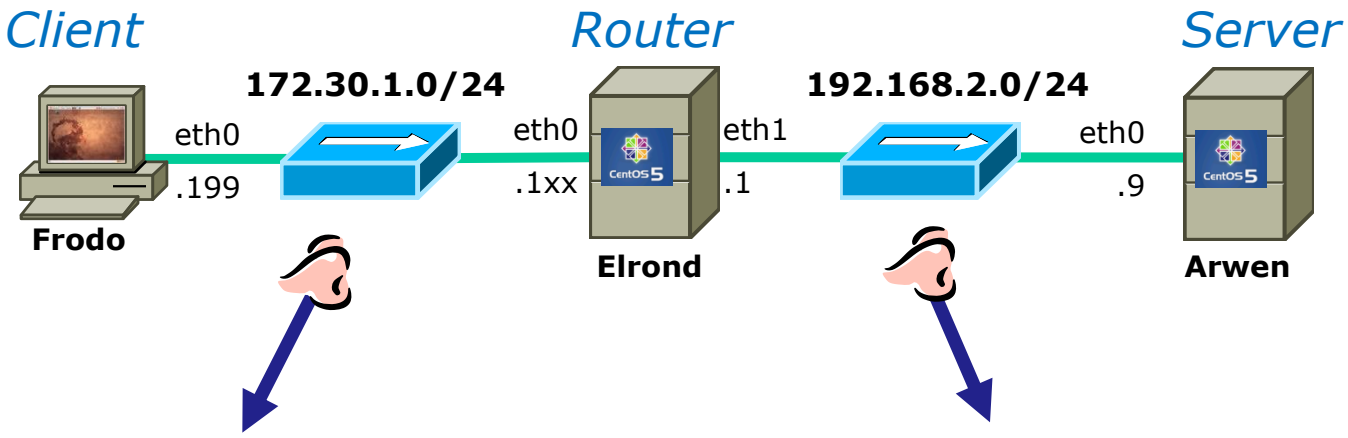
Enable port forwarding in first terminal

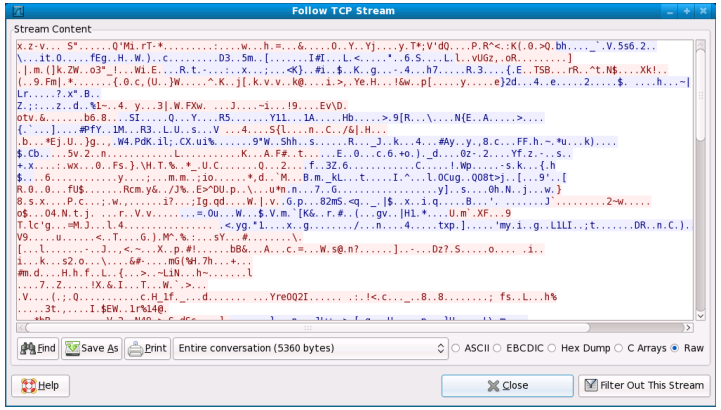
```
cis192@elrond:~
File  Edit  View  Terminal  Tabs  Help
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
cis192@elrond's password:
Last login: Sun Mar 15 03:11:14 2009 from frodo
[cis192@elrond ~]$
```

Use port forwarding in second terminal

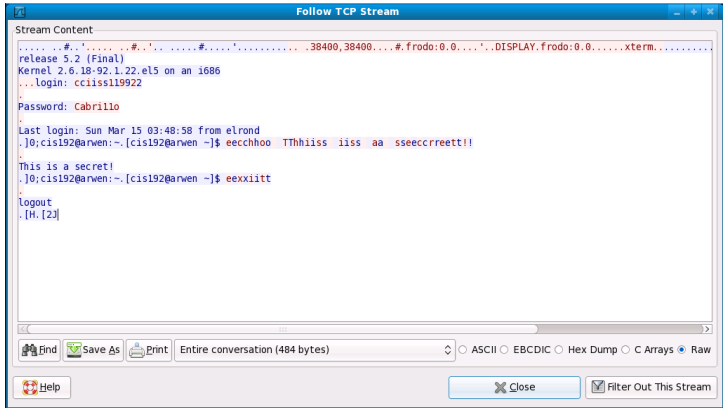
```
cis192@frodo: ~
File  Edit  View  Terminal  Tabs  Help
cis192@frodo:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
login: cis192
Password:
Last login: Sun Mar 15 03:48:58 from elrond
[cis192@arwen ~]$ echo This is a secret!
This is a secret!
[cis192@arwen ~]$ exit
logout


Connection closed by foreign host.
cis192@frodo:~$
```

SSH Port Forwarding

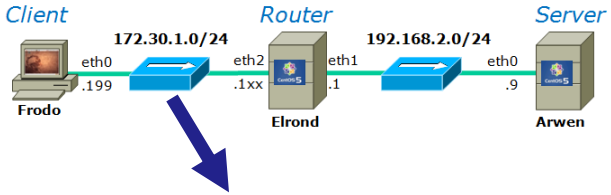




This portion is encrypted



This portion is in clear text



SSH Port Forwarding

Encrypted portion of the connection

(Untitled) - Wireshark

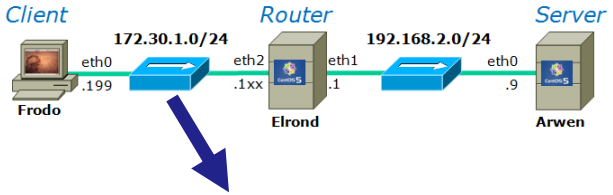
File Edit View Go Capture Analyze Statistics Help

Filter: (ip.addr eq 172.30.4.107 and ip.addr eq 172.30 Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
30	4.479350	172.30.4.199	172.30.4.107	TCP	44022 > ssh [ACK] Seq=561 Ack=625 Win=316 Le
31	4.662263	172.30.4.199	172.30.4.107	SSH	Encrypted request packet len=48
32	4.662313	172.30.4.107	172.30.4.199	SSH	Encrypted response packet len=48
33	4.662325	172.30.4.199	172.30.4.107	TCP	44022 > ssh [ACK] Seq=609 Ack=673 Win=316 Le
34	4.830786	172.30.4.199	172.30.4.107	SSH	Encrypted request packet len=48
35	4.834560	172.30.4.107	172.30.4.199	SSH	Encrypted response packet len=48
36	4.834600	172.30.4.199	172.30.4.107	TCP	44022 > ssh [ACK] Seq=657 Ack=721 Win=316 Le
37	5.581184	172.30.4.199	172.30.4.107	SSH	Encrypted request packet len=48
38	5.586744	172.30.4.107	172.30.4.199	SSH	Encrypted response packet len=48
39	5.588110	172.30.4.199	172.30.4.107	TCP	44022 > ssh [ACK] Seq=705 Ack=769 Win=316 Le
40	5.588788	172.30.4.107	172.30.4.199	SSH	Encrypted response packet len=48
41	5.589934	172.30.4.199	172.30.4.107	TCP	44022 > ssh [ACK] Seq=705 Ack=817 Win=316 Le
42	7.824815	172.30.4.199	172.30.4.107	SSH	Encrypted request packet len=48

Frame 10 (118 bytes on wire, 118 bytes captured)
Ethernet II, Src: Vmware_4e:21:af (00:0c:29:4e:21:af), Dst: Vmware_6f:53:d9 (00:0c:29:6f:53:d9)
Internet Protocol, Src: 172.30.4.107 (172.30.4.107), Dst: 172.30.4.199 (172.30.4.199)
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 44022 (44022), Seq: 161, Ack: 257, Len: 64
SSH Protocol

Frame (frame), 118 bytes | Packets: 168 Displayed: 168 Marked: 0 Dropped: 0 | Profile: Default



SSH Port Forwarding

Encrypted portion of the connection

Follow TCP Stream

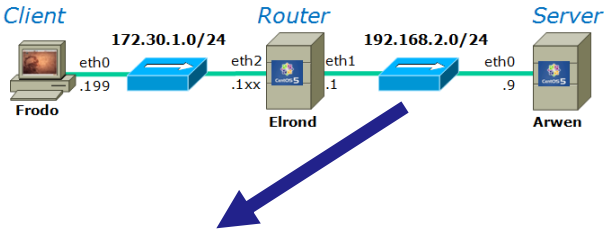
Stream Content

```
x.z-v... S".....Q'Mi.rT-*.........:....w...h.=...&.....O..Y..Yj....y.T*;V'dQ....P.R^<.:K(.0.>Q.bh...._`.V.5s6.2..
\...it.O.....fEg..H..W.)..c.........D3..5m..[.......I#I...L.<....."..6.S....L.l..vUGz,.oR.........]
.|.m.(]k.ZW..o3"_!...Wi.E....R.t.-...:..x...;...<K}..#i..$..K..g...-.4...h7.....R.3....{.E..TSB...rR..^t.N$....Xk!..
(..9.Fm|.*.......{.0.c,(U..}W.....^.K..j[.k.v.v..k@....i.>,.Ye.H...!&w..p[.....y.....e}2d...4..e.....2.....$. ....h...~|
Lr.....?.x".B..
Z.;:...z..d..%1~..4. y...3|.W.FXw. ...J....~i...!9....Ev\D.
otv.&.......b6.8....SI.....Q...Y....R5.......Y11...1A.....Hb.....>.9[R...\....N{E..A.....>....
{.`...]....#PfY..1M...R3..L.U..s...V ...4....S{l....n..C../&|.H...
.b...*Ej.U..}g..,.W4.PdK.il;.CX.ui%.......9"W..Shh..s......R..._J..k...4...#Ay..y.,8.c...FF.h.~.*u...k)....
$.Cb.....5v.2..n............L...........K...A.F#..t......E..0...c.6.+o.)._d....0z-.2....Yf.z.-..s..
+.x....:.wx...0..Fs.}.\H.T.%..*_.U.C.......Q...2....f..3Z.6...........C.....!.Wp.....-s.k...{.h
$....6............y....;...m.m..;io......*,d..`M...B.m._kL...t.....I.^...l.OCug..QO8t>j..[...9'..[
R.0..0...fU$.......Rcm.y&../J%..E>^DU.p..\...u*n.n...7..G.................y]..s....0h.N..j...w.}
8.s.x....P.c...;.w.,.....i?...;Ig.qd....W.|.v..G.p...82mS.<q.._.|$..x..i.q.....B...'. .......J`.........2~w.....
o$...O4.N.t.j. ...r..V.v.....=.Ou...W...$.V.m.`[K&..r.#..(...gv..|H1.*....U.m`.XF...9
T.lc'g...=M.J...l.4.............. .<.yg."l....x..g......../...n....4.....txp.].....'my.i..g..L1LI..;t.......DR..n.C.)..
V9.....u......<..T....G.).M^.%.:...sY...#........\.
[...l.......-..J..,<.~...X..p.#!......bB&...A...c.=...W.s@.n?......]..-...Dz?.S.....o.... .i..
i...k...s2.o...\....&#-....mG(%H.7h...+...
#m.d....H.h.f..L..{...>..~LiN...h~.......l
....7..Z.....!X.&.I...T...W.`.>...
.V....(.;.Q...........c.H_1f._...d..... ...YreOQ2I..... .:.!<.c..._..8..8.......; fs..L...h%
.....3t.,....I.$EW..1r%14@.
```

Find | Save As | Print | Entire conversation (5360 bytes) | ASCII | EBCDIC | Hex Dump | C Arrays | Raw

Help | Close | Filter Out This Stream

SSH Port Forwarding



Clear text portion of the connection

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
6	10.945158	192.168.2.10	192.168.2.9	TCP	35155 > telnet [SYN] Seq=0 Win=5840 Len=0 MS
7	10.945253	192.168.2.9	192.168.2.10	TCP	telnet > 35155 [SYN, ACK] Seq=0 Ack=1 Win=57
8	10.946441	192.168.2.10	192.168.2.9	TCP	35155 > telnet [ACK] Seq=1 Ack=1 Win=5888 Le
9	10.973505	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
10	10.974504	192.168.2.10	192.168.2.9	TCP	35155 > telnet [ACK] Seq=1 Ack=13 Win=5888 L
11	10.985690	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...
12	10.993869	192.168.2.9	192.168.2.10	TCP	telnet > 35155 [ACK] Seq=13 Ack=13 Win=5824
13	10.994944	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
14	11.001281	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...
15	11.051578	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
16	11.055691	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...
17	11.083456	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
18	11.083690	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...

Internet Protocol, Src: 192.168.2.9 (192.168.2.9), Dst: 192.168.2.10 (192.168.2.10)

Transmission Control Protocol, Src Port: telnet (23), Dst Port: 35155 (35155), Seq: 52, Ack: 104, Len: 69

Telnet

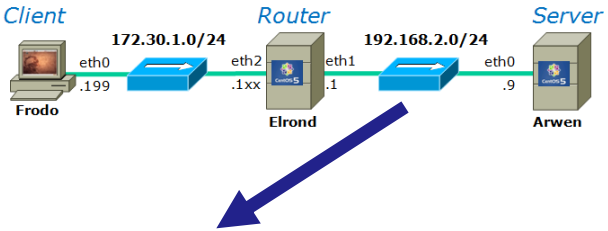
Command: Will Echo

Data: CentOS release 5.2 (Final)\r\n

Data: Kernel 2.6.18-92.1.22.el5 on an i686\r\n

File: "/tmp/etherXXXXruBIW6" 14 ... Packets: 168 Displayed: 168 Marked: 0 Dropped: 0 Profile: Default

SSH Port Forwarding



Clear text portion of the connection

Follow TCP Stream

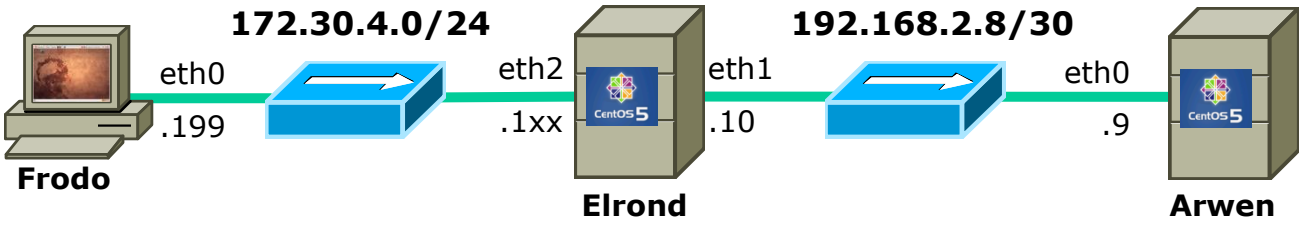
Stream Content

```
..... ..#..'..... ..#..'.. .....#.....'.......... .38400,38400....#.frodo:0.0....'..DISPLAY.frodo:0.0......xterm..........
release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
...login: cciiss119922
.
Password: Cabrillo
.
Last login: Sun Mar 15 03:48:58 from elrond
.]0;cis192@arwen:~.[cis192@arwen ~]$ eecchhoo  TThhiiss  iiss  aa  sseeccrreett!!
.
This is a secret!
.]0;cis192@arwen:~.[cis192@arwen ~]$ eexxiitt
.
logout
.[H.[2J
```

Find | Save As | Print | Entire conversation (484 bytes) | ASCII | EBCDIC | Hex Dump | C Arrays | Raw

Help | Close | Filter Out This Stream

SSH Port Forwarding



Public Client

Gateway and Firewall

Private Network Server

Like it?

Try it with Lab X1

TCP Wrappers

TCP Wrappers

Access controls

- Implemented by the tcpd daemon
- **/etc/hosts.allow** – to specify hosts that may access services
- **/etc/hosts.deny** – to specify hosts that may not access services

Use ldd command on to see if daemon supports TCP Wrappers (i.e. libwrap has been compiled in)

TCP Wrappers

Access controls

- Use **ldd** command to see if daemon supports TCP Wrappers (i.e. libwrap has been compiled in)

```
[root@arwen ~]# type xinetd
xinetd is /usr/sbin/xinetd
[root@arwen ~]# ldd /usr/sbin/xinetd
        linux-gate.so.1 =>  (0x00a8e000)
        libselinux.so.1 => /lib/libselinux.so.1 (0x00cb5000)
        libwrap.so.0 => /usr/lib/libwrap.so.0 (0x007c7000)
        libnsl.so.1 => /lib/libnsl.so.1 (0x004a6000)
        libm.so.6 => /lib/libm.so.6 (0x00e72000)
        libcrypt.so.1 => /lib/libcrypt.so.1 (0x00f7a000)
        libc.so.6 => /lib/libc.so.6 (0x00110000)
        libdl.so.2 => /lib/libdl.so.2 (0x00bd9000)
        libsepol.so.1 => /lib/libsepol.so.1 (0x0054d000)
        /lib/ld-linux.so.2 (0x00f22000)
[root@arwen ~]#
```

TCP Wrappers

/etc/hosts.allow and **/etc/hosts.deny** syntax

daemon : hosts : options

allow
deny
spawn shell command
many more …

ALL
or hostname(s)
or net., e.g. 192.168. matches all 192.168.x.x addresses
or net/netmask , e.g. 172.0.0.0/255.0.0.0 matches all 172.x.x.x addresses
more …

ALL
or name of daemon

TCP Wrapper Examples

```
[root@arwen ~]# cat /etc/hosts.allow
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
sshd: frodo
vsftpd: 172.30.
in.telnetd:  192.168.2.10  127.0.0.1
```

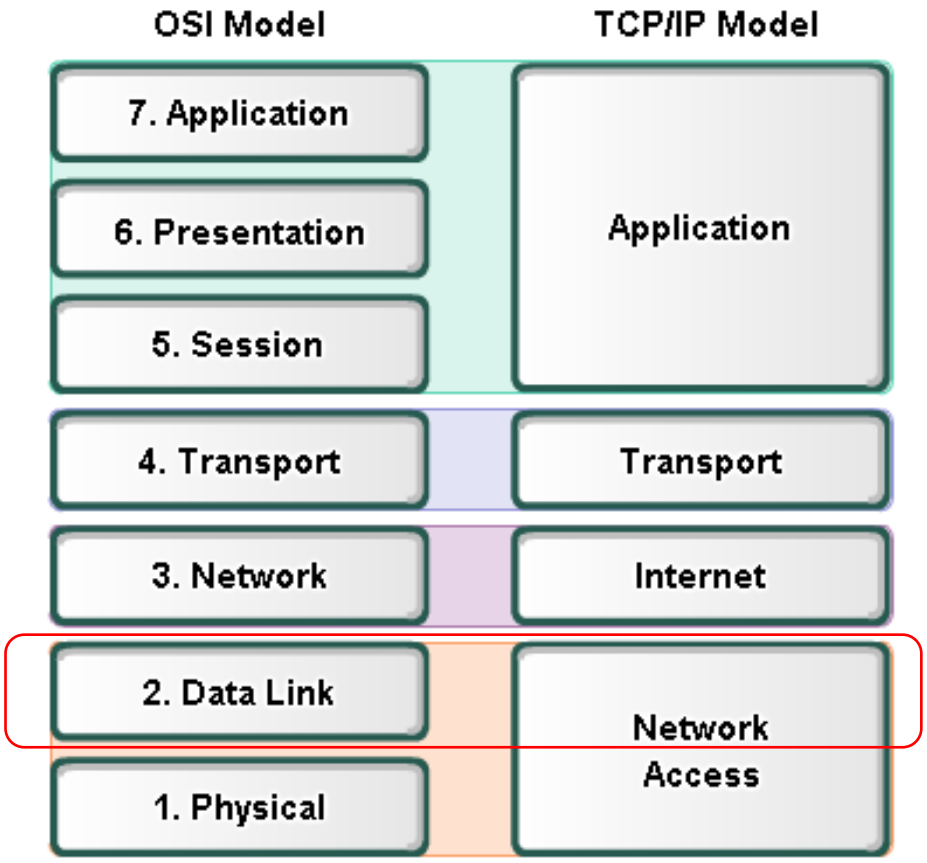
daemons *hosts*

```
[root@arwen ~]# cat /etc/hosts.deny
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!

#deny everything
ALL: ALL
```

All daemons and all hosts

PPP

Layer 2 Technologies

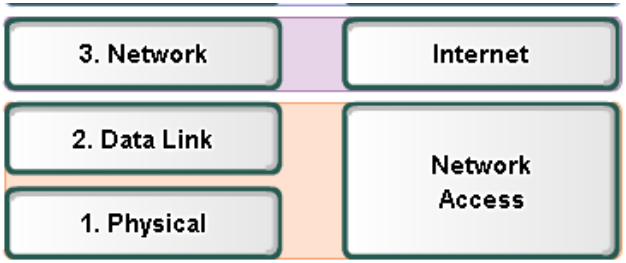


Layer 2 technologies

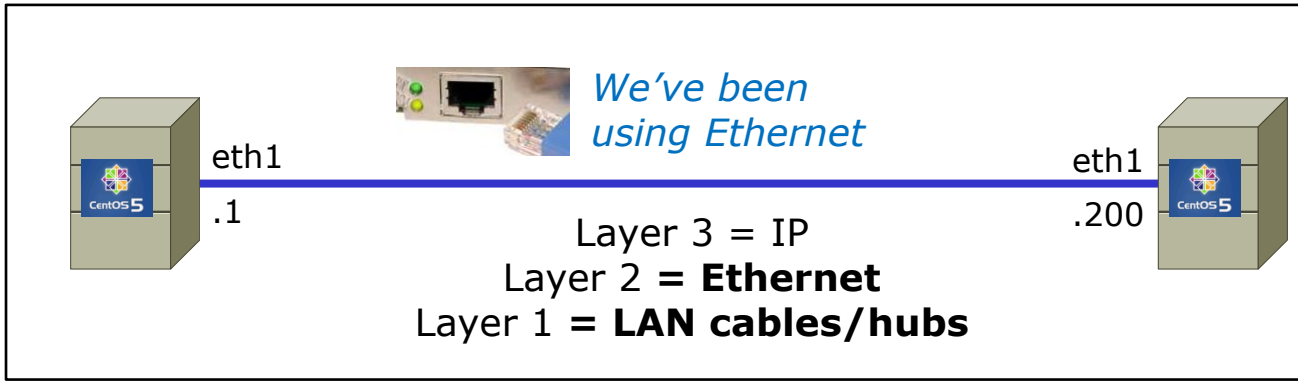
- X.25
- HIPPI
- Ethernet/IEEE 802.3
- Token Ring
- FDDI/CDDI
- Fibre Channel
- ATM
- PPP

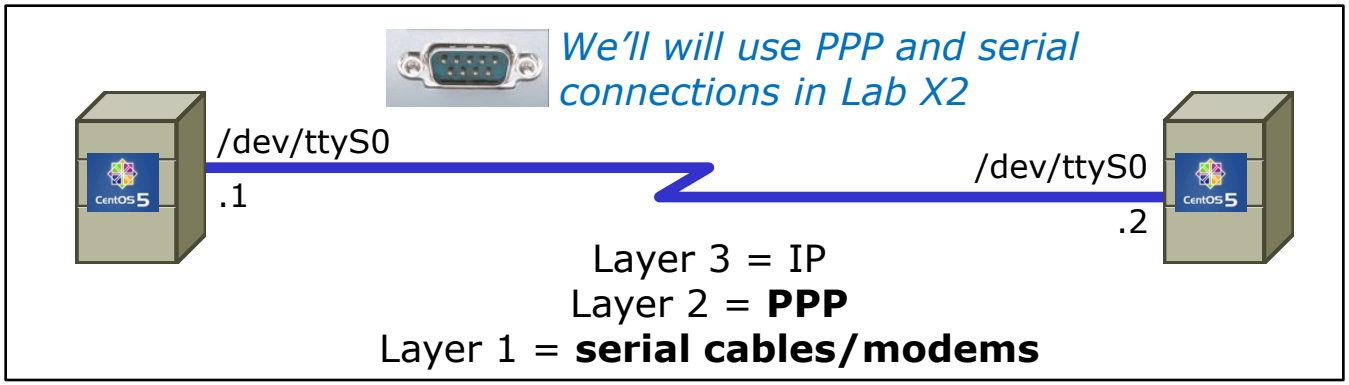
*Up to now we have been using **Ethernet** for Layer 2.*

*In LabX2 we will implement **PPP** over a serial connection.*



Layer 2 Technologies

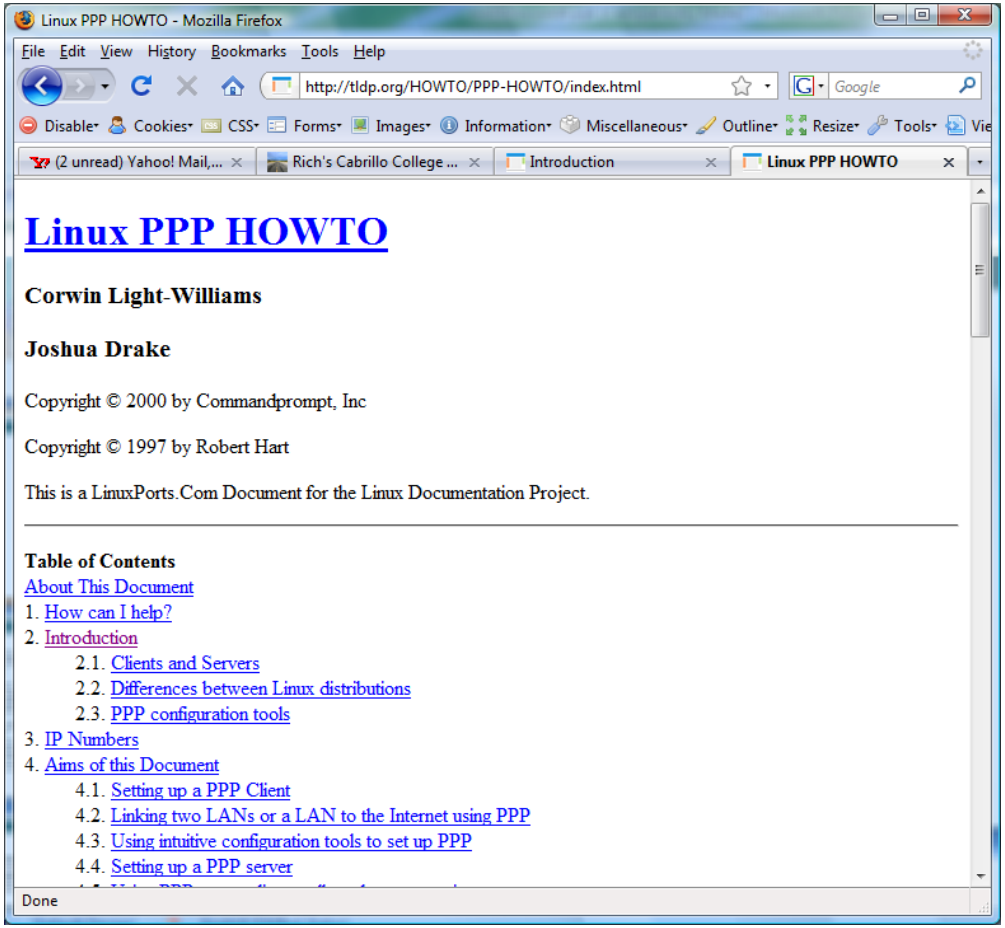




PPP is used rather than Ethernet for serial lines

PPP

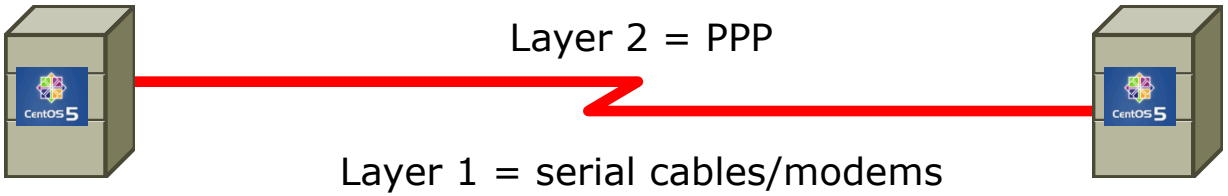
http://tldp.org/HOWTO/PPP-HOWTO/index.html



Old, buts lots of good information on PPP here!

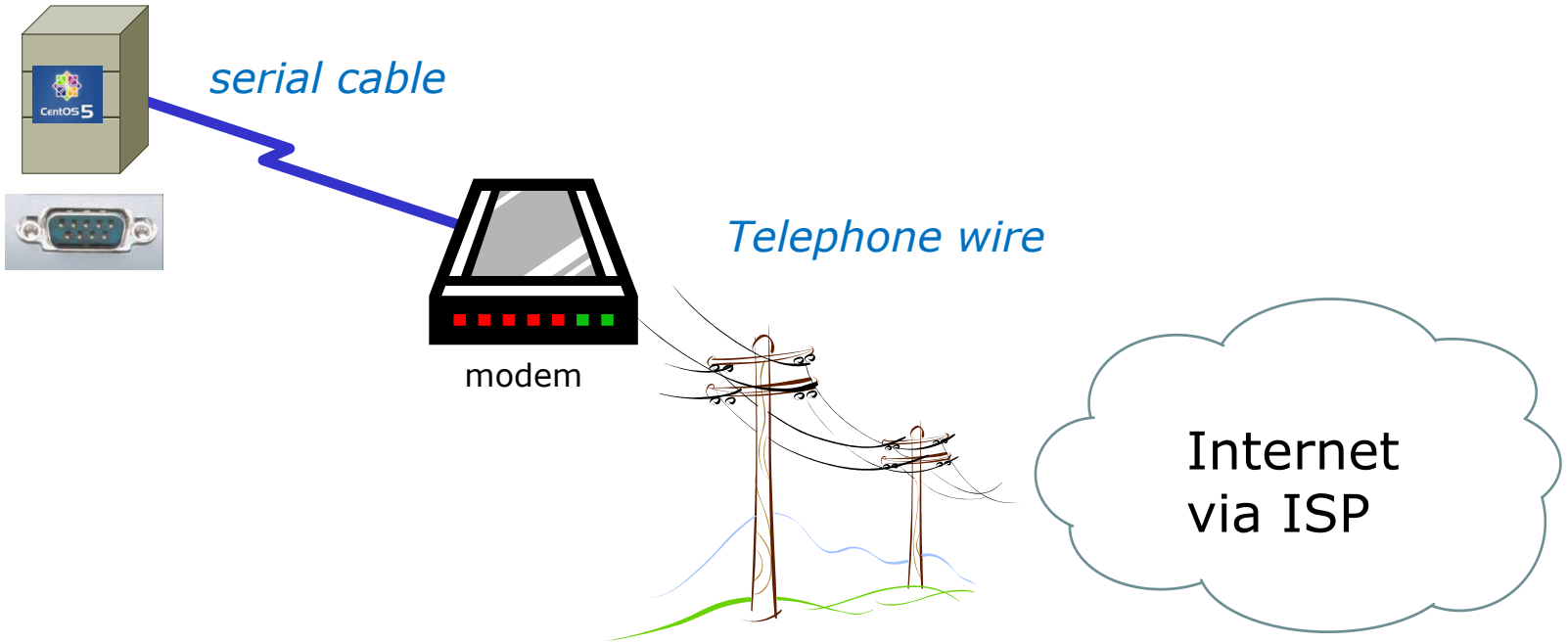
PPP

- PPP = Point to Point protocol (RFC 1331)
- A point to point network has only two hosts (at each end of the serial connection)
- PPP allows running IP and other network protocols over a serial link
- Serial links can be:
  - Direct connections using a null-modem cable
  - Using modems and telephones lines



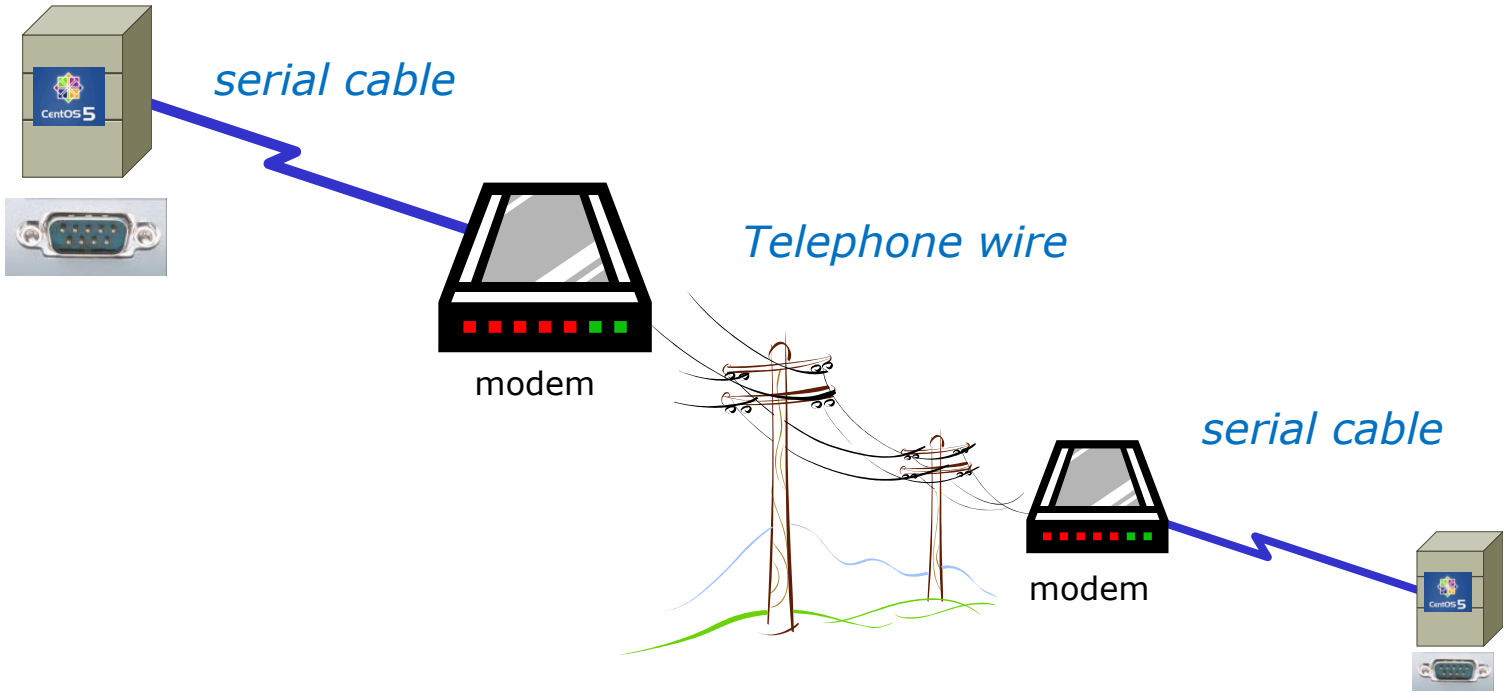
PPP

- PPP can be used as a dial-up connection to the Internet via your ISP



PPP

- PPP can be used as a WAN technology to connect LANs together



Features of PPP and SLIP

Both protocols offer the ability to send datagrams over a serial-line connection.

SLIP

- Works only with TCP/IP
- No error detection unless SLIP headers become corrupted
- Supports header compression only
- Supports only *clear-text* authentication

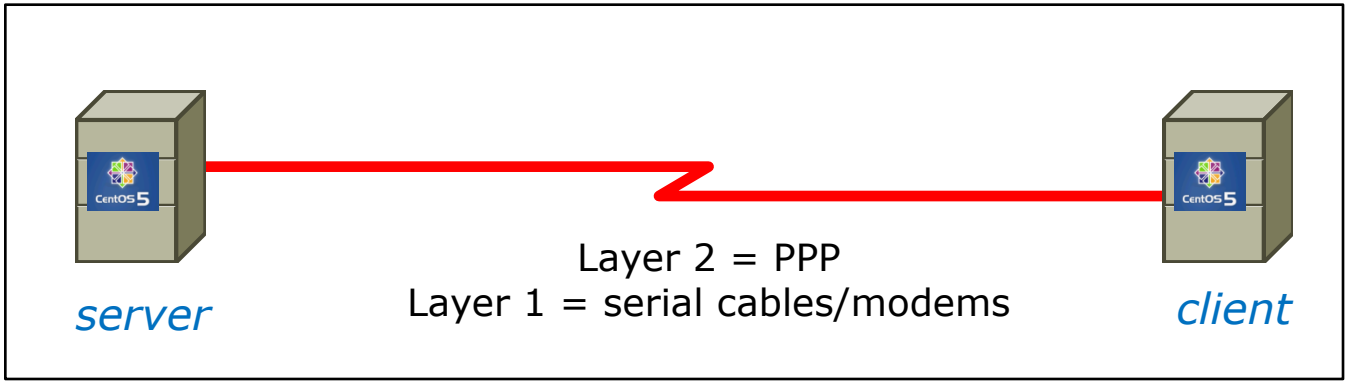
PPP

- Supports TCP/IP as well as UDP/IP, IPX/SPX, and Appletalk
- Built-in error detection
- Supports built-in data compression using the Van Jacobson compression algorithm
- Supports various authentication mechanisms e.g. PAP and CHAP

Password Authentication Protocol

Challenge Handshake Authentication Protocol

PPP Architecture



- PPP is also called a *Peer-to-Peer* protocol because there is fundamentally no difference between the server and the client.
- The ppp daemons (services) must be running on both sides of the connection.
- The computer that initiates the call is called the client, the one who answers the call is the server.

PPP Architecture

PPP runs as two major components:

1. Kernel portion - consists of and manages low-level protocols

```
[root@gothmog ~]# lsmod | grep  "^ppp"
ppp_deflate            9793  2
ppp_async             15169  1
ppp_generic           30037  6 ppp_deflate,ppp_async
```

2. User portion - consists of and manages the authentication protocols
   - **pppd** - runs the various protocols
   - **chat** - provides automated dialing management for modem connections

Both of these programs rely on command line options and/or shell scripts to configure how they operate

Setting Up PPP

- Install the software if necessary which may require building and adding kernel modules:
  - Red Hat, CentOS and Ubuntu already have PPP kernel support out of the box.
  - Make sure the pppd service has been installed:

```
[root@gothmog ~]# rpm -qa | grep ppp
ppp-2.4.4-2.el5
rp-pppoe-3.5-32.1
```

- Check your serial port
  - **setserial /dev/ttyS0** to look for modern, higher speed 16450A/16550A UART chip
  - **stty –a** to look for baud rate, parity and stop bits
- Configure your modem

setserial and stty commands

```
[root@gothmog ~]# setserial /dev/ttyS0
/dev/ttyS0, UART: 16450, Port: 0x03f8, IRQ: 4
[root@gothmog ~]#
```

Has modern UART chip

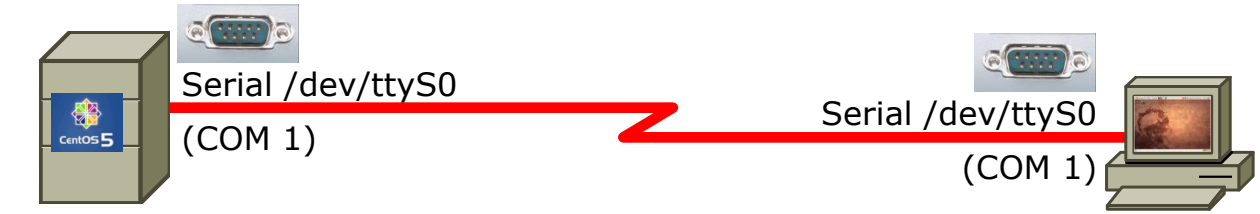
```
[root@gothmog ~]# stty -a
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = M-^?; eol2 = M-^?;
swtch = M-^?; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts -cdtrdsr
-ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc ixany imaxbel iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
[root@gothmog ~]#
```

38400 baud, no parity, data 8 bits, one stop bit, XON/XOFF flow control
*(use **man stty** for complete details)*

Lab X2

Exploring Serial Connections

Console port example with **minicom**



On gothmog, add this line to /etc/inittab:

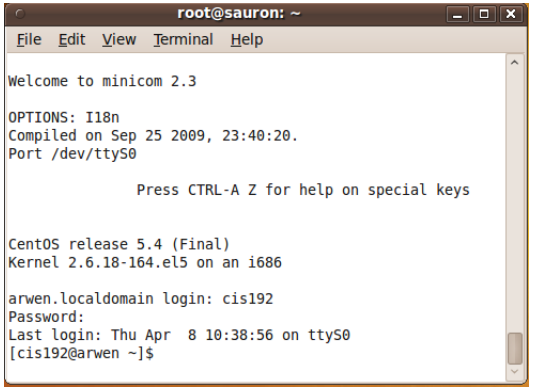
```
s1:35:respawn:/sbin/agetty 38400 ttyS0
```

This enables the login process for any connections to the serial port /dev/ttyS0

On gimli, configure minicom (a terminal emulator) to use:

- /dev/ttyS0
- 38400 baud
- 8 bits data
- no parity
- 1 stop bit
- hardware flow control

Note: PPP is not used yet in this example, just using the serial connection for console access



Login to gothmog using ***minicom -o***

Exploring Serial Connections

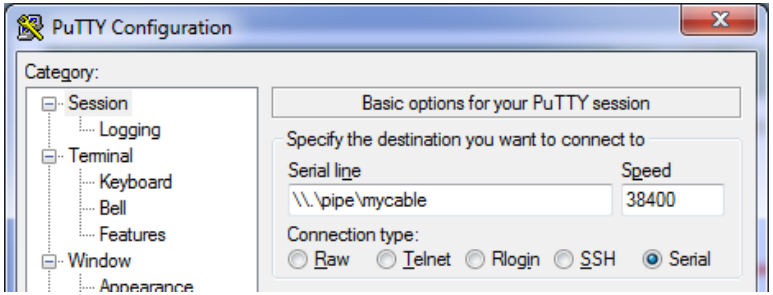
Console port example using **Putty**

Server

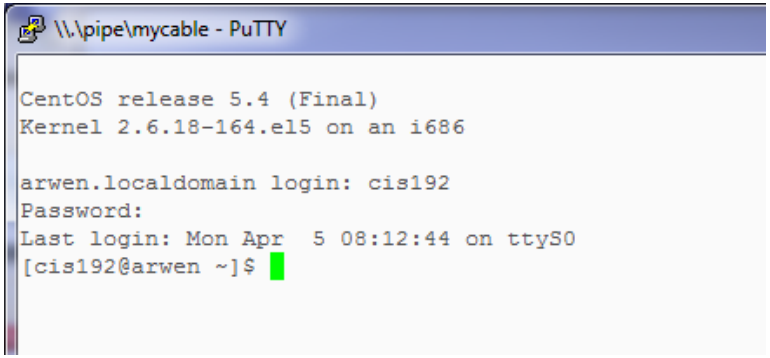
On windows station, configure Putty to use com port or pipe

On gothmog, add this line to /etc/inittab:

```
s1:35:respawn:/sbin/agetty 38400 ttyS0
```

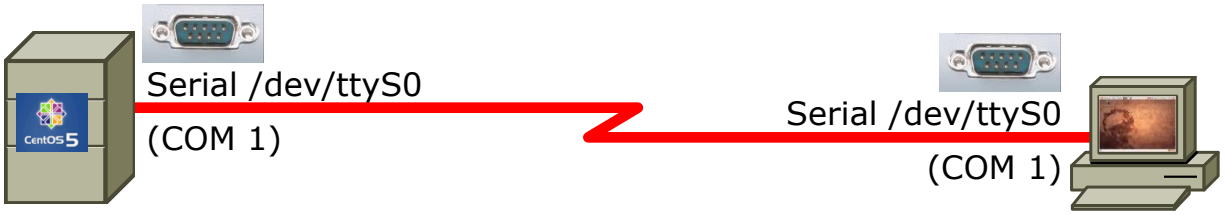


Note: PPP is not used for this, just using the serial connection for console access



Exploring Serial Connections

PPP example with bash_profile script on server, minicom on client (part 1)



gothmog **gimli**

On gothmog,
Add this line to /etc/inittab:

```
s1:35:respawn:/sbin/agetty 38400 ttyS0
```

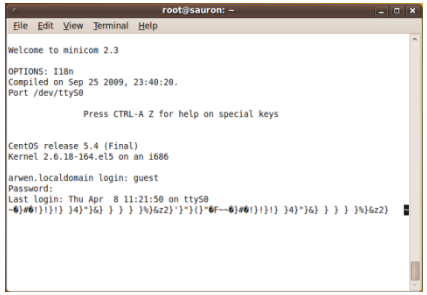
Add a user guest that runs this command at login (added to bash_profile):

```
/usr/sbin/pppd -detach crtscts
proxyarp 10.0.0.1:10.0.0.2 /dev/ttyS0
38400
```

(all on one line)

pppd must be run on both ends to establish the connection

On gimli,



Login as guest on gothmog using ***minicom -o***

Exit minicom and run this command quickly:

```
pppd -detach crtscts
/dev/ttyS0 38400 &
```

(all on one line)

Exploring Serial Connections

PPP example with bash_profile script on server, minicom on client (part 2)

On gimli,

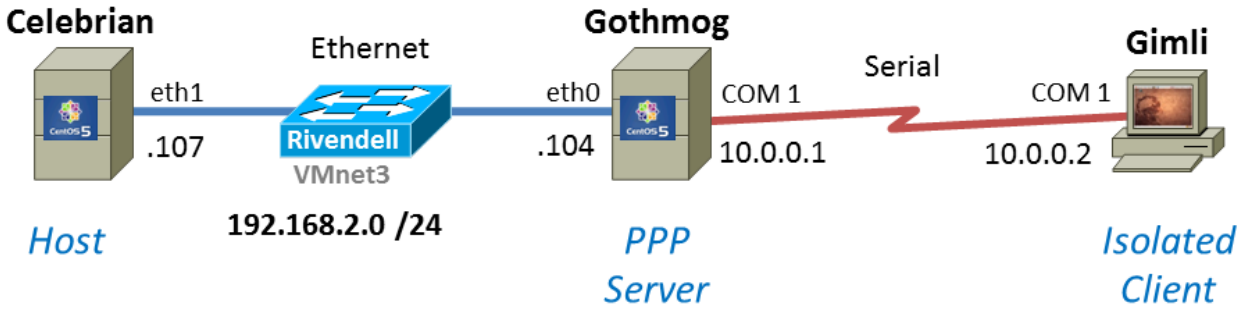
```
root@gimli:~# pppd -detach crtscts /dev/ttyS0 38400 &
[1] 1675
root@gimli:~# Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
Deflate (15) compression enabled
Cannot determine ethernet address for proxy ARP
local  IP address 10.0.0.2
remote IP address 10.0.0.1

root@gimli:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:240 (240.0 B)  TX bytes:240 (240.0 B)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.0.0.2  P-t-P:10.0.0.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:69 (69.0 B)  TX bytes:75 (75.0 B)
```

PPP connection established

Note both the local IP address and remote IP address are shown in ifconfig output

Lab X2

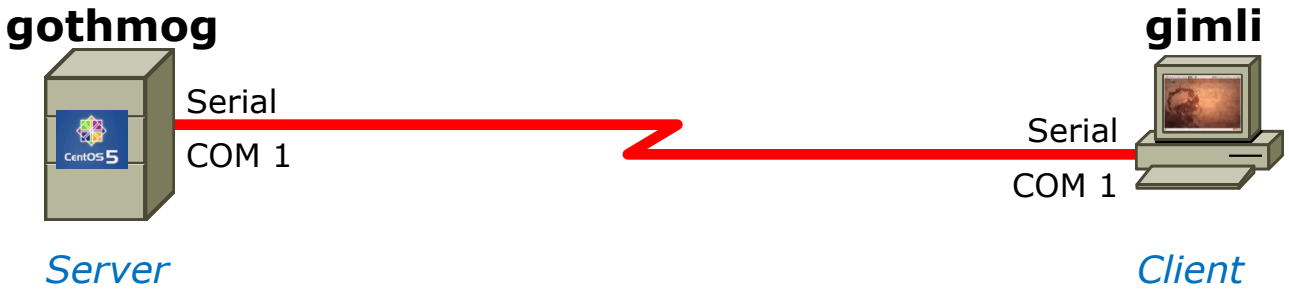
Using a named pipe for the virtual null modem cable between the two serial COM ports



Using Ethernet as the LAN layer 2 protocol over the hub and LAN cables

Using PPP as the WAN layer 2 protocol over the serial connection

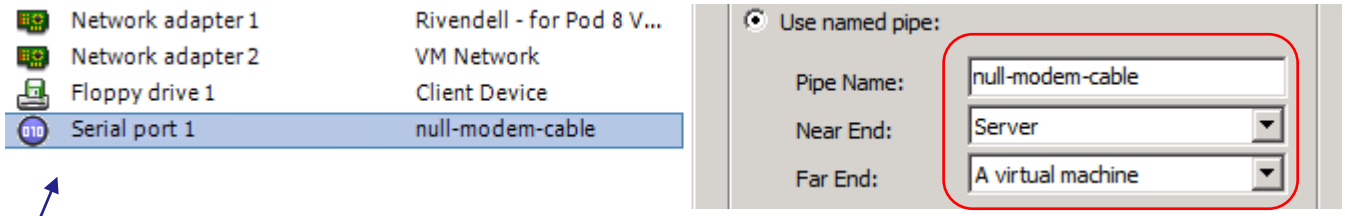
Lab X2 – Serial connections



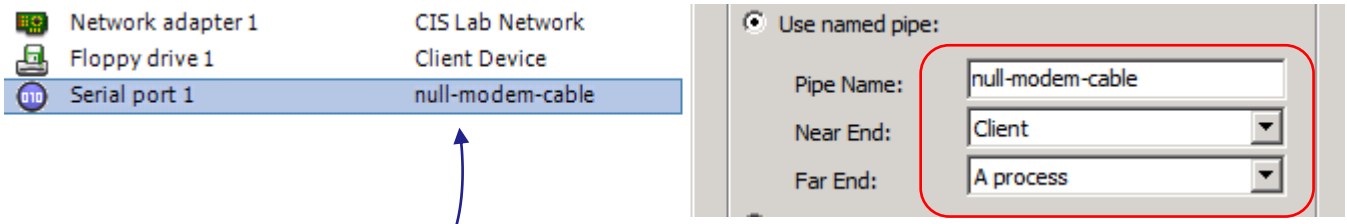
- *If you use real computers to do Lab X2, then you would connect the COM ports using a **null modem cable***
- *If you use VMware or VirtualBox VMs, then you would make a virtual serial connection using OS **pipes***

Lab X2 – Serial connections with VMware ESXi/vSphere

gothmog (the server end)



Gimli (the client end)



Use the Hardware Wizard to add serial ports

Lab X2

In the DOS/Windows world serial ports are called COM 1, COM 2, etc.

```
[root@gothmog ~]# ls -l /dev/ttyS?
crw--w---- 1 ppp  tty  4, 64 Mar 25 06:56 /dev/ttyS0
crw-rw---- 1 root uucp 4, 65 Mar 24 16:39 /dev/ttyS1
crw-rw---- 1 root uucp 4, 66 Mar 24 16:39 /dev/ttyS2
crw-rw---- 1 root uucp 4, 67 Mar 24 16:39 /dev/ttyS3
[root@gothmog ~]#
```

Each serial port is considered by UNIX to be a device. In the past these serial ports were used to connect terminals. Teletypes were terminals without a screen (had a keyboard and printer).

Note: DOS COM1 = Linux /dev/ttyS0

Lab X2 Commands

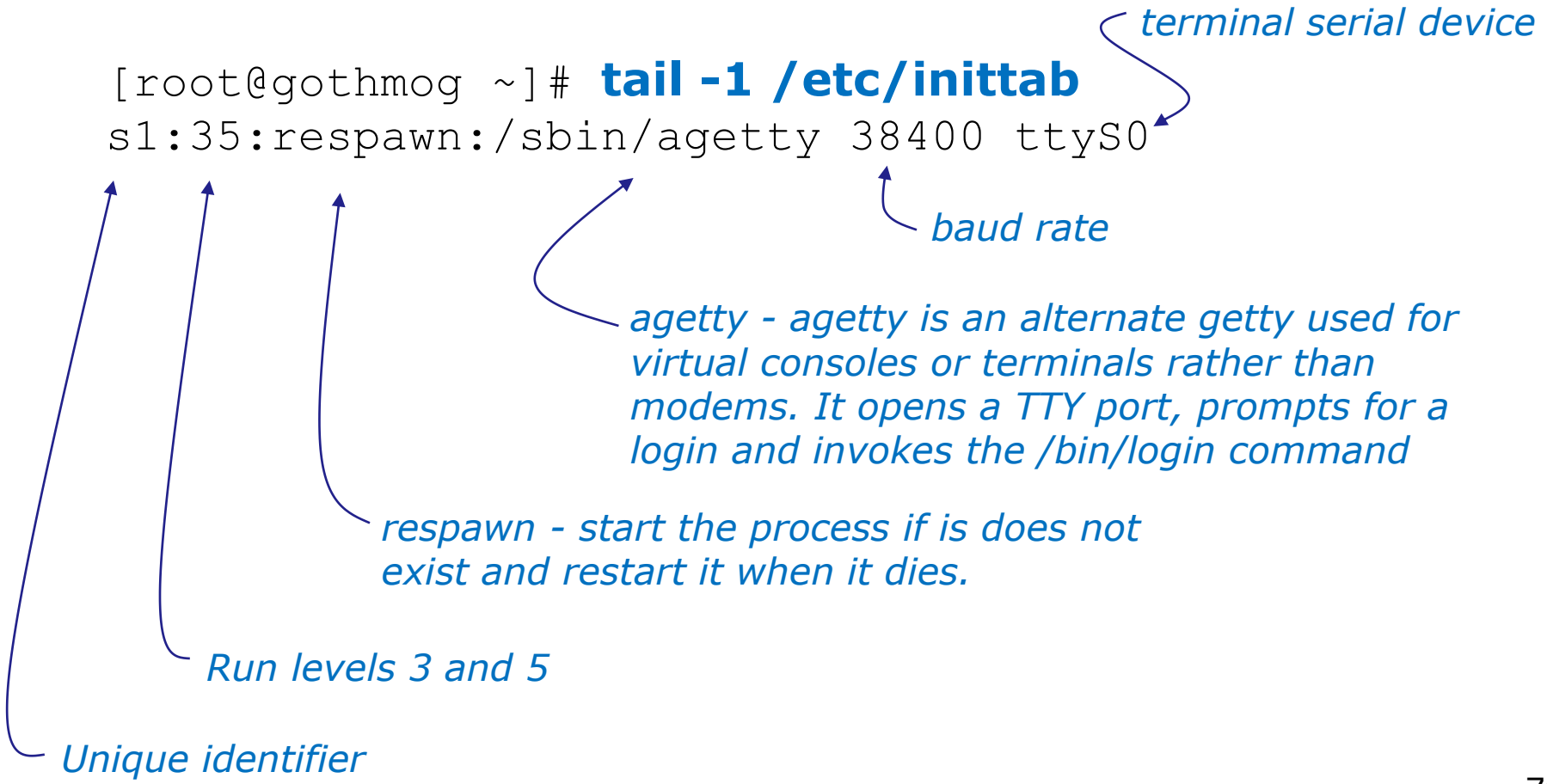
Lab X2


```
[root@gothmog ~]# setserial /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
[root@gothmog ~]#
```

The setserial command sets or reports on serial port configuration.

Lab X2

Handling the login process on the pppd server



Lab X2

Handling the login process on the pppd server

```
[root@gothmog ~]# telinit q
```

*Tells init to reread the **/etc/inittab** file after making changes*

Lab X2

```
[root@gothmog ~]# chmod u+s /usr/sbin/pppd
[root@gothmog ~]# ls -l /usr/sbin/pppd
-r-sr-xr-x 1 root root 312236 Mar 14  2007 /usr/sbin/pppd
```

*This sets a special permission called the **setuid** bit. This allows users to run an executable with the permissions of the executable's owner.*

```
[root@gothmog ~]# stat /usr/sbin/pppd
  File: `/usr/sbin/pppd'
  Size: 312172          Blocks: 632        IO Block: 4096
regular file
Device: fd00h/64768d    Inode: 308263      Links: 1
Access: (4555/-r-sr-xr-x)  Uid: (    0/    root)   Gid: (
0/    root)
Access: 2010-04-04 03:20:12.000000000 -0700
Modify: 2009-01-20 20:27:13.000000000 -0800
Change: 2010-04-04 19:45:23.000000000 -0700
```

*FYI, the **stat** command provides additional inode information about a file than a long listing (ls –l) does.*

Lab X2

minicom
is a small terminal emulator with a dialing capability

```
[root@gothmog ~]# minicom -s
                  -o
```

-s option is used to setup defaults which are saved in ***/etc/minicom/minirc.dfl***

-o option prevents initialization. Useful for restarting a session

Use ***apt-get install minicom*** *to install on Ubuntu*

Lab X2

minicom
is a small terminal emulator with a dialing capability

```
root@gimli:~# minicom -s
```

Select choice and hit Enter

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Use Escape to go back up one level
Use Enter to make sections
Use Letters to make choices

```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/tty8                                 |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 115200 8N1                                |
| F - Hardware Flow Control : Yes                                       |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

Select option and type new configuration value

Lab X2

```
+---------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyS0                              |
| B - Lockfile Location     : /var/lock                               |
| C -   Callin Program      :                                         |
| D -  Callout Program      :                                         |
| E -    Bps/Par/Bits       : 38400 8N1                               |
| F - Hardware Flow Control : Yes                                     |
| G - Software Flow Control : No                                      |
|                                                                     |
|    Change which setting?                                            |
+---------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

When finished use Esc to exit menu

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Use Save setup as dfl to save

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Use Exit from Minicom to exit

Lab X2

```
root@gimli:~# minicom -o

Welcome to minicom 2.3

OPTIONS: I18n
Compiled on Oct 24 2008, 06:37:44.
Port /dev/ttyS0

               Press CTRL-A Z for help on special keys


CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686

gothmog.localdomain login: cis192
Password:
Last login: Tue Mar 24 17:27:32 on ttyS0
[cis192@gothmog ~]$ hostname
gothmog.localdomain
[cis192@gothmog ~]$



CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686

gothmog.localdomain login:

                          +----------------------+
                          | Leave without reset? |
                          |     Yes       No     |
                          +----------------------+

CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.3    | VT102 | Online 00:01
root@gimli:~#
```

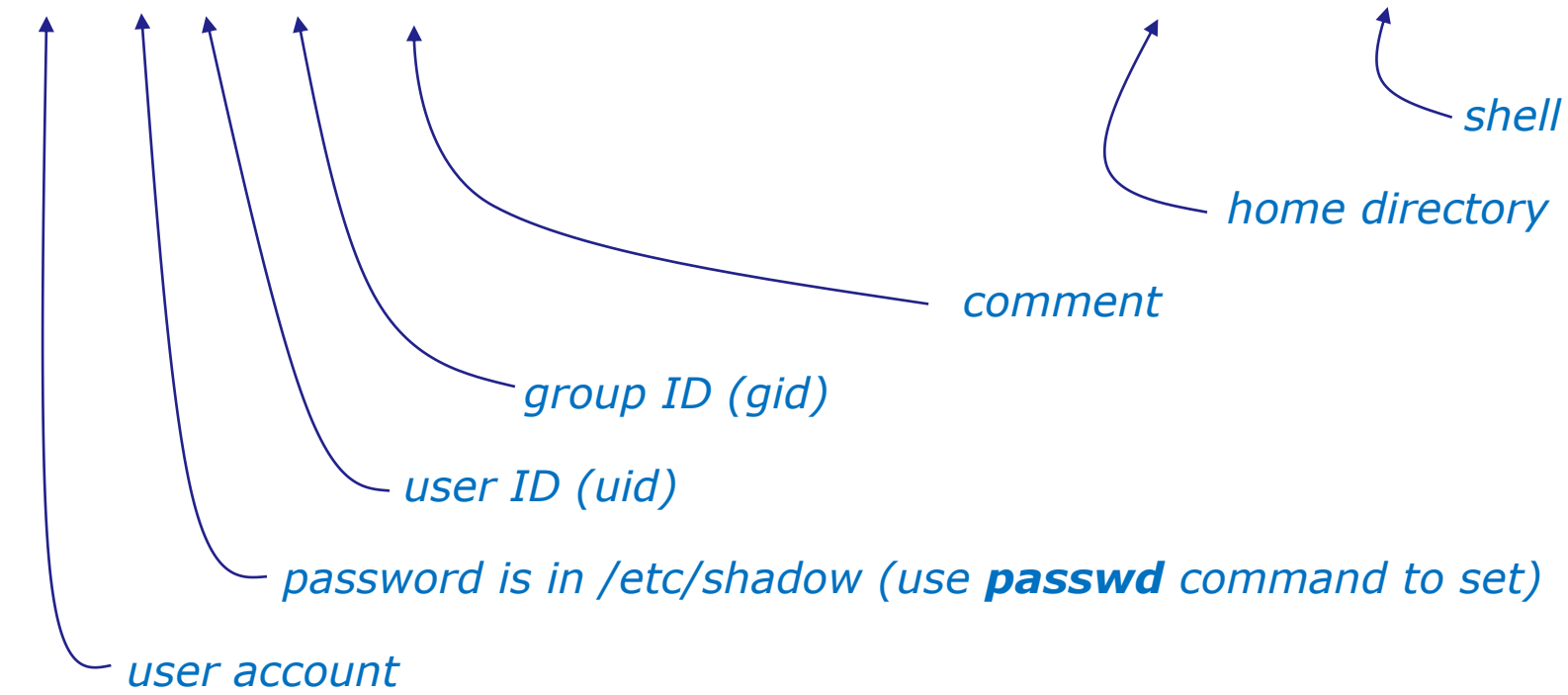
Example session using minicom –o to log into gothmog at other end of the serial connection

Ctrl-A z q
(press Ctrl and A keys together, then z then q)

Lab X2

Creating a new user account on the server side with **useradd**

```
[root@gothmog ~]# useradd -c "Guest account for serial access" guest
[root@gothmog ~]# cat /etc/passwd | grep guest
guest:x:501:501:Guest account for serial access:/home/guest:/bin/bash
```



Lab X2

The .bash_profile file for the guest user

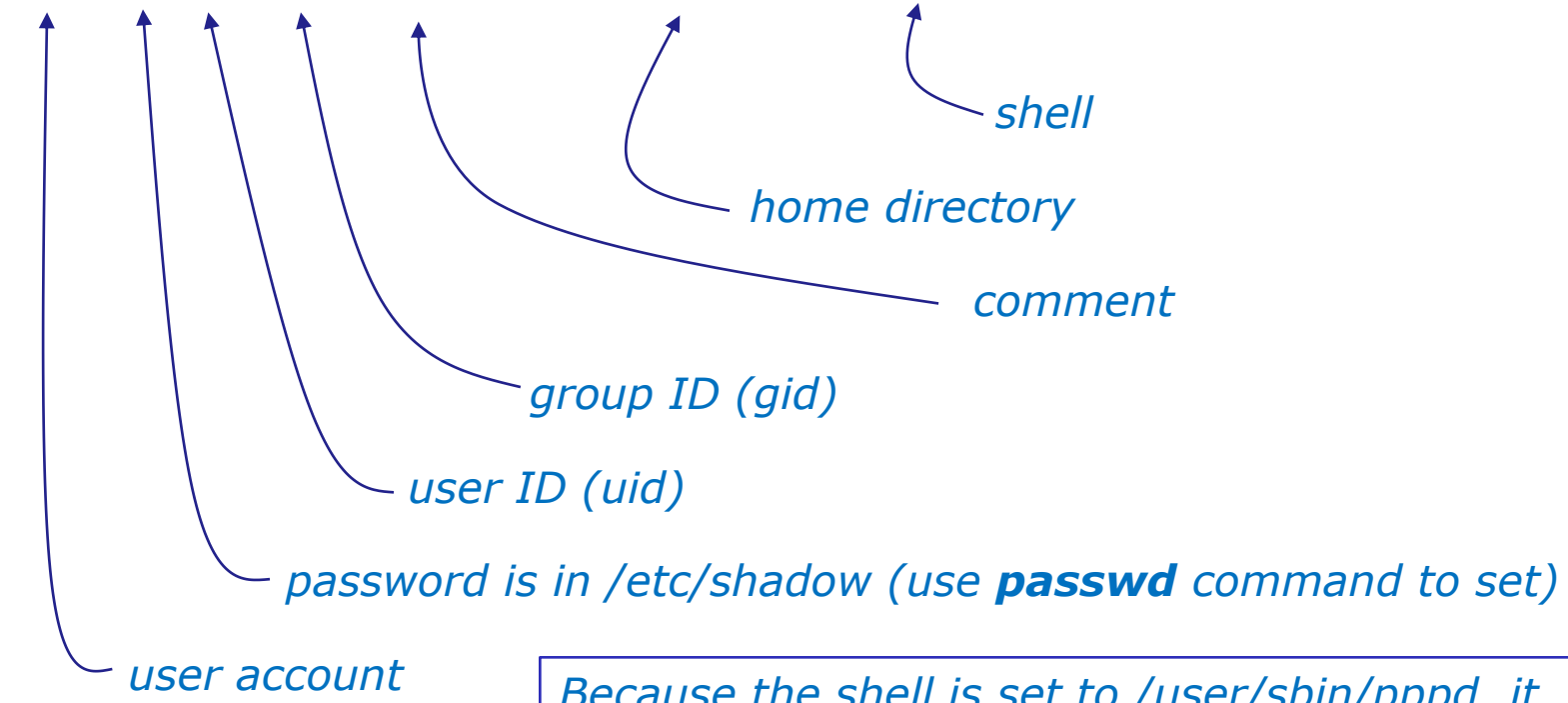
```
[root@gothmog ~]# cat /home/guest/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
/usr/sbin/pppd -detach crtscts proxyarp 10.0.0.1:10.0.0.2 /dev/ttyS0 38400
[root@gothmog ~]#
```

This is used in Part 3 of Lab X2. As soon as guest logs in, the pppd service is run automatically on the server.

Lab X2

Creating a new user account on the server side with **useradd**

```
[root@gothmog ~]# useradd  -c "PPP Account" -d /etc/ppp -s /usr/sbin/pppd ppp
[root@gothmog ~]# cat /etc/passwd | grep ppp
ppp:x:502:502:PPP Account:/etc/ppp:/usr/sbin/pppd
```



Because the shell is set to /user/sbin/pppd, it is run as soon as the ppp user logs in using the option in /etc/ppp/options

Lab X2

The server side options can be put on the command line

/usr/sbin/pppd -detach crtscts proxyarp 10.0.0.1:10.0.0.2 /dev/ttyS0 38400

or in the configuration file

```
[root@gothmog ~]# cat /etc/ppp/options
-detach
crtscts
lock
proxyarp
10.0.0.1:10.0.0.2
/dev/ttyS0
38400
```

Don't fork to become a background process (otherwise pppd will do so if a serial device is specified).

Use hardware flow control using RTS and CTS signals to control the flow of data on the serial port.

Specifies that pppd should use a UUCP-style lock on the serial device to ensure exclusive access to the device.

Add an entry to this system's ARP [Address Resolution Protocol] table with the IP address of the peer and the Ethernet address of this system.

IP address for server-end: client-end

Serial device

Desired baud rate

Refer to **pppd** man page for full details

Lab X2

Command line (client side) to make a connection

With this option, pppd will detach (run in the background) once it has successfully established the ppp connection (to the point where the first network control protocol, usually the IP control protocol, has come up).

Add a default route to the system routing tables, using the peer as the gateway, when IPCP negotiation is successfully completed. This entry is removed when the PPP connection is broken.

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```

command line (client side)

Lab X2

Command line (client side) to make a connection

```
root@gimli:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
root@gimli:~#
root@gimli:~# pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
Serial connection established.
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
Deflate (15) compression enabled
Cannot determine ethernet address for proxy ARP
local  IP address 10.0.0.2
remote IP address 10.0.0.1
root@gimli:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.1        0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
0.0.0.0         0.0.0.0         0.0.0.0         U     0      0        0 ppp0
root@gimli:~#
```

***updetach** option:*
Makes pppd run in the background when link comes up

***defaultroute** option:*
Adds a route to the peer for all traffic

Lab X2

Command line (client side) to make a connection

pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \ "exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"

*The **connect option** can be used to run a script which in this case runs the chat command.*

The chat command is used to handle the login automatically.

Lab X2

Command line (client side) to make a connection

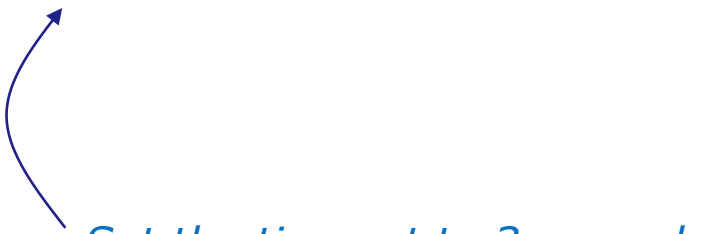
**pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect **
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"

Requests verbose mode for logging purposes.

Lab X2

Command line (client side) to make a connection

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```



Lab X2

Command line (client side) to make a connection

**pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect **
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"

***expect:send** pairs:*
*expect …**ogin** then send **ppp**,*
*expect …**assword** then send **secret***

*Note: the **--ogin** is **sub-expect:sub-send** pair. If the first login is not received, send a single return (empty line) and look again for another login*

Note, because the beginning of the expected word may be garbled due to a flakey modem connection, just look for the end of the word (e.g. login to ogin, password to assword)

Lab X2

Troubleshooting

Tips

- Serial connection can only be used by one pair of computers at a time.
  E.g. Both minicom on gimli and Putty workstation cannot access serial COM 1 on gothmog at the same time.

- View log file:
  cat var/log/messages | grep pppd

Lab X2

Troubleshooting

```
cis192@gimli:~$ su -
Password:
root@gimli:~# ./ppp-on
Serial connection established.
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
LCP: timeout sending Config-Requests
Connection terminated.
Modem hangup
root@gimli:~#
```

Remove default gateway on gothmog

Lab X2

Troubleshooting

```
root@gimli:~# ./ppp-on
Connect script failed
root@gimli:~#
```

Make sure you have logged out from any previously made serial connections. You may need to run minicom –o again to see if you are still logged in as guest.

The Final Exam

From the syllabus on the website:

Student Learner Outcomes

- Install and configure a local area network (LAN) that meets the needs of a small business.
- Troubleshoot and repair malfunctions in common network services.

Objectives

Upon satisfactory completion of the course, students will be able to:

- Use basic network terminology to describe the five layers of the TCP/IP Reference Model, and describe at least one major function of each layer.
- Locate a specific Request For Comment (RFC) article on the Internet.
- Use the arpwatch daemon to collect IP/hardware addresses, and manually add an address to the ARP table.
- Install the device drivers and configure the network interface card (NIC) of a Linux system so that it may join a network.
- Configure appropriate IP addresses, network and subnet masks, and broadcast addresses based on the size and number of network segments required.
- Connect multiple network segments together using Linux servers as routers and configuring the appropriate routing tables.
- Use a network sniffer to analyze network traffic between two hosts.
- Plan a subnet topology based upon a given set of constraints and performance needs.
- Define the term 'socket' and describe its importance to the transport layer of the protocol stack.
- Create a secure tunnel between two hosts that allows port forwarding into a private network.
- Configure a network service with security restrictions for its use using either TCP Wrappers or a superdaemon.
- Install and configure DHCP to assign reserved and dynamic IP addresses, a gateway, a DNS server, and a domain name to a client.
- Use iptables to build a permissive firewall by selectively filtering packets based on protocol type.
- Use Network Address Translation (NAT) to allow hosts on a private network to access the Internet.
- Identify, isolate, and correct malfunctions in a computer network.

*All Cabrillo College classes have "**SLOs**" (Student Learner Outcomes) which get assessed.*

The final exam is the assessment for this course.

It's also a good excuse to bring pizzas to class!

The Final Exam

- Worth 60 points (plus some "uncapped" extra credit)
- Time limit = 2 hours 50 minutes (1-3:50PM, Room 2501, Dec 13th)
- Stations will be assigned to students by the instructor when they enter the classroom.
- Multiple implementation levels which must be done and recorded in sequence.
- To get credit for a level you **must submit requested information on Opus**. In addition you **must demonstrate your final level to the instructor** and leave your VMs running at the end of the test.
- Open book, open notes, open computer.
- During the exam, students may not receive or give assistance to others.
- Exam is available in advance on the website so students can practice.
- Remote students **must** make arrangements in **advance** if they cannot be in the classroom for the exam.
- Contact the instructor if you wish to take the exam online prior to Dec 13th.

The Final Exam

7	12/6	**Quiz 5** **PPP and WAN Protocols** • Connect two LANs together through a serial line • Configure a PPP server and associated support files • Identify, isolate, and fix problems associated with PPP **Materials** • Presentation slides (download) **TBA Assignment** • Exam Prep • Extra Credit Lab X2 (PPP) **CCC Confer** • Enter virtual classroom • Class archives	14.9	Lab 6
- -	12/13	**Final Exam (1-3:50PM) Room 2501**		5 posts Extra Credit Labs

The exam is available now on the website.

Practice, practice, practice!

Wrap

New commands, daemons:

- pppd
- chat
- minicom
- setserial
- stty

Configuration files

- /etc/ppp/options
- /etc/minicom/minirc.dfl

Next Class

No Lesson, just the final

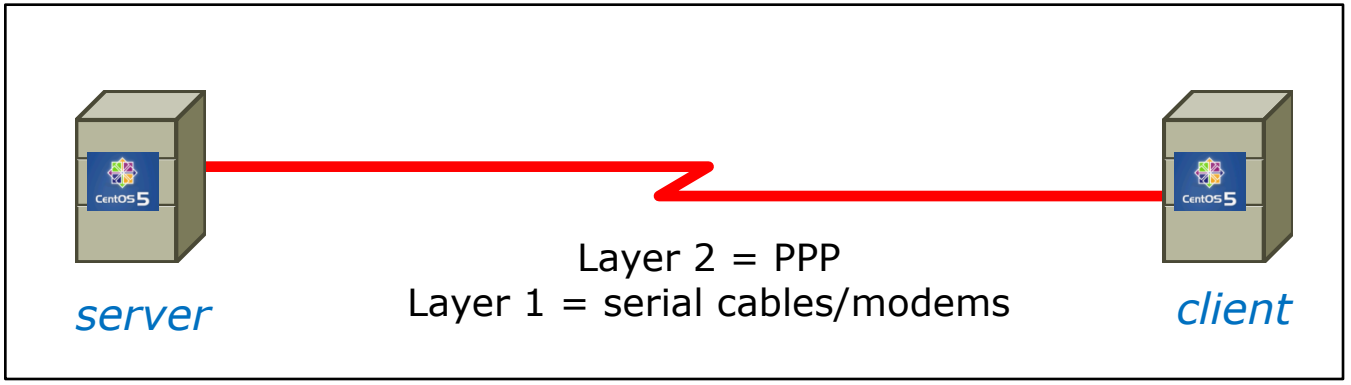
Final Exam Workshop

1. Download the final exam from the website
2. Do a practice run on it and see how far you get between now and the end of class today.
3. Remember, collaboration is encouraged **prior** to the actual final.

Use the forum this week to ask, answer and clarify questions

Backup

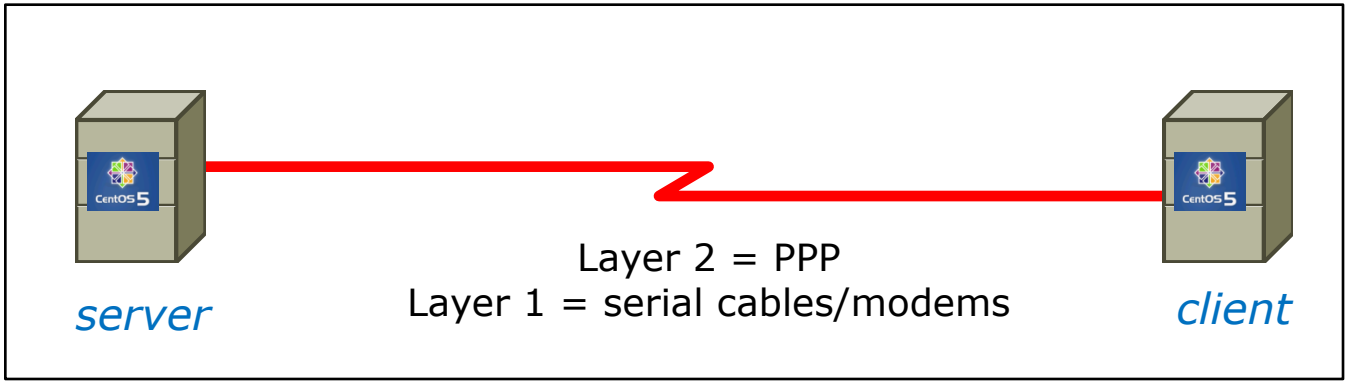
PPP Architecture (continued)



- Network Control Protocol (NCP) provides PPP with a means of differentiating between the different stacks it can transport, such as using IPCP for delivering TCP/IP packets.

- Authorization Protocol Provides a built-in authentication mechanism for PPP connections using either:
  - Password Authentication Protocol (PAP) or
  - Challenge Handshake Authentication Protocol (CHAP)

PPP Architecture (continued)



- Link Control Protocol (LCP) negotiates important link establishment options such as the maximum datagram size. Also helps to facilitate automated link establishment setup.

- High-level Data Link Control Protocol (HDLC) Provides frame boundary information and an added checksum for built-in error detection.